Rui Miguel Neves Gonçalves Madeira

Mestre em Engenharia Informática

Model-driven Personalisation of Human-Computer Interaction across Ubiquitous Computing Applications

Dissertação para obtenção do Grau de Doutor em Informática

Orientador:	Prof. Doutor Nuno Manuel Robalo Correia, Professor Catedrático, FCT/UNL
Co-orientador:	Prof. Doutor Octavian Adrian Postolache, Professor Auxiliar, ISCTE/IUL

	Júri:
Presidente:	Prof. Doutor Luís Manuel Marques da Costa Caires
Arguentes:	Prof. Doutora Carmen Santoro
	Prof. Doutor Rui João Peixoto José
Vogais:	Prof. Doutora Ana Paula Pereira Afonso
	Prof. Doutora Sofia Carmen Faria Maia Cavaco
	Prof. Doutor Nuno Manuel Robalo Correia

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Dezembro 2017

Model-driven Personalisation of Human-Computer Interaction across Ubiquitous Computing Applications

To my grandparents. Miss you so much!

Acknowledgements

First of all, I would like to express my sincerest gratitude to my supervisors. Once more, I had the honour of having Professor Nuno Correia as supervisor, continuing the friendship-based guidance and the transmission of knowledge that started more than ten years ago with the master's work supervision. I also had the pleasure of having the supervision of Professor Octavian Postolache, who embraced this work with all the energy and availability, always bringing interesting insights, relevant expertise and great support.

My acknowledgements extend to the members of the thesis advisory committee, professors Ana Paula Afonso and Sofia Cavaco. Their feedback and expert comments were very important to direct this research work at some important stages along the way.

A special thank you goes to all my family for the love and unconditional support. Particularly, my parents, for everything and much more as I am what I am because of them, in all senses. My wife, for still "being here" after all this time, finding the needed patience to cope with my "moments of absence". Tobias, my faithful companion, for putting a smile on my face and warming up my heart (and feet) every time I needed it. Moreover, Legolas, for brightening up my grey days of the last months of the dissertation's writing. Love you all!

I would like to thank all of those that by some means contributed to the development of this work. Since I do not want to forget anyone, I will not refer any names, besides one, and I will thank everyone personally. Nonetheless, a special thank you goes to Pedro Albuquerque Santos since he is the one who worked more closely with me in this research work, always making me go a little further. Without him, this particular research work would not be possible.

Furthermore, I thank all my colleagues, students and friends for all the support, collaboration and friendship.

As a final point, it is also important to mention the different funding sources involved in this research work:

- The development of this work was partially subsided by *Fundação para a Ciência e Tecnologia* (Portuguese *Science and Technology Foundation*) within the scholarship reference SFRH/PROTEC/50203/2009. This grant was set initially under the PROTEC program, an initiative to promote Ph.D. formation for the teaching staff of Polytechnic Institutes, in this case, *Instituto Politécnico de Setúbal*.

- The research has also been enclosed by NOVA Laboratory for Computer Science and Informatics, under the grant UID/CEC/04516/2013, former *Center for Informatics and Information Technologies* (CITI), which has also subsidised the work under the grant PEst-OE/EEI/UI0527/2011.

- This thesis benefited of tuition exemption in agreement to the cooperation protocol established between *Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa* (*Faculty of Sciences and Technology, New University of Lisbon*) and *Instituto Politécnico de Setúbal*.

Abstract

Personalisation is essential to Ubiquitous Computing (Ubicomp), which focuses on a human-centred paradigm aiming to provide interaction with adaptive content, services, and interfaces towards each one of its users, according to the context of the applications' scenarios. However, the provision of that appropriated personalised interaction is a true challenge due to different reasons, such as the user interests, heterogeneous environments and devices, dynamic user behaviour and data capture.

This dissertation focuses on a model-driven personalisation solution that has the main goal of facilitating the implementation of a personalised human-computer interaction across different Ubicomp scenarios and applications. The research reported here investigates how a generic and interoperable model for personalisation can be used, shared and processed by different applications, among diverse devices, and across different scenarios, studying how it can enrich human-computer interaction. The research started by the definition of a consistent user model with the integration of context to end in a pervasive model for the definition of personalisations across different applications.

Besides the model proposal, the other key contributions within the solution are the modelling framework, which encapsulates the model and integrates the user profiling module, and a cloud-based platform to pervasively support developers in the implementation of personalisation across different applications and scenarios. This platform provides tools to put end users in control of their data and to support developers through web services based operations implemented on top of a personalisation API, which can also be used independently of the platform for testing purposes, for instance.

Several Ubicomp applications prototypes were designed and used to evaluate, at different phases, both the solution as a whole and each one of its components. Some were specially created with the goal of evaluating specific research questions of this work. Others were being developed with a purpose other than for personalisation evaluation, but they ended up as personalised prototypes to better address their initial goals. The process of applying the personalisation model to the design of the latter should also work as a proof of concept on the developer side.

On the one hand, developers have been probed with the implementation of personalised applications using the proposed solution, or a part of it, to assess how it works and can help them. The usage of our solution by developers was also important to assess how the model and the platform respond to the developers' needs. On the other hand, some prototypes that implement our model-driven personalisation solution have been selected for end user evaluation. Usually, user testing was conducted at two different stages of the development, using: (1) a non-personalised version; (2) the final personalised version. This procedure allowed us to assess if personalisation improved the human-computer interaction. The first stage was also important to know who were the end users and gather interaction data to come up with personalisation proposals for each prototype. Globally, the results of both developers and end users tests were very positive.

Finally, this dissertation proposes further work, which is already ongoing, related to the study of a methodology to the implementation and evaluation of personalised applications, supported by the development of three mobile health applications for rehabilitation.

Keywords: Personalisation, Adaptation, Model-driven engineering, User modelling, Ubiquitous computing, Mobile computing, Cloud-based platform, Machine learning, HCI, Human factors.

Sumário

A personalização é essencial para a Computação Ubíqua (Ubicomp), que se foca num paradigma centrado no ser humano, com o objetivo de proporcionar uma interação adaptativa com conteúdo, serviços e interfaces apropriadas para cada um de seus utilizadores, de acordo com o contexto dos cenários das aplicações. No entanto, a disponibilização dessa interação personalizada é um verdadeiro desafio devido a diferentes razões, tais como, os interesses do utilizador, os ambientes e dispositivos heterogêneos, o comportamento dinâmico do utilizador e a captura de dados.

Esta dissertação centra-se numa solução de personalização guiada por um modelo que tem o objetivo principal de facilitar a implementação de uma interação pessoa-máquina personalizada transversal a diferentes cenários de aplicações Ubicomp. O trabalho aqui relatado investiga como um modelo genérico e interoperável para personalização pode ser usado, partilhado e processado por diferentes aplicações, entre diversos dispositivos, e ao longo de diferentes cenários, estudando como pode enriquecer a interação pessoa-máquina. A investigação começou pela definição de um modelo de utilizador consistente, com integração do contexto, para culminar num modelo genérico para a definição de personalizações em diferentes aplicações.

Além do modelo, as outras contribuições-chave dentro da solução são a platforma de modelação, que encapsula o modelo e integra o módulo de aprendizagem automática, e uma plataforma baseada na nuvem para suportar os programadores na implementação da personalização em diferentes aplicações e cenários. Esta plataforma fornece ferramentas que colocam os utilizadores finais no controlo dos seus dados e suportam os programadores através de serviços Web para personalização.

Foram utilizados diversos protótipos de aplicações Ubicomp para avaliar a solução como um todo e cada um dos seus componentes separadamente. Alguns foram criados com o objetivo de avaliar questões de investigação. Outros estavam sendo desenvolvidos com um outro propósito, terminando como protótipos personalizados para endereçar melhor os seus objetivos. O processo de aplicação do modelo de personalização no desenvolvimento destes também deve funcionar como prova de conceito no lado do programador.

Por um lado, envolveram-se programadores na implementação de aplicações personalizadas usando a solução proposta de modo a avaliar a mesma. O uso da solução pelos programadores também foi importante para avaliar como o modelo e a plataforma respondem às necessidades dos programadores. Por outro lado, alguns protótipos foram selecionados para avaliação por utilizadores finais. Usualmente, os testes com estes foram conduzidos em dois estágios diferentes de desenvolvimento, usando: (1) uma versão não personalizada; (2) a versão final personalizada. Este procedimento permitiu avaliar se a personalização poderá melhorar a interação pessoa-máquina. Globalmente, os resultados de ambos os tipos de testes foram muito positivos.

Por fim, esta dissertação propõe como trabalho futuro, já em curso, o estudo de uma metodologia para a implementação e avaliação de aplicações personalizadas, suportado pelo desenvolvimento de três aplicações móveis de saúde para reabilitação.

Palavras-chave: Personalização, Adaptação, Desenvolvimento guiado por modelos, Modelação de utilizador, Computação ubíqua, Computação móvel, Plataforma baseada na nuvem, Aprendizagem automática, IPM, Fatores humanos.

Contents

List of Figures

List of Tables

Acronyms

AL	after lunch
ACR	Automatic Content Recognition
API	Application Programming Interface
BL	before lunch
BLE	Bluetooth Low Energy
CDM	Cheater-Detection Module
CP	Cerebral Palsy
CPM	Cheating-Penalisation Module
DSL	Domain-Specific Language
EP	Evolutionary Psychology
EU	European Union
EUP	End User Programming
GPL	General-purpose Programming Languages
GPS	Global Positioning System
HCD	Human-Centred Design
HCI	Human-Computer Interaction
HTTP	Hypertext Transfer Protocol
IaaS	Infrastructure as a Service
IDMS	Inter-Destination Media Synchronisation
IoT	Internet of Things
IPM	Interação Pessoa-Máquina
IS	Information System
JPA	Java Persistence API
jPk	just Physio Kidding
JSON	JavaScript Object Notation
KVP	Key-Value Pairs

LEY	Less energy Empowers You
MVC	Model-view-controller
NIST	National Institute of Standards and Technology
NUI	Natural User Interfaces
PaaS	Platform as a Service
PD	Public Display
PHP	PHP: Hypertext Preprocessor
PIR	Passive Infrared (Sensor)
PJB	PHP/Java Bridge
PUI	Personalised User Interfaces
PVR	Personal Video Recorder
PWS	People Who Stutter
REST	Representational State Transfer
RFID	Radio-Frequency IDentification
RPC	Remote Procedure Call
RS	Recommender Systems
SaaS	Software as a Service
SDS	Semantic Differential Scales
SMUF	Synchronisation Mechanism through User Feedback
SOAP	Simple Object Access Protocol
TCP	Transmission Control Protocol
Ubicomp	Ubiquitous Computing
UCD	User-Centred Design
UDD	User-Driven Development
UI	User Interface
UM	User Modelling
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VR	Virtual Reality

W3C	World Wide Web Consortium
WS	Web Services
WSDL	Web Services Description Language
XML	Extensive Mark-up Language

1. Introduction

"A relaxed mind is a creative mind."

(Yogi Teas, 2017)[1]

Nowadays, mobile and ubiquitous devices can be found everywhere around people. We are «living in a ubiquitous world» as we face widespread ubiquitous computing (Ubicomp) due to the billions of devices shaping the very fabric of an active world (Rodden, 2008).

The main goal of Weiser's vision of Ubicomp, also called pervasive computing (Satyanarayanan, 2001), is the deployment of rich computational infrastructures to automatically and unobtrusively support and facilitate the everyday life of humans (Weiser, 1991). In the last decades, the development of viable Ubicomp solutions was realistically made possible due to some important technological advances, such as: wireless (data) communications infrastructures providing higher bandwidth, lower power consumption and commodity (readily available and secure); the small form factor of devices based on ever-shrinking electronics; novel interaction methods based on natural user interfaces (NUI); and enhanced displays. Automatic identification (e.g., RFID), sensing and actuation (e.g., bio-sensing) and context awareness are others factors that were important for achieving the Ubicomp's goal.

Many researchers have been excited with the vision of a world augmented by interconnected computational devices that understand and respond to human activities and actions (Rodden, 2008). Abowd considers that the Ubicomp vision has succeeded in pervading the thoughts of a large community of researchers, becoming broader and broader which makes any attempt to maintain a well-scoped research on the topic increasingly difficult (Abowd, 2012). Therefore, Ubicomp had a profound impact becoming now indistinguishable from computing itself (Abowd, 2012). The world in which implicit and natural interactions between humans and computers occur in a natural way across multiple interfaces in diverse environments correspond now to the vision of Ubicomp (Salber et al., 1998; Weiser, 1991). Motion is an integral part of everyday life, so Ubicomp must support mobility or, otherwise, for instance, a person would be acutely aware of the technology by its absence when s/he moves (Satyanarayanan, 2001).

1.1. Motivation

Ubicomp is nowadays especially supported by mobile and context-aware computing. Smartphones, tablets, and other devices are good examples of how technology has evolved mainly in the last decade. The number of increasingly faster and more efficient mobile and wearable devices is continuously increasing, which also explains the consolidation of Ubicomp. Mobile devices have the great potential to be used as the default physical interfaces of Ubicomp due to their advanced capabilities (Ballagas et al., 2006). These devices became common in everyday life of people in most countries, even in the developing ones. People are increasingly able to access the desired content since they are being used

[1] I drank many teas from Yogi Tea® (https://www.yogitea.com) while doing this research work, and mainly when I was writing related papers and this dissertation. The little quotes on their tea bags helped me a lot during this stressful period reminding me of what life is all about.

almost anywhere on a day-to-day basis. Thus, they have changed the way people interact with information. Indeed, as Figure 1.1 illustrates, mobile is changing the world.

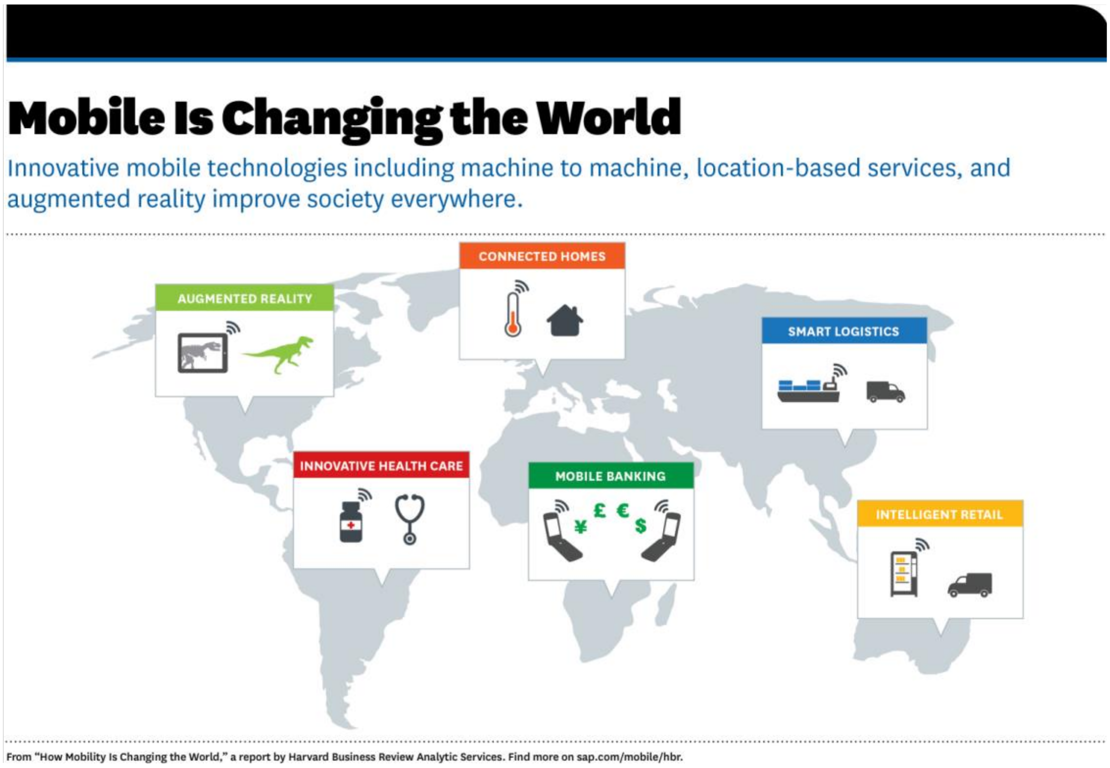


Figure 1.1 – Mobile is changing the world (HBR, 2012).

With Ubicomp, all the information becomes accessible from just about anywhere, at any time, and is delivered in a manner appropriate to our location and context (Greenfield, 2006). The paradigm aims to provide the right information and services in context to the end users of its environments and applications, according to their needs and preferences. It is desired that a (ubiquitous) computing environment proactively reacts to individuals who inhabit it. The vision of Weiser for Ubicomp was then human-centred (Dix et al., 2003; Yu et al., 2011; York and Pendharkar, 2004). Mobile devices present a very personal condition reinforcing this human-centred principle, in which the user is placed at the centre of a new way of understanding and designing computer systems (Chalmers et al., 2006). Due to their mobility, people can use the mobile devices anywhere and anytime, which makes them more personal than, for instance, a normal desktop workstation. While desktop computing presents a static nature, mobile computing is more prone to location variance and therefore, mostly used by moving users (Panayiotou et al., 2006). In 2001, Satyanarayanan stated that "the research agenda of pervasive computing subsumes that of mobile computing," but he added four additional research thrusts into the agenda: (1) effective use of smart spaces; (2) invisibility; (3) localised scalability; and (4) techniques for masking uneven conditioning of environments (Satyanarayanan, 2001). The last one focuses on reducing the amount of variation regarding the smartness seen by a user across different Ubicomp environments, proposing a user's personal computing space to compensate for "dumb" environments (Satyanarayanan, 2001).

Therefore, this human-centred vision demands personalisation and ubiquitous user modelling (Berkovsky et al., 2010). Adaptation and personalisation technologies must be a major component in a Ubicomp solution (Satyanarayanan, 2001; Yu et al., 2011). Even little personalisation can go a long way towards improving important aspects such as user-friendliness, efficiency, and capabilities of shared devices, contributing to getting closer to the goal of Ubicomp (Hilbert and Trevor, 2004).

Several empirical studies demonstrated in different application areas that well-designed adaptive user interfaces could give users considerable benefits (Kobsa, 2004). One of the main reasons for the increasing importance of personalisation, or adaptation, is that people interact with applications in different contexts of use, which are more and more varied because of the advent of those mobile technologies and smart environments (Paterno, 2014). Adaptation has been previously considered as a useful means of overcoming the limitations of the mobile devices (Al-bar and Wakeman, 2001), which are resource-constrained platforms. Increasingly, it becomes essential that applications adapt to the users, rather than forcing users to adapt to the several applications with which they interact simultaneously.

Moreover, these smart (ubiquitous) computing environments also lead to the information overload problem (Maes, 1994) since the quantity of available information rapidly grows and exceeds current processing capabilities (Berkovsky, 2005). This information overload issue can be overcome by the use of personalisation techniques that can filter irrelevant information reaching the user (Berkovsky, 2005). Therefore, this rapidly growing amount of information and services available to people demands proper user modelling and personalisation in Ubicomp environments (Berkovsky et al., 2010).

1.2. Research Problem

The process of obtaining and choosing the appropriate content, interfaces, or services, for each user in many applications from diverse domains in Ubicomp scenarios is still a critical challenge (Gorgoglione et al., 2006). It requires more than wirelessly networked computers, displays, sensors and devices working together, as it relies implicitly on some recommendation mechanism to directly serve the individual. The offer of the right personalised experience is a challenge due to various issues, such as user interests, heterogeneous environments and devices, ubiquitous data capture, dynamic user behaviour and user privacy (Yu et al., 2011). Some of the more important specific challenges are (Berkovsky et al., 2010): inferring relevant information about a user from sensors; aggregating and integrating such information effectively into long-term user models, while achieving user model interoperability; and providing ubiquitous information access in a personalised manner.

Moreover, there are many proposals related to personalisation, but a great number of them are much focused on the specificities of the final product, the application, system or the Ubicomp environment, not offering generic and global solutions (e.g., (Yan and Chen, 2011) and (Geleijnse et al., 2011)). Many others, such as the work of (Jeh and Widom, 2003), are not really directed to Ubicomp scenarios. Indeed, personalisation approaches have been quite isolated and partially demonstrated without considering maintaining the consistency of personalised interaction controls and conditions across the multiple environments and scenarios with which a user interacts (O'Mahony et al., 2012). Since Ubicomp implementations are proliferating and presenting great diversity across many locations, it is important to take into account the holistic view of Ubicomp, which spans technology, use, and users, in which the unit of design should be social people, in their environment (Chalmers et al., 2006).

However, usually, the Ubicomp environments and applications do not share a common knowledge and view of the users. This fact limits developers not allowing them to fine-tune the user experience of the applications they create, meaning that the users are not getting what they exactly want and need. The current state of models, tools, and technology for personalisation does not really help developers in implementing personalised solutions, simply and easily, with a shared knowledge of users across diverse environments.

1.3. Research Goal and Questions

The main goal of this research is the proposal of a generic, interoperable and pervasive model provided by a cloud-based platform for personalisation across different Ubicomp applications.

Therefore, we envision a Ubicomp based world in which a personalisation model can and should be applied to different applications or systems, even if from different domains. The model should appear as the guiding basis of a cloud-based platform solution, which should provide the tools, services, and partners to help developers in the implementation of personalised solutions. On the other hand, users would be less bothered when starting to use a new system or application that needs "to know them". Based on data from the generic global model, the new application already might know something about the users to automatically adapt itself to the users' preferences and needs. A user's interactions with a previous system or application might be useful to personalise the experience of the same user when interacting with a new one. This scenario should be possible due to the use of that pervasive model that promotes interoperability and should guide the developers in the implementation of both solutions.

The main research questions that appear from the work goal are the following:

[$\mathbf{RQ_A}$] *Is it possible to have a generic model to be followed in the personalisation of different applications, or systems, across different domains in a Ubicomp based world?*

[$\mathbf{RQ_B}$] *How can user modelling and context modelling be integrated into a generic pervasive personalisation model to facilitate the implementation of personalised Ubicomp applications?*

[$\mathbf{RQ_C}$] *Is the user interaction with an application useful to personalise the experience of the same user when interacting with another application?*

[$\mathbf{RQ_D}$] *Can a cloud-based solution be easy to use and be developer-friendly in the support of the rapid implementation of generic personalisation across different Ubicomp applications?*

After presenting the research questions, we can summarise the purpose of this thesis in the following research statement. "*In order to provide developers with a set of easy to use and user-friendly guidelines and tools for applying personalisation to a wide spectrum of applications, we propose a model-driven and cloud-based platform targeted to the rapid development of personalised Ubicomp applications*". The platform should be simple to use while encapsulating the important mechanism of user modelling, allowing user interactions control and user preferences to be consistent and memorable across multiple Ubicomp applications.

1.4. Research Plan

The work hereby described followed a plan that was decomposed into six main stages, each one of them including several tasks. A summary of each stage is presented next.

Stage 1 – Research.

This first stage was initiated at the beginning of this research work and was continuously updated considering new proposals and the directions followed by the work. Albeit this, the main research effort took place in the beginning, spanning the first two years with special focus on the model, the user profiling module and the platform for personalisation of Ubicomp applications.

Stage 2 – Experimentation.

This stage started at the beginning of the research work, but most of it was done in parallel with the design of the prototypes in order to help understand the usage of personalisation aspects, being mostly concluded by the end of the second year. However, some experimentation was done after that in order to finish the deployment of the last prototype.

Stage 3 – Prototypes implementation.

It was dedicated to the implementation of several proof-of-concept prototypes that tested most of the concepts presented in this work. It began in the first year with the analysis phase in order to evaluate which projects and applications would be viable options for this research. The second year was particularly devoted to the design phase and the development of the prototypes, whilst the implementation of the final working prototypes with personalisation applied was left for the third year. Some of this work was done in parallel with the final thesis writing.

Stage 4 – Validation.

The work has been continuously assessed and discussed, beginning in the second year with the first results of the research and getting special focus on the third year, and even in the final year, with the end user tests of the prototypes.

Stage 5 – Peer-reviewing.

The work has regularly been submitted to international conferences for peer-reviewing after each milestone of results gathering and discussion. The documentation of the work based on peer-reviewing papers has occurred regularly across the thesis, reflecting the obtained results, even if preliminary. The submission to international journals was expected to happen at the end of the thesis, which is already planned.

Stage 6 – Final thesis writing.

The writing of the thesis has occupied most of the time of the final year.

1.5. Contributions

The research presented here consists of the accumulated expertise summed up in the following contributions to the scientific community:

- A study on how a generic model for personalisation can be defined, made available and distributed among devices, shared and instantiated by Ubicomp applications;
- The personalisation model proposal, integrating ubiquitous user and context sub-models, with the main goal of facilitating the implementation of personalised Ubicomp applications;
- An XML-based Domain-Specific Language (DSL) that is the standard format for the definition of the rules specifying the instantiation of a Ubicomp application's personalisations;
- A personalisation modelling framework in the form of personalisation API[2]s and configuration modules;
- A cloud-based platform to better support and help developers in the implementation of personalised Ubicomp systems;
- The implementation of prototypes, based on the model and connected to the platform, which works as a proof of feasibility of the proposed solution;

[2] In computer programming, an *Application Programming Interface* (API) can be a set of subroutine definitions, protocols, or tools for developing application software.

- Human factors studies in the design of personalised solutions for different domains.

1.6. Publications

During this research work, the obtained results were materialised into the following publications, grouped into four subjects according to the topics addressed by them in the research:

Personalisation Solution

- *Building a Platform for Pervasive Personalisation in a Ubiquitous Computing World* (Madeira, Santos and Correia, 2014), which presented and summarised the cloud-based platform.

- *Model-Based Solution for Personalisation of the User Interaction in Ubiquitous Computing* (Madeira, Santos, Vieira, et al., 2014), which presented the full solution focusing a little more on the model. Developers' tests with two prototypes were also presented, demonstrating how the solution was used in their personalisations.

- *Personalisation in Pervasive Spaces towards Smart Interactions Design* (Madeira, 2012), where preliminary work on the research motivation, problem, goal, and plan, besides the initial studies, were presented for discussion at IEEE PerCom 2012's Doctoral Consortium.

- *An Infrastructure for Real Objects Augmentation with Additional Personalised Information Services* (Madeira and Correia, 2010), which focused on previous work that presented initial ideas for personalisation of the interaction integrated into an infrastructure for the augmentation of Ubicomp scenarios.

Prototypes Implementation

- *just Physio kidding - NUI and Gamification based Therapeutic Intervention for Children with Special Needs* (Madeira et al., 2016), which introduced a new solution based on PhysioMate, but now directed to the rehabilitation of children with special needs. This version intensifies the personalisation regarding gaming elements.

- *Synchronising Live Second Screen Applications with TV Broadcasts Through User Feedback* (Centieiro, Romão, et al., 2015b) and *A Synchronisation Mechanism based on User Feedback for Second Screen Applications* (Centieiro, Romão, et al., 2015a), presented a mobile second screen app that uses a user feedback based synchronization mechanism.

- *PhysioMate - Pervasive Physical Rehabilitation based on NUI and Gamification* (Madeira, Luís Costa and Postolache, 2014) presented the design of a pervasive computing solution mainly focused on the rehabilitation of the upper-limb of stroke patients on wheelchairs.

- *Gaming for Therapy in a Healthcare Smart Ambient* (Madeira, Postolache, et al., 2012) addressed the inclusion of gamification elements in the previous environment.

- *LEY! Persuasive Pervasive Gaming on Domestic Energy Consumption-Awareness* (Madeira, Silva, et al., 2011), which introduced the first prototype used in the evaluation of the personalisation model by developers. LEY was developed for mobile devices and integrated a sensor platform for wireless home real-time energy monitoring.

- *Designing a Pervasive Healthcare Assistive Environment for the Elderly* (Madeira et al., 2010) presented the healthcare environment design in which the PhysioMate system would be integrated.

Personalisation of the Prototypes and Human Factors Studies

- *Serious…ly! Just Kidding in Personalized Therapy through Natural Interactions with Games* (Madeira, Antunes, et al., 2017) is the full paper that presents the aforementioned *just Physio kidding* prototype in more detail, giving particular attention to the first study with experts (the healthcare professionals), in which the need for personalisation, at least in systems for health rehabilitation, became evident.
- *Mobile Apps to improve ThErapy - The Health Practitioner in your pocket knows you* (Madeira, Macedo, et al., 2017), which is a paper presenting a project directed to the development and assessment of personalised mHealth applications that use the personalisation platform and follow the design guidelines taken from our studies.
- *Adaptation to TV Delays Based on the User Behaviour towards a Cheating-Free Second Screen Entertainment* (Madeira et al., 2015) and *In sync with fair play!: delivering a synchronized and cheat-preventing second screen gaming experience* (Centieiro, Madeira, et al., 2015) present, respectively, an intermediate stage and the full solution of the WeSync prototype, already showing results obtained with the implementation of personalisation to deal with possible cheating behaviours.
- *FCT4U -- When Private Mobile Displays Meet Public Situated Displays to Enhance the User Experience* (Santos et al., 2013) which presented a system designed as a mobile and ubiquitous distributed system for the university campus. It also addressed the prototype's personalisation, which was important as a demonstrator for the inclusion of context in the personalisation process.
- *Personalisation of an Energy Awareness Pervasive Game* (Madeira, Vieira, et al., 2012), where the initial personalisation design of LEY was addressed.
- *Towards Personalised Pervasive Theragames in Smart Wheelchairs* (Madeira and Correia, 2011), which addressed studies for the integration of personalised serious games in the environment of the PhysioMate prototype.
- *Designing Personalised Therapeutic Serious Games for a Pervasive Assistive Environment* (Madeira, Correia, et al., 2011), which presented a more thorough study than the previous publication and already addressed the personalisation design of a gamification-based application for the wheelchair system.

Related Projects in which the Research Work was being Studied and Applied

- *Super-Fon: Mobile Entertainment to Combat Phonological Disorders in Children* (Madeira, Patrícia Macedo, Reis, et al., 2014) and *Phonological Disorders in Children? Design and user experience evaluation of a mobile serious game approach* (Madeira, Mestre, et al., 2017), which presented a mobile healthcare prototype using gamification to address phonological disorders in children.
- *Mobile Integrated Assistance to Empower People Coping with Parkinson's Disease* (Pereira et al., 2015) and *ONParkinson – Innovative mHealth to support the triad: patient, carer and health professional* (Pereira et al., 2016) presented the work conducted by an interdisciplinary team to create a mobile healthcare solution to help not only the patient, but mainly carers and healthcare professionals, to cope with the Parkinson's Disease.
- *Building on Mobile towards Better Stuttering Awareness to Improve Speech Therapy* (Madeira et al., 2013) and *The Impact of Stuttering;: How Can a Mobile App Help?* (Demarin et al., 2015) are two papers about a mobile healthcare prototype that addresses the stuttering pathology.

1.7. Dissertation Outline

This document contains five more chapters, structured as follows:

- **Chapter 2 (Research Context)** presents the concept of personalisation and describes the main notions of the background in which this research work is inserted.
- **Chapter 3 (A Model-driven Personalisation Solution)** focuses on the solution for personalisation of ubiquitous computing based applications, introducing the X-Users model, the CAPE framework in which the model is integrated, and the cloud-based platform that provides the framework for all developers and end users of the applications.
- **Chapter 4 (Experimenting the Personalisation Solution)** presents the conducted studies with developers to respond to the research questions raised previously. We have been involved in the development of different prototypes that worked as the basis for the study cases throughout the different phases of the evaluation process. This chapter presents the two studies that were used mainly to evaluate the personalisation solution by developers.
- **Chapter 5 (Studying the Personalisation Solution)** presents the studies conducted with the involvement of end users to respond to research questions regarding the user experience side, mainly if the personalised prototypes deliver value to the end users. Once again, the involvement in the development of several prototypes allowed us to design different study cases throughout the whole evaluation process.
- **Chapter 6 (Conclusions and Research Status)** summarises the research work in the most relevant conclusions and presents the current research status and future work.

2. Related Research

"When ego is lost, then universality exists."

(Yogi Teas, 2017)

This chapter presents background and state of the art related to the main topics focused on this research work. It starts by introducing what means personalisation in the scope of this thesis (in section 2.1), presents the major topics to take into account in what we call the personalisation modelling process (in section 2.2), introduces the fundamentals of web services and cloud computing (in section 2.3) and, finally, describes a set of projects related to our research work (in section 2.4).

2.1. The Personalisation Concept

Personalisation has enjoyed a revival since the year of 2000, having a rich history in computer science crossing fields like human-computer interaction, computer-supported collaborative work, operating systems, or the World Wide Web (Hilbert and Trevor, 2004). Therefore, personalisation is now increasingly expected, but so many people use the word in so many ways that we must first make sure we are using the right meaning in the context of the current research. Many times, people use the words personalisation and customisation interchangeably. However, for the area of our research, there is a subtle difference between the two, which has a great impact on the real purpose of developing personalised computational systems. Customisation means that the user explicitly states interests and preferences through direct configuration of human-computer interfaces, system's options or screens. On the other hand, personalisation should be implicit and automatic, possibly resulting in a "Wow" moment when the user feels that things appear tailored to her/his needs and preferences. "Personalisation goes a step beyond customisation in that it does not just affect the way things look to users, it also is the way things feel to users" (Even, 2015). Customisation is considered user-led which results in an adaptable system, whereas personalisation is system-led resulting in an adaptive system (Oppermann et al., 1998).

Personalisation dynamically responds to users' demands automatically adapting the items based on their preferences, their past use or even their location. Therefore, personalisation can be seen as a method that provides users with what they want, or need, without requiring them to ask for it explicitly (Mulvenna et al., 2000). Personalisation is a special form of adaptation which generally focuses on making an application more receptive to the unique and individual needs of each user (Cingil et al., 2000). It is still about the adaptation of a system's look and behaviour but focused on the personal profiles of the system's users. An important and well-known definition for adaptive user interface is (Langley, 1999): "An adaptive user interface is a software artefact that improves its ability to interact with a user by constructing a user model based on partial experience with that user.". Adaptive or personalised user interfaces (PUI) should use intelligent mechanisms to adapt the behaviour automatically and accordingly to the user characteristics, preferences, needs or abilities. Users are different and therefore they have different needs from an interactive system. The system should adapt to the user, instead of expecting the user to adapt to the system.

Oppermann identified the following three high-level components in a PUI (Oppermann, 1994):

1) *Afferential* - Observes and records user behaviour and system reactions;
2) *Inferential* - Analyses the gathered data to draw conclusions and decide how the system should adapt to the User Model;
3) *Efferential* - Adaptations lead to modifications of how the system behaves.

Moreover, according to (Brusilovsky and Maybury, 2002), a system acts at three stages during the process of personalisation in order to control:

1) the process of collecting data about the user,
2) the process of building up the user model (user modelling),
3) the final adaptation procedure.

The whole process of the personalisation cycle comprises these iterative stages, but it also should include measuring impacts on the users and adjusting all the previous stages based on the measured feedback (Mobasher and Tuzhilin, 2008). Regarding Ubicomp, according to (Satyanarayanan, 2001), adaptation is particularly necessary when there is a "significant mismatch between the supply and demand of a resource" (e.g., wireless network bandwidth, memory, user interfaces and content). The author identified three important strategies for adaptation:

1) the applications can be guided to change their behaviour to use less of a scarce resource;
2) the environment (Ubicomp space) can be asked by the clients' applications to guarantee a certain level of a resource;
3) the application can suggest corrective actions to the user so that the resource supply can become adequate to meet demand.

2.2. Personalisation Modelling

This section addresses topics that are important to understanding better the modelling process of personalisation towards the solution we want to provide.

2.2.1. Personas towards a User-Centred Design

Personalisation can be best viewed as an evolving set of tools (technologies and applications features) used in the design of the best possible end user experience (Kramer et al., 2000). Moreover, it is important to note that the use of personalisation might result in decreased value to the final user (Kramer et al., 2000). Software engineers and designers, who build a personalised application, are poised to make the classic error of putting technology before the needs of the end users and personalisation can take a back seat to end-user value (Kramer et al., 2000). They need to understand the capabilities, limitations, and end-user costs of the tools used by them (Kramer et al., 2000). Many approaches for software systems design do not adequately capture the needs and values of end users since organizations fail to consider the user or consumer needs as the focal point of their design processes (Gulliksen et al., 2003). It is important to follow design approaches and methodologies that are appropriate for helping to gather relevant information about the end users in order to design the best possible personalised systems and applications.

The usability of websites, systems, and many products has improved alongside with the growing popularity of User-Centred Design (UCD) (Vredenburg et al., 2002). Understanding the users is the core principle of UCD. Its methodologies maintain a strong focus on bringing value to the end user. Actually, UCD represents a general philosophy that brings the users into the design process (Veryzer and Borja de Mozota, 2005). The UCD approach, also known as human-centred design (HCD) and user-driven development (UDD) or even customer-centred design, focuses on a development driven by the user's requirements instead of the usual technical requirements. The design should depend on measures related to how well the product can be used by highlighting the user's interaction and by using an iterative prototyping development process (Junior and Filgueiras, 2005).

However, the widespread acceptance of user-centred principles has not eliminated the frustration that often appears with the design and development of new solutions and products (Miaskiewicz and Kozar, 2011) and, along the years, many design processes had difficulties in reaching their target, the end users. In a UCD approach, in which the user is in the centre of the application design, the user modelling process must identify and analyse the end users. There are several modelling techniques (Junior and Filgueiras, 2005), such as, user roles, user profiles, user segments, marketing segments, extreme characters, personas, and non-user personas. Moreover, the three common representations of user data in forms that are useful to designers are personas, scenarios, and, more recently, user stories (Turner et al., 2013). From all the techniques and tools related to design process and user experience, only personas appear as a consistent common denominator (Churruca, 2013). Most of the designers agree that the secret of a great user experience strategy is supported by the use of personas; even so, the building process varies significantly.

Personas stand out with the potential to help address some of the issues with recent user-centred approaches, existing many benefits in incorporating them into UCD approaches (Miaskiewicz and Kozar, 2011). A consistent approach should be defined to obtain a product (e.g., a personalised application) that is user-oriented, meeting the user goals, so user personas have the potential to help achieve the adage fundamental to UCD: 'Know thy user, for she is not you' (Miaskiewicz and Kozar, 2011). The utilisation of personas was already booming more than a decade ago, being a popular way to customise, incorporate and share the research about users (Junior and Filgueiras, 2005).

The term "persona" comes from the Latin *dramatis personae* (the characters of the story). Alan Cooper popularised it as a means of clarifying the audience for a software application, helping a product's developers team (especially engineers) understand they may not be the audience (Cooper, 1999). Personas are described in a format that is meaningful and create user empathy among the development team's members (Adlin and Pruitt, 2010), ensuring their users are always the focus of their efforts. The ultimate purpose of creating personas is to put all stakeholders involved in the project's development into the user's shoes. Thus, the creation of a common understanding of the final user is the most important reason to create personas. They act as a multipurpose tool used to drive many important product development tasks, such as the creation of user scenarios, feature generation, and feature prioritisation (Flaherty and Laubheimer, 2017).

Initially, Cooper treated personas as stereotypes, which are a result of preconceived opinions, a standardised image or idea that is usually oversimplified. Currently, the most followed definition is that "personas are archetypes built to identify our real users profile, needs, wants and expectations in order to design best possible experience for them" (Churruca, 2013). Persona is a representative technique with high impact in associating and representing real users, fulfilling the need of mapping and grouping a relevant number of users (Junior and Filgueiras, 2005). Persona modelling is the process of representing typical users and their goals, gathering realistic representative information about them, which can include additional fictitious data in order to reach a more accurate characterisation (Cooper, 1999; Cooper and Reimann, 2003). User personas are fictitious yet realistic representations of target users.

Personas must be based on ethnographic and demographic data from the users (Cooper, 1999; Cooper and Reimann, 2003). The information that may be taken into account can be personal, technical, relationship, or opinion based. There are many different layouts that may be used to characterise a persona, but (Churruca, 2013) resumes some main elements that are common to the majority of the layouts, such as, a profile area, personality elements, expertise, must does/ must never, referents & influences, user type, archetype, among others. Personas should have names like real people and we can use pictures to represent them, adding realism (see Figure 2.1).

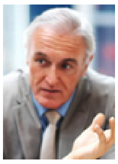
Persona:	USDA Senior Manager Gatekeeper
Photo:	
Fictional name:	Matthew Johnson
Job title/ major responsibilities:	Program Staff Director, USDA
Demographics:	• 51 years old • Married • Father of three children • Grandfather of one child • Has a Ph.D. in Agricultural Economics.
Goals and tasks:	He is focused, goal-oriented within a strong leadership role. One of his concerns is maintaining quality across all output of programs. Spends his work time: • Requesting and reviewing research reports, • preparing memos and briefs for agency heads, and • supervising staff efforts in food safety and inspection.
Environment:	He is comfortable using a computer and refers to himself as an intermediate Internet user. He is connected via a T1 connection at work and dial-up at home. He uses email extensively and uses the web about 1.5 hours during his work day.
Quote:	"Can you get me that staff analysis by Tuesday?"

Figure 2.1 - Persona developed by the U.S. Department of Agriculture's (USDA) Economic Research Service (ERS) (usability.gov, 2013)

A persona is ideally shaped according to user studies, and used for ideation; however, the process varies since sometimes the idea comes first and the persona is created later and used to verify or enrich the idea (Nielsen, 2013; Chang et al., 2008). In UCD processes, the user personas are built following different techniques, such as exhaustive observation and interviews (Churruca, 2013). Usually, personas are created based on the contact with potential end users, but data used to analyse and create the basis for personas can be gathered from a corporation, organisation, or collected by a team responsible for the analysis and modelling phases. After gathering information on users, it is possible to build a set of personas, identifying recurring patterns within the users. It is recommended that a persona definition should not be based on an isolated person. The definition of personas makes assumptions and knowledge about users explicit, creating a common language with which to talk about users meaningfully. It allows designers to focus on and design for a small set of specific users, helping to make better decisions (Adlin and Pruitt, 2010).

(Nielsen, 2013) presented a 10-step process covering the entire process for dealing with personas, from preliminary data collection, through active use, to continued development of personas. It contains four main parts: data collection and analysis of data (steps 1, 2), persona descriptions (steps 4, 5), scenarios for problem analysis and idea development (steps 6, 9), and acceptance from the organization and involvement of the design team (steps 3, 7, 8, 10). It would be the ideal process according to Nielsen, but it is not possible to include all the steps in all the projects that use personas.

Summarising, there are a few common points across all user personas methods that are essential to understand this design tool (Churruca, 2013):

- personas are 'fictional' characters, but they are created based on real data and research around a problem domain, or a focus target;
- a product should have the minimum number of personas to have a more focused design in order to guarantee better success;
- a personas approach must find the user needs, wants and limitations;
- and is more important to be precise than accurate in user personas.

It is important to note that the use of personas for IT system development is mainly to explore interaction and navigation, not being so suited to describing what kind of information the system is to contain (Nielsen, 2013). Personas can be a first step towards personalisation. Jumping straight to the personalisation of an application might seem a step too far in the short term, so it is better to understand its user personas to offer a faster benefit and to serve as a first step to achieving personalisation in the medium term. This way, using personas in the design process would help reveal what is personally meaningful to each targeted end user.

2.2.2. User Modelling

User modelling is an essential mechanism for the support and enhancement of personalisation (Brusilovsky and Maybury, 2002). It is not a recent field of research as it can be traced back to 1979 (Kobsa, 2007), year in which we can find seminal works from Allen, Cohen, and Perrault (Allen, 1979; Cohen and Perrault, 1979), and the relevant work from Elaine Rich (Rich, 1979b; Rich, 1979a). However, its popularity increased with the appearance of the recommender systems. The user modelling process results in the user model, which attempts to gather, for instance, users' characteristics, preferences or interactions patterns. The user model can be defined as an abstract representation containing explicit assumptions on all aspects of the user that may be relevant for the behaviour of the system (a particular perspective) (Zhang and Ghorbani, 2007). PUIs rely on a variety of models to improve their interaction with the user, of which the user model is the most important.

The result of the instantiation of a user model for a particular user is the user profile. The user may have one or several profiles, depending on the perspectives or views that are taken into account. Different profiles may be completely disjoint or may share some attributes. A particular profile corresponds to a particular perspective, which is described as the domain of interest. The user profile is defined as a collection of information about a user, which can combine (Kobsa, 2004; Bouzeghoub and Kostadinov, 2007):

- demographic data (e.g., age, gender, nationality);

- usage/interactions data (e.g., functionalities executed, "skipping" of user interfaces (UI) or streaming media, time spent interacting with a UI);
- information or assumptions about the user's knowledge, skills, capabilities, interests, preferences, goals, and plans;
- environmental data, such as data about the user's software and hardware environments, and data about the user's current location and personalisation-relevant data of this location.

User data can be organised into two types, describing (1) individuals and (2) groups of users. Many times, obtaining the information about the users can be a hard task and several methods can be used to acquire it (Kobsa, 2004). It can be explicitly stated by the user or implicitly derived by the system. Usually, making an initial questionnaire is a simple way to ask users directly. However, the questionnaire must be kept extremely short since people are generally reluctant to spend efforts on work that is not immediately related to their current tasks (Kobsa, 2004). An alternative is the use of smart cards (Kobsa, 2004), having basic information about users, which can be quickly used before the beginning of a computer session, or virtual representation of them on mobile devices that can even be read from a distance as users approach a reader terminal. Moreover, often people are not willing to provide the information because of privacy concerns (Kobsa, 2004).

Nonetheless, there are diverse methods that draw assumptions about the user based on her/his interaction behaviour (Kobsa, 2004). These methods include simple rules that predict user characteristics or assign the users to predetermined user groups with known characteristics when certain user actions are being observed ("stereotype approach") (Kobsa, 2004). Usage information can be tracked from observing users' behaviours. Usually, it can be extracted from a Web server log, but, actually, it can be gathered from any application that stores interactions data logs. Anyway, a good approach to make user modelling simpler and more reliable is always the one that involves the user in the modelling process, persuading and motivating the user to give additional information about her/him.

User modelling is the whole process of incrementally building up a user model and creating, updating or deleting user profiles, maintaining the consistency of the model and supplying assumptions about the user. Therefore, the user model must represent the needed attributes of the user in the application context. A user model should be applied every time an individualised response from a system or application is expected (Koch, 2000). There are several techniques for user modelling and some of them have presented integrated ways to represent the user model. Actually, representation and reasoning in user modelling are tightly coupled (Kay et al., 2012).

Finally, there was a recognition of the link between the goals of Ubicomp and the need to support users in such environments (Kay et al., 2012). Thus, ubiquitous user modelling closely follows the evolution of Ubicomp, corresponding to ongoing modelling and exploitation of the user behaviour with a variety of systems that share their user models (Kuflik et al., 2008). For instance, a mechanism for reusing generic user modelling data with different applications called UM Toolkit was already developed in 1995 (Kay, 1995). A repository of user data was the core of the solution and it was stored in the user's file space. This tool evolved into Personis (Kay et al., 2002), where the main difference is the appearance of a server to store a collection of user models for many users. More focused on Ubicomp was even the development of PersonisAD (Assad et al., 2007), providing distribution of models, where authorised applications could request access to a model in order to use and reuse the model's allowed parts. Berkovsky et al. presented the user model mediation concept (Berkovsky et al., 2007), employing appropriate methods to transform the syntax and semantics of the user model used in one system into those of another system. The work introduced a generic framework. Moreover, (Kabassi et al., 2008) have presented a domain-independent method for user modelling sharing between applications. It is a user modelling server, called UM-Server, which embodies a multi-criteria decision-making theory on the server side and provides services to any application that requests for it.

2.2.3. Machine Learning and the User Modelling Techniques

Applications that provide large amounts of data regarding the users' interaction, such as usage data (e.g. number of clicks, time spent doing something), preferences or even context of usage, should use machine learning to reasoning about the users. It should be used to acquire models of individual users interacting with the information system and grouping them into communities or stereotypes with common interests. Machine learning algorithms can be organised into a taxonomy based on the outcome of the algorithm (Ayodele, 2010): supervised learning; unsupervised learning; semi-supervised learning; reinforcement learning; transduction; and learning to learn. There are several different machine learning techniques for user modelling and to provide personalisation to applications. Usually, the most used techniques are based on clustering, classification and recommender systems (RS).

Clustering algorithms are able to decide on their own which distinctions appear to be important (unsupervised learning algorithms) while classification algorithms learn to mimic examples of correct decisions (supervised learning), which implies the need to have labelled training data. Classification algorithms are also different from recommendation algorithms as they are intended to make a single decision with a limited set of possible outcomes, whereas recommendation algorithms select and rank the best of many possible alternatives and generally are item-oriented. They are commonly used in applications that are based on the notion of items, such as stores that sell products or applications that recommend services. This means that recommendation techniques can be useful for some applications and practically useless (or impossible to implement) for other applications that are designed with different ends.

Clustering or cluster analysis is the organisation of a collection of patterns (usually represented as a vector of measurements, or a point in a multidimensional space) into clusters based on similarity, i.e., dividing data into groups of similar objects. Intuitively, patterns within a cluster are more similar to each other than they are to a pattern that belongs to another cluster. Clustering a collection of items usually involves three entities: an algorithm, a notion of both similarity and dissimilarity and a stopping condition. The algorithm is basically the method that is used to group data into groups of similar objects. In order to organise data into patterns, the second entity is used since data must be compared between each other to predict a level of similarity or dissimilarity. Finally, the algorithm will have to stop eventually. That should happen when it is not possible to move objects around between clusters, i.e., when each cluster is already quite dissimilar with the others, which makes the usage of a stopping condition a requirement. The notion of distance between points is crucial to the success of the clustering operation, and that is why the most important issue in clustering is finding a function that quantifies the distance between any two data points as a number. It is not possible to identify the best distance measure because performance and effectiveness of each one strongly depend on the data that is used in each case.

On the other hand, since finding relevant (personalised) information has become a challenge in an information overload era, RS have emerged to face that problem. A possible definition for RS is: "any system that produces individualised recommendations as output or has the effect of guiding the user in a personalised way to interesting or useful objects in a large space of possible options" (Burke, 2002). Usually, an RS for Web (especially, e-commerce) applications receives, implicitly or explicitly, information about users and their behaviour to recommend items that may suit the users' interests (Sarwar et al., 2000). Based on the techniques that are used, RS solutions often fall into the following categories (Burke, 2000): content-based, collaborative filtering and knowledge-based. The first two are the most reported in the literature.

Usually, content-based filtering means an algorithm based on the attributes of the items and the ratings of the targeted user, with the interpretation of the users' preferences as a function of the attributes. It presents two main methods (Pazzani and Billsus, 2007):

1) **heuristic-based** - use of common techniques of information retrieval;
2) **model-based** - use of a probabilistic model to learn the prediction of users from attributes.

Collaborative (or social) filtering focuses on ratings of items by multiple users. For example, music is recommended to a user if it is highly rated by others who share the user taste in music. Usually, there are three main collaborative methods:

1) **item-based** (Sarwar et al., 2001) - calculate similarity between items and make recommendations (usually, can be computed offline);
2) **user-based** (Resnick et al., 1994) - recommend items by finding similar users, but this is often harder to scale because of the dynamic nature of users;
3) **slope-one** (Lemire and Maclachlan, 2007) - a very fast and simple item-based recommendation approach applicable when users have given ratings (and not just boolean preferences).

On the other hand, knowledge-based requires that knowledge about how a particular object satisfies the user needs and can use this knowledge to search the information space for objects relevant for the user in a particular situation. These systems "do not attempt to build long-term generalisations about their users".

Burke adds even two more categories to the previous set (Burke, 2002): demographic and utility-based. A demographic RS categorise the user, usually by requiring the user to answer a set of questions explicitly, and generate recommendations based on demographic classes. The utility-based systems can be considered a special case of knowledge-based systems where the utility function is defined for each user (automatically or manually). Finally, hybrid recommenders are also usual solutions, which often appear in literature, because the systems above present different weaknesses and different strengths (Burke, 2002). Therefore, it is usual to find combinations to improve their performance.

By the end, the most important aspect is to find the most appropriate technique to the scenario of application that is being modelled for personalisation. Kay et al. have selected a set of techniques for user modelling that should support efficient reasoning about the users, especially in cases in which we start with limited information about them and need to infer more (Kay et al., 2012). The techniques include the following: content-based, case-based, collaborative (or social), demographic, knowledge-based reasoning, and various hybridisations. These techniques differ in their complexity, the amount of information they keep, presenting pros and cons depending on each application.

2.2.4. Context Integration

Ubiquitous user modelling must overcome three major barriers (Kay et al., 2012), which are the following:

1) user modelling issues related to the differences in user modelling techniques;
2) the need to efficiently and effectively bootstrap a user model for ad-hoc services and continuous updating of the user model;

3) domain differences and contextual differences.

The first two generations (pre-Web and The Web) of adaptive systems explored mainly adaptive content selection and adaptive recommendation based on modelling user interests (Brusilovsky and Maybury, 2002). Nowadays, a third generation, much based on the mobile paradigm, extends the basis of the adaptation by adding models of context (e.g., location, time, bandwidth, or computing platform) to the classic user models and explores the use of known adaptation technologies to adapt to both the individual user and the context of the user's work (Brusilovsky and Maybury, 2002). Applying individual user profiles in context requires these profiles to be portable and tied to intelligence, which provides services based on an awareness that the user is not, for example, at home or in their home country (Lewis, 2004). Thus, it is important to add context to the personalisation modelling process, mainly to cope with Ubicomp scenarios.

Context-Aware personalised services towards the user's situation is already a determinant topic in mobile and ubiquitous computing. The term 'Context-aware' has been defined as "systems [that] adapt according to the location of use, the collection of nearby people, hosts, and accessible devices, as well as to changes to such things over time" (Schilit et al., 1994). There are many definitions for the context in the literature, depending on the domain of application. One of the most used definitions was proposed by Abowd et al. (Abowd et al., 1999): "Context is any information that can be used to characterise the situation of an entity." An entity can be a person, object or space and the specific situation can be related, for example, with the current location, season, person role, and/or temperature. Based on the generally adopted original definition for context by (Schilit et al., 1994), Barkhuus defined a very interesting contextual information model (Barkhuus, 2003). This categorisation is based on a user's viewpoint where the model separates strictly between contextual information and sensor information.

Traditional clustering techniques generally anchor in pure content-based analysis and do not consider context. As a consequence, those clustering techniques create a set of clusters that are not tailored to users' current context. With context, a machine learning system perceives a user model as something more dynamic, which is also more realistic. A user on some occasions might behave in a certain way, and in others s/he might have different interests or needs. That is why the integration of context may be relevant, having a significant effect on the prediction accuracy of users' profiles. However, the context should be considered as a separate entity in the data model and not as part of the users' descriptions since there are some substantial differences (Woerndl and Schlichter, 2007). A user profile is rather static and somewhat longer lasting, being implicitly observed or explicitly provided by the user, and presenting a simpler or rawer nature. The context is highly dynamic and transient, not being permanently stored, in general, observed and not inserted by the user (there are exceptions).

Palmisano et al. refer the existence of a trade-off between transaction homogeneity and data sparsity, because the act of providing contextual information will reduce the number of user transactions in a given context (Palmisano et al., 2008). It results in fewer data points to fit the model while the homogeneity of the transactions increases, which grants improved prediction accuracy. This makes context identification crucial for the success of a machine learning tool. The process may be executed in: (1) a manual way, where appropriate personnel (e.g., market experts and system designers) elect relevant context entities, based on their expertise; or (2) an automated way, where some machine learning techniques are applied to the system in order to study the effect of different context elements on the outcome.

Broadly, there are three ways to identify contextual information (Ricci et al., 2010). It can be made:

1) **explicitly**, which consists in directly querying the user or other sources of information about the current access status;

2) **implicitly**, where information is gathered implicitly, i.e., there is no direct interaction with the system's users or sources of context information;

3) **inferring**, where it is possible to infer context using data mining or statistical techniques.

Moreover, the incorporation of context in a machine learning algorithm, usually a recommender system, can take one of three approaches (Adomavicius and Tuzhilin, 2010):

1) **Contextual pre-filtering** - Context information is used to filter out unwanted data before it is used by the machine learning tool;

2) **Contextual post-filtering** - Similar to pre-filtering. Instead of filtering data at the beginning, it is filtered in the end after executing the machine learning tool;

3) **Contextual modelling** - Contextual information is used directly in the modelling technique used by the machine learning tool.

In order to achieve the best results, the three techniques should be implemented and tested to see which ones are the best options according to the application's purposes. This procedure is advisable since their performance is strongly related to the application, its domain, the users and the environment.

2.3. Web Services and Cloud Computing

This section introduces the fundamentals of web services and cloud computing, which are important to contextualise the implementation of the cloud-based platform and its HTTP-based API for personalisation.

2.3.1. Web Services

Web services are defined by W3C[1] (Haas and Brown, 2004) as "a software system designed to support interoperable machine-to-machine interaction over a network." They make available a Web Services Description Language (WSDL) which is an interface described in a machine-processable format. Third-party systems interact with a Web service in a manner prescribed by its description using Simple Object Access Protocol (SOAP) messages, typically conveyed using HTTP with an XML serialisation in conjunction with other Web-related standards. In this work, the adopted definition is more comprehensive, unless when specifically talking about W3C Web Service standards. Moreover, two different classes of Web services can be found (Booth et al., 2004), both using URIs to identify resources and use web protocols and open data formats for messaging and communication. They are the following:

1) **REST-compliant Web services**, in which the primary purpose of the service is to manipulate XML2 or JSON3 representations of Web resources using a uniform set of "stateless" operations;

2) **arbitrary Web services** that expose an arbitrary set of operations.

Web Services, which follow the W3C standards, are usually composed of three distinct entities at a high-level architectural view (Booth et al., 2004; Haas and Brown, 2004):

[1] World Wide Web Consortium - http://www.w3.org/
[2] Extensible Markup Language - https://www.w3.org/TR/2004/REC-xml-20040204/
[3] JavaScript Object Notation - http://www.json.org/

1) **Service provider**: this is the provider of the web service, implementing the service and thus making it available for consumers (clients);
2) **Service requester**: a consumer invokes an existing web service by opening a network connection and sending SOAP messages;
3) **Service registry**: it is a centralised directory of services, which is used as a central place where providers or developers can publish new services and find existing ones.

The web service development can be summarised as follows. First, a programmer creates a web service using a specific programming language and the service is published using WSDL in order to describe its interface, which presents the set of operations provided by the web service. Then, the web service must be deployed into a server container to make it available to consumers. On the client side, an object representing the remote Web services must be generated, allowing clients to call the operations defined on the server side's service interface.

A developer does not have to worry about manipulating SOAP messages, because this is usually done by a library provided by the programming language. Also, as the objective of Web services is to be interoperable, a web service developed using one language and running on a specific operating system can be accessed by a client written in any other language/platform running on any other operating system. Web services are platform-independent and based on structured messages (e.g., XML). The idea is to distribute services over the Internet, making them available for clients. These services can be implemented or accessed through any programming language. The language/platform agnostic characteristics are especially appropriate for this research work, which aims to be widely adopted by developers, regardless of their favourite development tools. Finally, it is possible to summarise some key determining features of Web services (Papazoglou, 2008; tutorialspoint, 2016):

- **XML-based**: Web Services rely on XML for data representation and transportation. The use of XML avoids any network, operating system or platform dependency;
- **Loose coupling**: There is no direct tie between a web service and its users, contrasting with tightly coupled systems, where the client and server logic are closely bound to each other;
- **Ability to be synchronous or asynchronous**: The interaction between consumers and the web service can be made synchronously or asynchronously. Asynchronous invocations allow clients to make a request and then immediately execute other operations without waiting for the result;
- **Supports RPC (Remote Procedure Call)**: Web services enable clients to invoke methods and operations on remote objects using SOAP. Thus basically enabling the remote invocation of any code that is made available through the web service;
- **Enables reusability**: Web services can be remotely accessed, providing a way to make a pre-existing code available through the network. As a result, that code can be used by multiple applications;
- **Interoperability**: Web services enable communication between different applications, running on different operating systems and developed in different programming languages. Thus, it enables interoperability across systems and applications;
- **Standard Formats and Protocols**: Web services use well-defined industry standard protocols and formats across all the technology stack, promoting interoperability This standardisation gives organizations many advantages, like a wide range of choices, reduction of costs due to competition and increase in the quality;
- **Automatic Discovery**: Automatic discovery mechanisms allow to find easily web service descriptions that have been previously published.

2.3.2. Cloud Computing

Multiple authors present definitions for cloud computing while trying to reach a consensual and encompassing one (Vaquero et al., 2008). There are authors, such as Klems, Kaplan and Edwards, who consider that the cloud must be rapidly scalable and easily adaptable on-demand. Other authors, also consider that the cloud is in its essence a kind of resource virtualisation. Sultan even suggests that it should be automated. It makes perfect sense because the only way to get an on-demand allocation of resources and fast scalability is through the (usually automated) virtualisation of resources that can be dynamically allocated. According to (Hurwitz et al., 2009), the cloud is a business and economic model, and most definitions agree that providing a paid utility service is a cornerstone of cloud computing. The provided utility services can be of various nature, from a virtual infrastructure to a computing platform, or even end user software products. These services are usually delivered in an on-demand and pay-as-you-go basis, creating the concept of utility computing.

Finally, the NIST[4] definition for cloud computing tries to encompass most visions as follows (Mell and Grance, 2011). "Cloud computing is a business model in which service providers offer physical and logical resources, virtualized or not, with a high degree of scalability according to costumer's needs (on-demand) and billed according to the used resources (pay-as-you-go). All of this is done in a quick and possibly automatic way, providing computational resources as a service (utility computing)."

However, there are those authors who simply want to define the cloud based on the services it provides: IaaS (Infrastructure as a Service), PaaS (Platform as a Service) and SaaS (Software as a Service). The cloud architecture can be divided, at a high abstraction level, into four layers. On top of this overall architecture, cloud computing follows a business model in which services are offered according to customers' needs, i.e., the providers lease their hardware and other logical resources as a service that follows an on-demand policy. The hardware and infrastructure layers give origin to IaaS, the platform layer to PaaS and the application layer to SaaS. There are more specific cases that may cross boundaries between layers and these definitions.

The most relevant cloud service type for this thesis is PaaS. IaaS could be used to deploy virtual machines to run the developed software, however, managing those virtual machines instances would involve creating kind of a management layer that would deviate us from the main goal of this work making it a major development overhead. On the other hand, SaaS is at a very high level, since it offers specific services and software. Therefore, PaaS offers an environment that serves as a platform to build and deploy applications and services. It can be used to develop and run the intended platform that will maintain the model and framework for personalisation. Finally, cloud computing can be categorised into different types, depending on whether the access to the resources is open or restricted to a given organization: Public, Private, Hybrid or Intercloud.

2.4. Related Work

In order to provide answers to (ubiquitous) user modelling and personalisation, different solutions in the form of frameworks, languages, models and/or servers have been proposed by the research community. These different proposals illustrate how ubiquitous user modelling is being addressed along the years in order to support personalisation of applications in an increasingly Ubicomp world.

Ontologies constitute an interesting tool for the representation of context models and user models since they facilitate the sharing of information and reinforce interoperability. The General User Model

[4] NIST is the National Institute of Standards and Technology, which is a measurement standards laboratory, and a non-regulatory agency of the United States Department of Commerce.

Ontology (GUMO) (Heckmann et al., 2005), based on OWL, is a proposal used to manage the syntactic and semantic variations between user modelling systems, modelling user attributes and their interrelationships. Moreover, Jørstad et al. state personalisation shall be possible across a range of various devices, networks and services, which is why it is necessary to develop conceptual and abstract models that are independent of the underlying technologies (Jørstad et al., 2006). An ontology for personalisation of information elements was modelled based on this conceptualisation, using UML notation, but then transformed into an OWL specification.

Another good example is the Unified User Context Model (UUCM) (Niederée et al., 2004) that can be used for modelling attributes of the user and her environment, the user context. My Own Web (Lee and Yong, 2005) is an interesting approach directed to the Web personalisation focused on customising the content and the structure of a Web site towards its users. Authors proposed a Unified Personalization Platform that has similarities to what we intend to implement. However, it can be envisioned as a raw data collector of user profiles, weblogs and other sources from the Web or PC to feed the Profiling Engine.

Panayiotou et al. exploit the importance of time and experience in personalisation, by granting the moving user a system that anticipates and compensates the time-dependent shifting of user interests (Panayiotou et al., 2006; Panayiotou and Samaras, 2004). A prototype system was implemented and derived from a previous system, focusing on two context factors: time and experience/activity. They concluded that it provides a way to merge any different instances of the user's profile (e.g. vacation, work) into a dynamic profile (Panayiotou and Samaras, 2006). Another research, from (Weiß et al., 2008), presents a generic framework for application developers, which allows filtering on users' preferences and involved context. The work introduces an approach for context-aware personalisation of mobile multimedia services. Since it was built for application developers, the framework is generic, extensible and configurable concerning the application domain. The approach focuses on a user-based collaborative as well as a content-based filtering system that is supported by MPEG-7/21 metadata and profiles. They chose MPEG-7[5] for content metadata since it allows for generic descriptions of all kinds of multimedia content, being application-independent and easy to extend. The user model provided by MPEG-7 was integrated into the MPEG-21[6] framework that provides a standard for the transparent usage of multimedia content in heterogeneous environments.

Berkovsky et al. presented the user model mediation concept (Berkovsky et al., 2007), employing appropriate methods to transform the syntax and semantics of the user model used in one system into those of another system. The work introduced a generic framework and also focused on dealing with the heterogeneity of the available user model information, giving special attention to the resolution of inconsistencies and conflicts among data obtained from diverse systems. A mechanism for reusing generic user modelling data with different applications called UM Toolkit was already developed in 1995 (Kay, 1995). A repository of user data was the core of the solution and it was stored in the user's file space. This tool evolved into Personis (Kay et al., 2002), where the main difference is the appearance of a server to store a collection of user models for many users. More focused on Ubicomp was the development of PersonisAD (Assad et al., 2007), providing distribution of models, where authorised applications could request access to a model in order to use and reuse the model's allowed parts.

Moreover, (Kabassi et al., 2008) have presented a domain-independent method for user modelling sharing between applications. It is a user modelling server, called UM-Server, which embodies a multi-

[5] Multimedia Content Description Interface (ISO/IEC 15938): http://mpeg.chiariglione.org/standards/mpeg-7
[6] Multimedia Framework (ISO/IEC 21000): http://mpeg.chiariglione.org/standards/mpeg-21

criteria decision-making theory on the server side and provides services to any application that requests for it. In order to show the reusability of UM-Server, they have used a common user model in three different applications, an e-learning system for health issues and two intelligent user interfaces, one for file manipulation and one for e-mailing. They considered that the generalised method proposed can be successfully used by various applications, despite the differences that exist between their domains.

Yudelson et al. focused on a specific implementation-level aspect of user modelling servers – the choice of push or pull approaches to evidence propagation (Yudelson et al., 2007). They presented a new push-based implementation of their user modelling server CUMULATE and compared its performance with the performance of the original pull-based CUMULATE server (Brusilovsky, 2004). Another important reference is BGP-MS (Kobsa and Pohl, 1995), which allows assumptions about the user and stereotypical assumptions about user groups to be represented in a first-order predicate logic. A subset of these assumptions is stored in a terminological logic. Different assumption types, such as (nested) beliefs and goals as well as stereotypes, are represented in different partitions that can be hierarchically ordered to exploit inheritance of partition contents (a partition together with all its direct and indirect ancestor partitions thereby establishes a so-called view of the full user model). The BGP-MS system can be used as a network server with multi-user and multi-application capabilities (Kay et al., 2012).

Another seminal work is DOPPELGÄNGER (Orwant, 1995) which accepts information about the user from hardware and software sensors. Techniques for generalising and extrapolating data from the sensors (such as Markov models) are put at the disposal of developers. Unsupervised clustering is available for collecting individual user models into so-called 'communities' whose information serves the purpose of stereotypes. In contrast to all other user modelling shell systems, membership in a stereotype is probabilistic rather than definite. The different representations of DOPPELGÄNGER are quite heterogeneous and users can inspect and edit their user models.

Asangansi and Poslad presented a user modelling and personalisation framework for providing personalised services to users through their mobile devices during large sports events (Asangansi and Poslad, 2011). The user model combines the knowledge of sports events the user physically attends and the knowledge of the user's interaction behaviour when consuming multimedia content from his mobile while away from the sporting event's venue(s). The user model employs both explicit and implicit modelling techniques which are able to learn and represent shifts in the user's preferences. Ontologies are used to formalise the user model and domain knowledge thereby disabling ambiguities in preferences specification but introducing reasoning capabilities.

Bouzeghoub and Kostadinov proposed a slightly different approach based on a taxonomy of knowledge that constitutes a user profile (Bouzeghoub and Kostadinov, 2007). The taxonomy of profile dimensions and preferences works as the basis of a generic profile model (GPM) that can be used as a baseline for a large class of personalisation systems, accordingly to the authors. The term "generic" does not mean here a closed set of attributes and preferences, but rather a list of high-level concepts which can be instantiated, specialised, refined and augmented in each personalisation environment. Some of their requirements are: it should capture the main knowledge categories known in current personalisation systems; it should be independent of any DBMS or any IR system and independent of any specific application; and it should be easy to specialise, generalise and instantiate any entity type of any dimension of the profile. The authors wanted to provide design guidelines to help developers and users to create and manipulate profiles.

2.5. Summary

This chapter categorised and summarised the areas, topics and main projects related to our research, which appear and are referred throughout the document. Therefore, the research context was approached from three perspectives giving emphasis to the meaning of personalisation in the scope of this research (in section 2.1), to the major aspects to take into account in the personalisation modelling process, which are user modelling and context integration (in section 2.2) and, the fundamentals of web services and cloud computing (in section 2.3). Finally, related projects were presented.

Related work will be revisited whenever appropriate to present additional work contextually in each chapter.

3. A Model-driven Personalisation Solution

"Practice compassion, forgiveness and kindness."

(Yogi Teas, 2017)

The research described in this document aimed to propose a solution to support developers in the personalisation process of the human-computer interaction across Ubicomp applications. This chapter details the generic personalisation model in which the solution is based on (in section 3.2), giving a special focus to the presentation of the language defined to specify personalisations instantiations, then presents the user modelling framework that integrates the model (in section 3.3), and finally introduces the cloud-based platform responsible for making the whole solution available to all developers (in section 3.4).

3.1. Introduction

Our research work is strongly based on the proposal of a generic and interoperable model for personalisation of Ubicomp applications regardless of the intended scenario. The same personalisation model can be followed and instantiated by different applications, even if from different domains.

The model is the core of a cloud-based platform that, on the one hand, provides the tools, services, and partners to help developers in the implementation of personalised solutions. On the other hand, end users will be less bothered when starting to use a new system or application that needs "to know them." Based on information given by the pervasive generic model, the new application already might know something about the end users in order to automatically adapt itself to the end users' preferences and needs. Hence, an end user's interactions with a previous system or application might be useful to personalise the experience of the same end user when interacting with a new one. This scenario should be possible due to the use of that pervasive interoperable model, which guides and supports developers in the implementation of both personalised solutions. Figure 3.1 illustrates the concept behind this model-driven personalisation solution.

Therefore, we designed a solution comprised of the following main components:

- The **X-Users model**, which is the **pervasive generic personalisation model** that works as the basis of the solution. It has the purpose of guiding developers when applying personalisation across different applications in a standardised and consistent way.
- The **Context-Aware based Personalisation Environment (CAPE)** is the **modelling framework** in which X-Users is inserted alongside with the important module of user profiling.
- The **cloud-based Personalisation Platform for Multimodal Ubiquitous Computing Applications (P²MUCA)** intends to be simple to use while encapsulating the important mechanism of user modelling, allowing user interactions control and user preferences to be consistent and memorable across multiple Ubicomp applications.

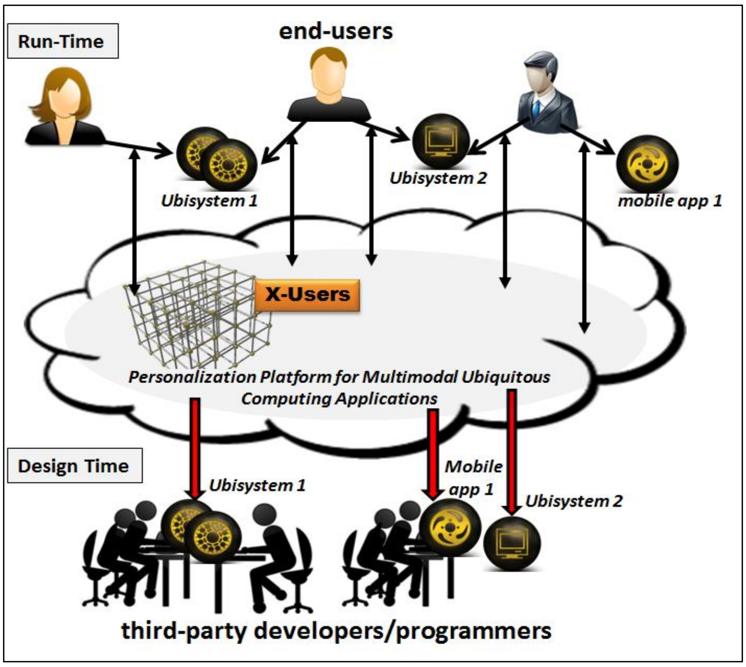


Figure 3.1 - A model-driven and cloud-based approach for personalisation.

3.2. The Generic Personalisation Model

As previously stated, we proposed to develop research needed to build a generic personalisation model. For a given application, personalisation can be represented by different elements such as personalised content, adapted interfaces or user interaction mode, and/or device's dynamic behaviour. The possibilities of personalisation can be wide and very dependent on the type and specificities of the application. Therefore, it is difficult to reach a generic solution for guiding a personalisation implementation and, especially, applying it in a completely automated way. A widespread and balanced solution to overcome the aforementioned limitations can be achieved by following the model presented in this research work.

A key task in systems that provide personalised services is the initialisation of a user model (Kay et al., 2012). Accurate initialisation of a user model is important and, in some cases, it is also important to reuse the user model. For instance, if one user only interacts with an application rarely, the latter needs a means of acquiring the user modelling information efficiently, without bothering the user (Kay et al., 2012). In such scenario, stored user modelling data from previous interactions is usually not available at the outset. Thus, the existence of a generic model based platform for personalisation, which provides interoperability, allows the extent to which systems and devices can exchange data and interpret that shared data. For two systems to be interoperable, they must be able to exchange data and subsequently present those data such that a user can understand it. Furthermore, other issues have been highlighted revealing how the identification and ownership of users' profiles may become susceptible to privacy and security concerns as they are left stored on diverse Ubicomp systems. A pervasive cloud-based platform would also address this issue partially, giving users more means to control their information since the focus is put essentially on one platform, which aggregates the data related to their profiles. We have defined a global architecture that embraces the previously presented features.

3.2.1. Model Overview

We can find the model divided into two parts:

1) a **personalisation sub-model**, where personalisation is really modelled;

2) the **user sub-model**, in which everything related to the user is represented, including context.

Developers should decide a priori, at design time, what to personalise in an application they are developing. It will be important that they have in mind who are the potential end users, mainly describing relevant attributes of some typical users (archetypes), which will drive the personalisations' requirements. Developers and researchers can use the personas design in order to find the needs of those potential end user. In recent years, interest in personas has been extended to the software engineering communities (Faily and Lyle, 2013). A persona is a "hypothetical archetype that represents a person who will interact with an information technology or system" (Cooper, 1999). Moreover, scenarios can be defined using the personas. Through scenario analysis, personas are used better to understand the user group requirements and needs. Since personas are behavioural specifications of archetypical users, they can justify design decisions at many levels of abstraction (Faily and Lyle, 2013). Persona and scenario analysis has some limitations as research methods, but it is useful to gather generalised information. Their specification can help determine what to personalise (instances of personalisation, e.g., a screen, an interaction mode) and the different options for each intended personalisation.

In our model, each personalisation instance (`Personalisation`) will have one or various `PersonalisationOption` according to the number of personas identified (Figure 3.2, see top entities). A `Persona` can be the driver to various Personalisations in the same application or across different applications.

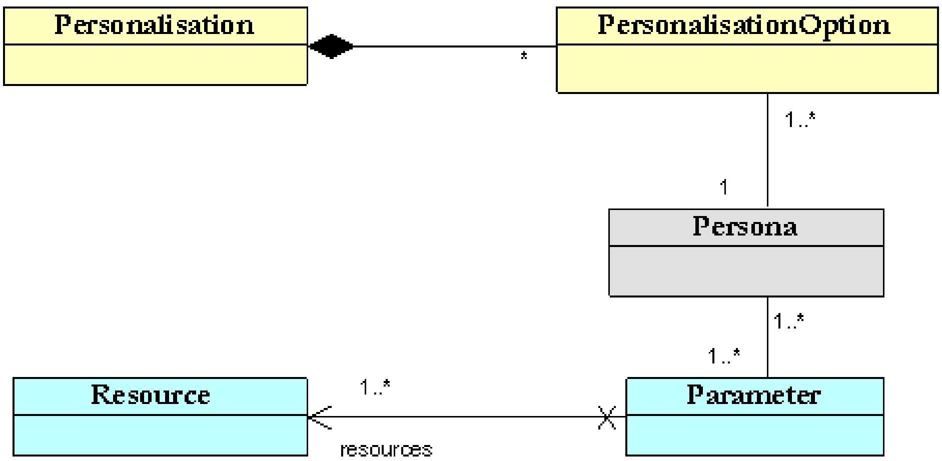


Figure 3.2 - Personalisation sub-model's core entities.

An application developer has to choose which combination of data must be taken into account in order to represent the personas. For instance, if for one personalisation of an application A we want to differentiate its users by level (e.g., basic, intermediate, advanced), it may be important to know how much time users spend using the application and the number of different functionalities executed by each of them. It means the developer of the application must study and select a priori the data resources, which are useful for personalisations, the application is capable of providing. In our perspective, it is desirable that the representation of a persona, in an application's context, should be made through detailed combinations of data resources that can be of different types. One or more arithmetic expressions can be defined by an application developer and referred as a parameter in order to represent personas, enabling the developer to use data more flexibly. A `Parameter` is expressed according to expression (1), which is described by the combination of N `Resource` through the summation of their weighted values.

$$\sum_{i=1}^{N} w_i * resource_i \tag{1}$$

For instance, a parameter *p1* could be obtained based on a value of N=2 (see expression (2)), in which a `resource`$_1$ could be the total time using the application by a user (`totalTime`), having a weight `w`$_1$ of 0.7, whilst `resource`$_2$ could be the number of different functionalities executed by the user (`countFuncs`), presenting a lower weight `w`$_2$ of 0.3 for calculating the final value of the level.

$$(0.7 * totalTime + 0.3 * countFuncs) \tag{2}$$

Therefore, after defining the desired personalisations and their options, developers will specify the set of resources, and how to combine and weigh them, resulting in parameters, in order to correspond to the intended personas (Figure 3.2, see bottom entities).

On the other hand, the user model is very important for supporting the appropriate implementation of personalisation. It is essential to have a consistent and generic user modelling process to gather the relevant information to support different users, which may evolve over time while using different applications. A significant change in the general profile of a user may be better followed inside an application for personalisation purposes if data obtained from the users' experience with other applications are also used to determine a profile for personalisation. The use of the cloud-based generic model, in which all registered applications rely on to implement their personalisations, provides the interoperability that enables a shared knowledge about the users that may be used to enhance their experience across the applications.

Having in mind the user model, what information about a user can and should be included and captured as a basis for the personalisation? To accomplish that capture, which are the potential sources for user modelling information? Data can be collected from, for instance, lifelogging sensors, personal machines and devices, and the cloud. With more sensors, such as GPS, camera, and gyroscope, embedded in mobile devices, it is possible to record the user's activities and behaviour. The usage of wearable sensors enables the capture of users' physical and physiological data. Even an emotional model (Gonzalez et al., 2007) can be predicted with the usage of the aforementioned sensors. The captured sensor data can be used to represent a situation event and extract patterns from the activities data logs. These patterns can be used to predict the user's situational interests, in order to create a user profile for the current domain, for better personalisation. Figure 3.3 presents a partial taxonomy of what can be used for user modelling.

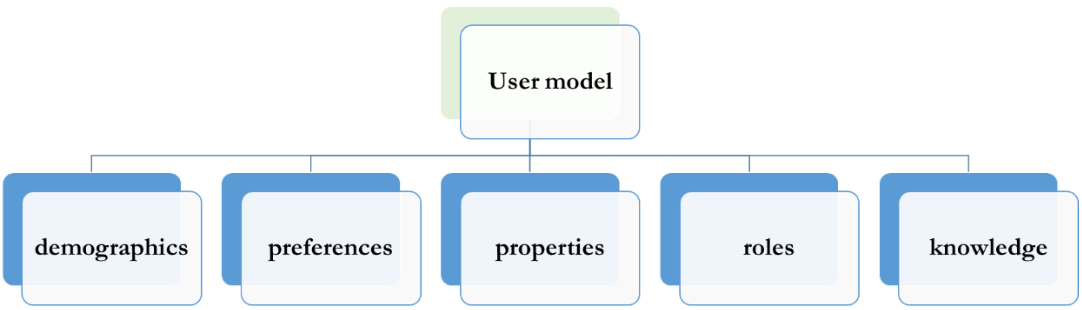


Figure 3.3 – The starting high-level taxonomy for the user model.

Currently, the X-Users' user model includes the essential user's data, such as demographics, and preferences or properties, as types of data resources, along with the (user's) interactions stream within an application. Moreover, it is important to add context to the process in Ubicomp scenarios, in which all information that usually has been provided by Web browsers should become accessible from just about anywhere, at any time, and be delivered in a manner appropriate to a specific location and context (Greenfield, 2006). Thus, regarding context, the environmental conditions can be inferred easily with the usage of sensors, while spatial information can be dependent on the use of smart objects included in the space, for instance. Figure 3.4 shows a high-level taxonomy that indicates what can be seen as context elements for inclusion in the model.

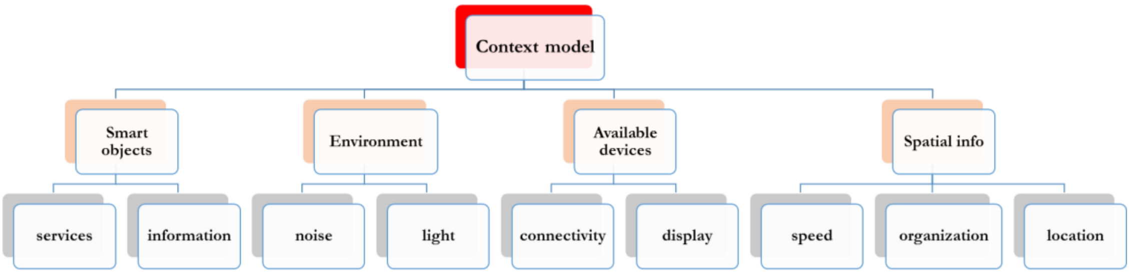


Figure 3.4 – A high-level taxonomy for the context model.

The user's interactions, combined with the possible context in which they happen, are very important data for personalisation. This type of context is referred throughout this work as "context segmentation" since it segments data according to the currently captured context. When an application updates data about a user (e.g., the number of clicks or logins), there may also be data on the current context of the user (e.g., weather conditions and location), depending on what was intended for the personalisation. In this case, the respective user's interaction data are stored, being associated with the specific context conditions. Context segmentation can refine personalisation results, since a user on some occasions might behave in a certain way, and in others s/he might have different interests or needs. That is why the incorporation of context may be relevant, having a significant effect on the prediction accuracy of user profiles. Therefore, user modelling and context modelling are strongly interrelated in our approach. However, context should be considered as a separate entity in the model and not as part of the user description because there are some substantial differences (Woerndl and Schlichter, 2007), as previously stated. The user description is rather static and somewhat longer lasting, implicitly observed or explicitly provided by the user, and presents a simpler or rawer nature. The context is highly dynamic and transient, not being permanently stored, and it is supposed to be observed or inferred and not inserted by the user (there can be exceptions).

Figure 3.5 depicts the main entities in the user part of the X-Users model, highlighting (with the darker background) the association to Persona. Summing up, when using an application, a user will have profiles that will match personas ("cluster profiles") according to each configured personalisation. Thus, the `UserProfile` can be seen as the output of the user sub-model that is used as an input for the personalisation sub-model, matching a cluster profile (Figure 3.6).

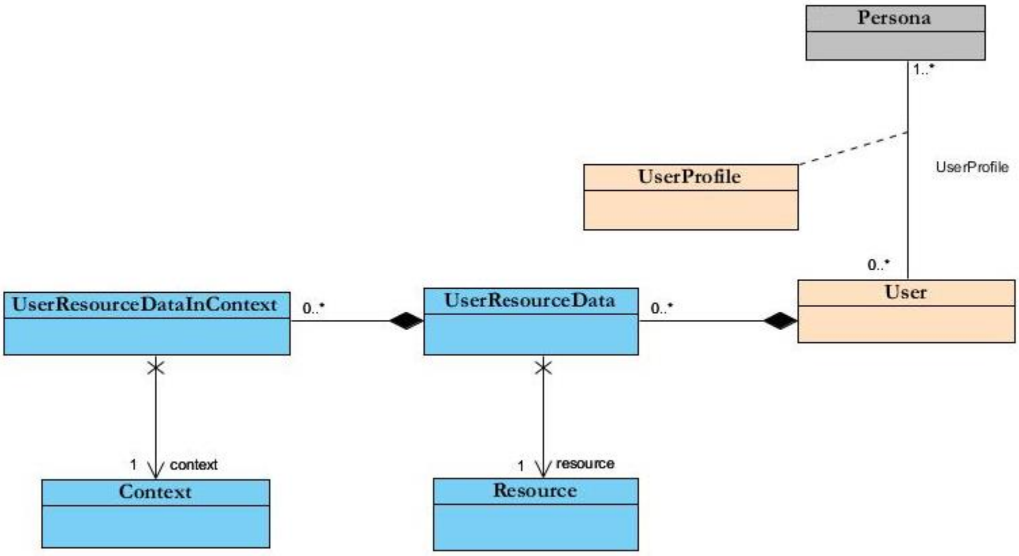


Figure 3.5 - User sub-model's core entities with the connection to the central entity of Persona.

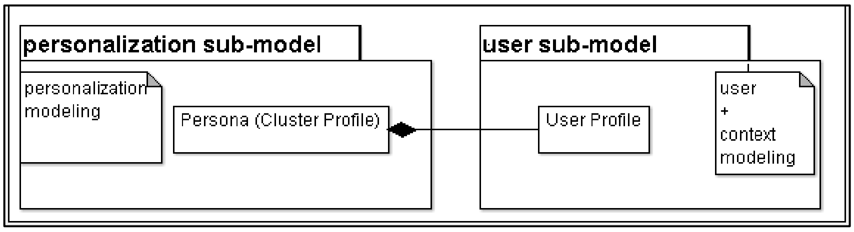


Figure 3.6 - X-Users overview "in a nutshell."

3.2.2. The XUPIL Language

The X-Users Personalisation Instantiation Language (XUPIL) is an XML-based description language, a Domain-Specific Language (Fowler, 2010), designed to be the format for the definition of the rules specifying the instantiation of a Ubicomp application's personalisations based on the X-Users model. This language allows interchanging personalisation instantiation definitions between different developers, applications, and modelling tools that may appear in the meanwhile. XUPIL defines an XML schema for specifying the declarative part of the personalisations instantiation according to the generic X-Users model. The language has been designed specifically to store all aspects for the definition of an application's personalisations. It contains elements to hold information about the core entities of the X-Users' personalisation sub-model, which are `personalizations`[1], `parameters` and `resources`.

Next, due to its expressiveness, we use the XML schema of XUPIL to detail the main aspects of the personalisation sub-model while presenting the language. The root element of the language is `xupil`, which is structured in four main elements in no particular order: personalisations, parameters, external services, and resources (see Figure 3.7). Moreover, it presents three important attributes that used in conjunction can provide the mechanism of reusability of parts of a personalisation, working on to the interoperability of the model and solution. The attributes are the following:

[1] This document uses British (UK) spelling, but XUPIL was defined using American (US) spelling.

- appname - refers to the identification name of the application which personalisation is being instantiated through X-Users.
- accessable – indicates if the entire personalisation can be used by other applications. The default value is private, which indicates the personalisation definition belongs only to the application in question. The value protected will indicate that the definition can be used by any other application from the same developer (team), and public indicates it is accessible to all applications registered with P²MUCA. Appendix A.3 shows how it was used in the personalisation of one of the prototypes used in the studies.
- refapp – indicates the application whose personalisation definition is being used to define the new application's personalisation. Appendix A.4 shows how it was used in the personalisation of one of the prototypes used in the studies.

Moreover, both attributes accessable and refapp can be used in all the elements that define a personalisation. For instance, if accessable is used with only one parameter then this might be used by another application that references it. The structure of xupil is detailed as follows.

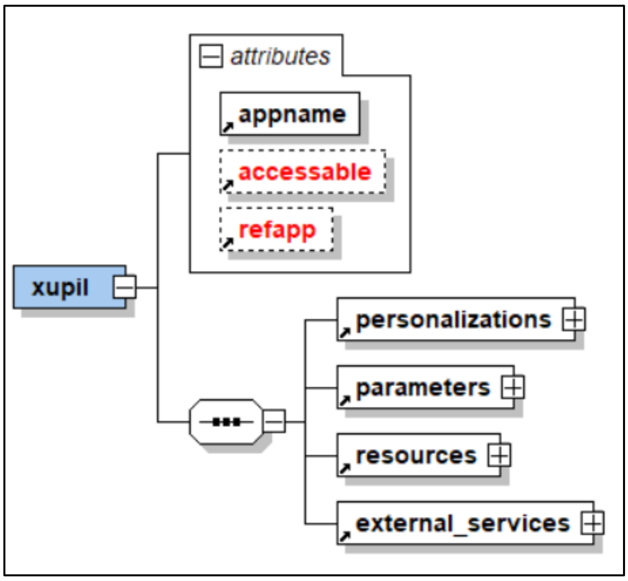


Figure 3.7 – The root node of XUPIL language.

Personalisations

The element personalisations specifies the set of personalisation instances that will be used in the personalisation of an application. Each instance of personalisation is defined by (see Figure 3.8): 1) a name, 2) a type, 3) used context (optional) and 4) the listing of all the personalisation options. The element name identifies the personalisation and it should be unique among all the personalisations instances of an application.

The element type is used for situations where different instances of personalisation require different user profiling techniques to get better results. At the moment, CAPE only implements one technique, but this element might be more relevant as more techniques are added to the framework.

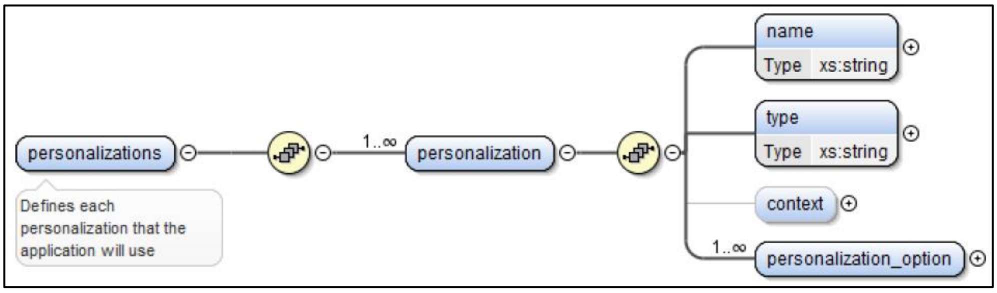


Figure 3.8 – Specification of the personalisation instances to be used in the personalisation.

The element `context` is optional and, as aforementioned, it works as "context segmentation," because it segments users' data according to the current context. The element `context` is composed of an element named `total_context_weight` and a set of context resources that have a name and weight (Figure 3.9). The name specifies what kind of context will be used and the weight identifies how much does a context element weight in comparison to other context elements in the personalisation. The sum of the weights of all context resources must always be equal to 1, i.e., each weight is given by a percentage.

After configured, the use of context works in the following way. The operation starts with an application personalisation request that sends the information concerning the user's context, according to the context resources included in the personalisation configuration. CAPE's repository stores the resources data segmented by context. Those values are retrieved in order to be weighted with the assigned weight for each context. The result of the weighting operation is multiplied by the total context weight value. User's resources data that are not segmented by context are retrieved from the database and multiplied by `(1-totalContextWeight)`. The non-contextual result is arithmetically added to the contextual result originating the final value, as defined in expression (3), which will be used for clustering purposes.

$$[(1 - totalContextWeight) * (nonContextData)] + [(totalContextWeight) * \sum_{i=1}^{n} (W_{C_i} * C_i Data)] \quad (3)$$

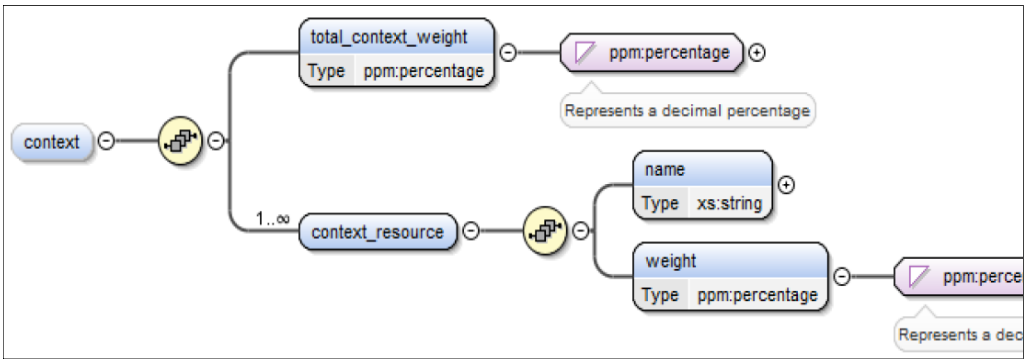


Figure 3.9 - Defining context in a personalisation instantiation.

The personalisation options are the information pool from which one will be selected and sent to the application client. The personalisation option is the answer to a personalisation request regarding an end user of the application. Figure 3.10 shows that each option is defined by a list of parameter options and possible external service selections. For example, a developer may want to define a personalisation instance in which the personalisation options depend on a parameter called userLevel (basic, intermediate or advanced) and on an external contextual service for weather. P²MUCA can integrate multiple external services (e.g., OpenWeatherMap[2]), providing an abstraction to applications' developers that can use them for personalisation. Thus, they are used for contextual post-filtering, which means it works as context used after the clustering process execution.

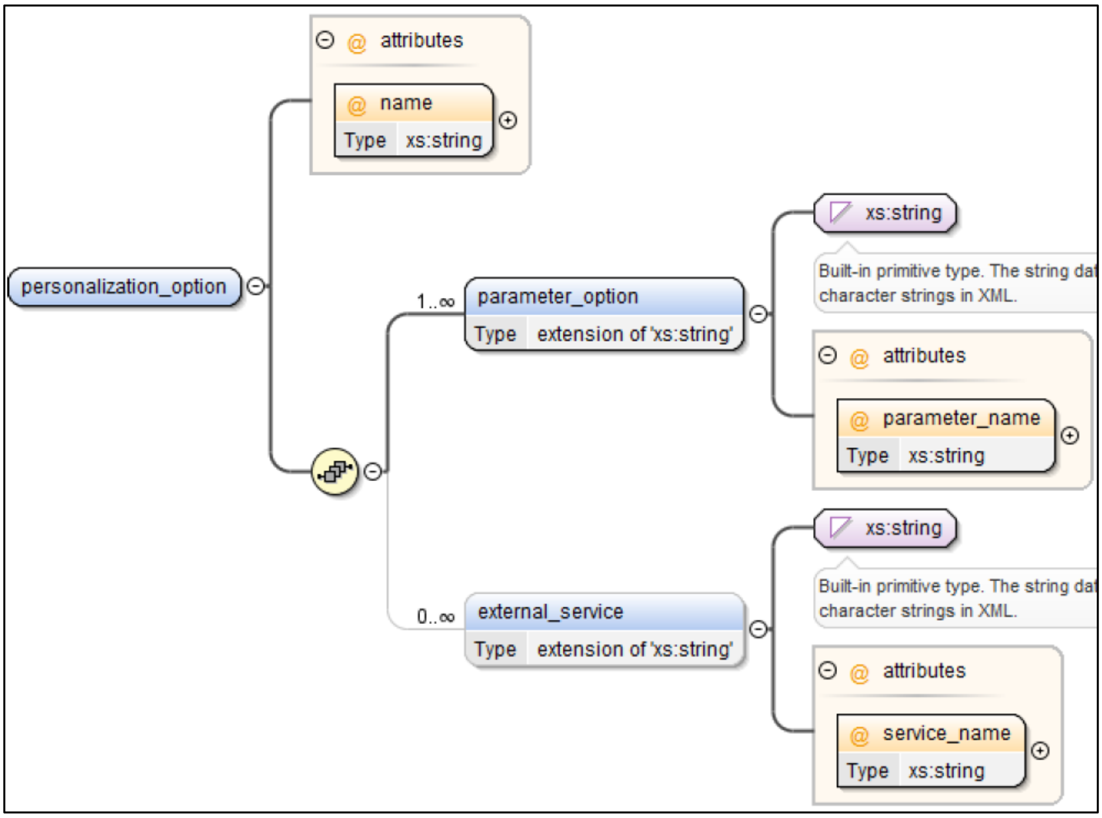


Figure 3.10 - Definition of a personalisation option.

Parameters

This element contains a set of ordered options that identify the clusters that will be used in the clustering process (see Figure 3.11). It is important to point out that the order in which the options are defined is crucial to get the correct results. When defining a parameter, the developer must be aware that the first parameter option should be semantically equivalent to the lower value, while the last parameter option should be equivalent to higher values. For instance, if a parameter is used to distinguish users by profile level (basic, intermediate and advanced) and used data consists of the sum of time each user spent logged-in, then a user with a small value would be considered basic, and another user with a higher value would be considered advanced (considering they have significantly different values). On the other hand, if the advanced option were defined before the basic one, users with a small value would be considered advanced and vice versa.

The developer may create one or more different arithmetic expressions for the parameter definition. Those expressions currently support integer and double type values, basic arithmetic operations (sum, subtraction, multiplication, and division) and the usage of parenthesis to express arithmetic priorities between different expressions. The algorithm needs an input in the form of a vector to perform the clustering operation. Each index of the vector represents a user of the application. Each parameter's expression represents a different dimension of the data vector that will be used in the clustering process. Instead of having a vector of users with only one value per user, a matrix of users is used,

[2] openweathermap - https://openweathermap.org/

where each user may have more than one value that corresponds to the number of expressions defined.

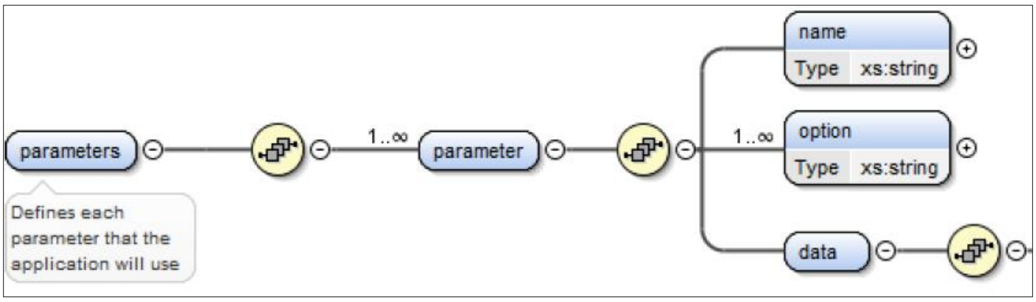


Figure 3.11 - The parameters modelling.

Resources

We considered three types of resources (Figure 3.12): interaction stream, context, and preference. The first element, referred as `resource_appdata`, addresses interaction stream data, i.e., data obtained from regular interaction of the user with the application (e.g., the number of logins or time logged-in). We have considered different types of data updates. For instance, the number of logins can be incrementally updated, and the time logged-in can be a sum of time values spent by the user every time s/he was logged-in. Therefore, the developer must specify what kind of update type will be used for a given resource.

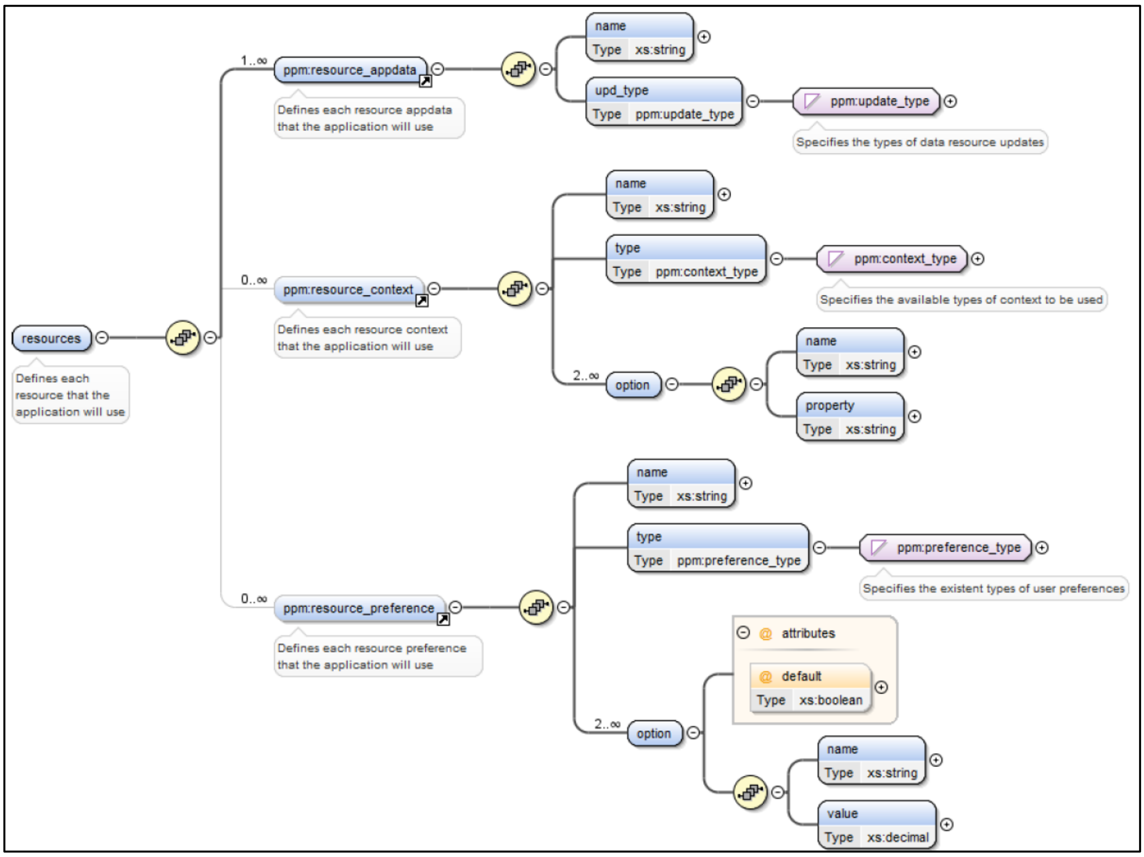


Figure 3.12 – Resources definition.

The second element - `resource_context` - addresses data concerning the context of use of the application. Each context resource has a name that identifies it and also a type to describe it. Following that, there is a set of options that depend on the chosen resource context type. Those options are used to describe how many context options will be used in any personalisation process that uses context segmentation. The number of options depends entirely on the level of detail the developer wants to use. Each context option presents a name and a property, which is used to define the circumstances under which it should be applied.

Finally, the third resource element – `resource_preference` - has a name to identify the preference, a type of preference (e.g., open-text or multiple-choice), and a set of options that may be used to describe multiple choice preferences. Each option also has a name, a numeric value and a Boolean attribute called default, which indicates if the option is the default (in case the user does not specify a preference). The value element can be used arithmetically in parameter expressions just like any other interaction stream resource, being important to refine the final values of personalisation towards users' preferences. Expression (4) defines a simple example of a parameter, which can represent the level of a user between basic and advanced, for instance.

$$timeLoggedIn * preference_1 \quad (4)$$

The first variable in the expression is the most important one and corresponds to an interaction stream value related to the total login time of the user when using the application. The second variable corresponds to a specific user preference value that can influence the user level. For instance, it can be obtained from a multiple-choice question in the application from which the user chooses the option with which s/he identifies her/himself. For example, if the user chooses an option (choice) with a corresponding value of 1.1, her/his final value for that parameter expression would suffer a numeric increase of 10%, which can be seen as a value refinement.

3.3. The Modelling Framework

This modelling framework, named **Context-Aware based Personalisation Environment (CAPE)**, encapsulates the X-Users model, which is its core, and consists of three main functional components:

1. A PHP-based **Web services interface** providing the XML-based configuration modules to register applications and specify their personalisations;
2. A Java-based **personalisation API** in order to present a high-level interface to developers;
3. The context-aware clustering solution to **user profiling**, which is an unsupervised learning approach.

3.3.1. CAPE's Architecture

CAPE is implemented as a client-server model architecture and was originally designed as a web-service to work upon HTTP requests. The adoption of a client-server architecture makes it possible to support heterogeneous clients. Figure 3.13 shows the diagram of the former architecture, including the main methods provided by the Web services and the Personalisation API components. It can be seen as a 3-tier client-server architecture, although the two upper tiers can be considered as being closely interconnected between each other due to sharing needs of some logic.

The upper tier presents the Web services component, which is the CAPE's interface to client applications. It receives the HTTP requests from the clients and forwards them to the corresponding Java tools of the Personalisation API that is in the middle tier. This forwarding procedure is shown in the diagram through the connection with the subtitle "Java/PHP Bridge." PHP was the scripting language used to implement the inner version of CAPE's web services, while Java is the programming technology used for the development of CAPE's core. CAPE is using the PHP/Java Bridge (PJB)[3]

[3] PHP/Java Bridge - http://php-java-bridge.sourceforge.net/pjb

in order to establish the communication between PHP and Java. PJB is an implementation of a streaming, XML-based network protocol that can be used to connect a native script engine (e.g., PHP, Scheme or Python) with a Java virtual machine (Bökemeier and Koerber, 2013). So, in CAPE, there is a PHP front end that is contained in, or associated with, a Java backend, which implements the Personalisation API. Therefore, the invoked front end uses HTTP/XML to invoke Java procedures of the running back end. The first tier is used to redirect personalisation requests and other operations concerning personalisation to the second tier of the architecture, which implements the logic part.

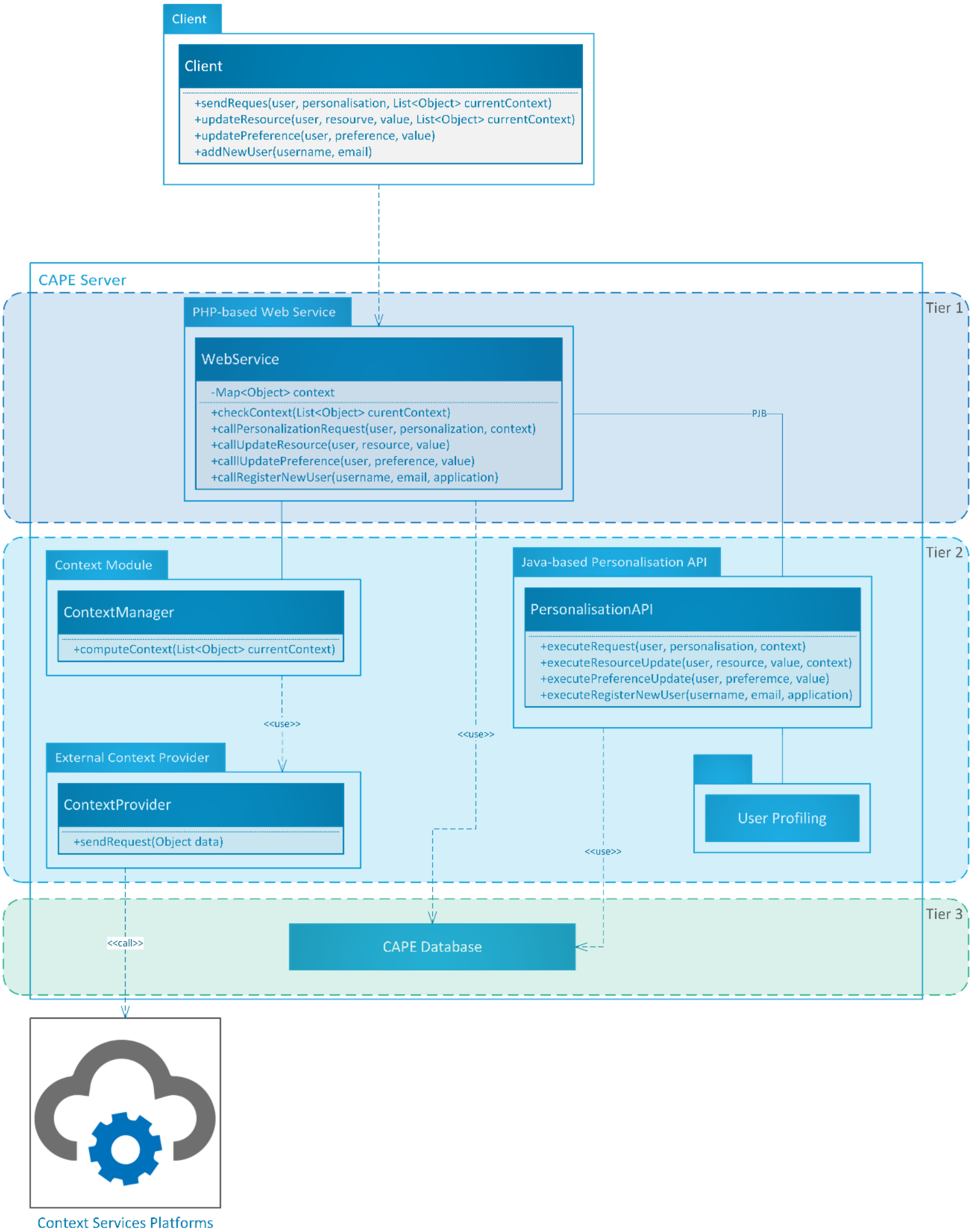


Figure 3.13 – CAPE's architecture.

The upper component also has other important functions, such as obtaining context information. As previously stated in Section 3.2, a personalisation instantiation may include a context-aware approach (Context Segmentation or External Context Services) to refine the user profiling results. Those context-aware approaches may need information that can be obtained from outside the CAPE system. For instance, an application may need to access a weather forecast service in order to improve personalisation. Given GPS coordinates as input, the Context Module will use them to handle the weather forecast procedure and provide the right contextual information to feed the personalisation component and the user profiling algorithm, more specifically.

The integration of context services providers (usually, Internet of Things (IoT) platforms) can prove to be helpful in order to facilitate the incorporation of context by developers into the personalisation of their applications. Besides the IoT platforms, others examples are weather stations and geographical coding (the act of converting addresses to latitude-longitude coordinates, and vice-versa). Therefore, the architecture shows the connection to those external services for context data gathering. The external context providers that are currently integrated and handled by CAPE are the following: World Weather Online[4], Weather Underground[5], and OpenWeatherMap[6]. The first version of Xively[7], which was known by Pachube, was also included. Sen.se[8], ThingSpeak[9], and GroveStreams[10] are other IoT platforms that should be taken into account in the near future.

As observed, the context module may use an external service to obtain more information. However, there are types of context that do not need that procedure. For instance, if hour intervals are used as a context, then the context module only needs to use the hour of the personalisation request in order to make a match between that timestamp and the previously defined hour intervals. This example is one of the simplest cases of context integration.

A usual interaction with CAPE consists of a client application making a request using the web services interface. This component will do the necessary pre-processing of data and redirect the request to the Personalisation component in the logic middle tier. Here, the desired outcome (e.g., the personalisation option or something like a simple update of resources) will be determined using the appropriated method and the database to obtain the necessary data and, finally, the result will be sent backwards to reach the client application. The Personalisation API was deployed as an external Java library to work as a proof of concept to validate the personalisation model during this research work. Therefore, it is also made available as an external library that can be imported to Java projects, mainly for testing purposes and additional implementations by developers.

If the request is for determination of the personalisation option, then it will execute the user profiling module, which is a context-aware and unsupervised learning approach. At this point, it is based on a clustering method since we found out that, in most cases, it is intended to provide personalisation with a minimal effort for the third-party developer. However, new machine learning methods can be easily added to this component of the framework so that they can be selected and used according to the specificities of each application. The user profiling module uses the Weka[11] framework, which is a collection of state-of-the-art machine learning algorithms and pre-processing data tools (Witten et al., 2011). The K-means algorithm is used due to its simplicity (easy to adapt), effectiveness, and

---

[4] worldweatheronline - http://www.worldweatheronline.com
[5] wunderground - https://www.wunderground.com/
[6] openweathermap - https://openweathermap.org/
[7] xively - https://www.xively.com/
[8] sen.se - http://open.sen.se/
[9] thingspeak - https://thingspeak.com/
[10] grovestreams - https://www.grovestreams.com/
[11] Weka framework - http://www.cs.waikato.ac.nz/ml/weka/

speed (especially for a small number of clusters), producing tighter and more accurate clusters in comparison to hierarchical approaches. The chosen metric distance is the classic Euclidean distance.

Lastly, the database is placed in the lower tier and is mainly the implementation of the X-Users model. It stores data concerning the definition of the applications' personalisations, the users, and their interactions with the applications. This database is filled with data for each client application's personalisation through the loading of the XUPIL file that declares them according to the X-Users modelling. The submission of the XUPIL file to CAPE by a developer marks the beginning of the personalisation process of an application that is being designed by that developer's team.

3.3.2. The User Profiling Algorithm for Personalisation Option Requests

The most important method of the CAPE's Personalisation API is the `executeRequest` (`sendRequest` in the Web Service interface) for a given personalisation of a given application. Every time a personalisation request is made, the user profiling algorithm will compute the user profile in order to provide the corresponding personalisation option (see Figure 3.14).

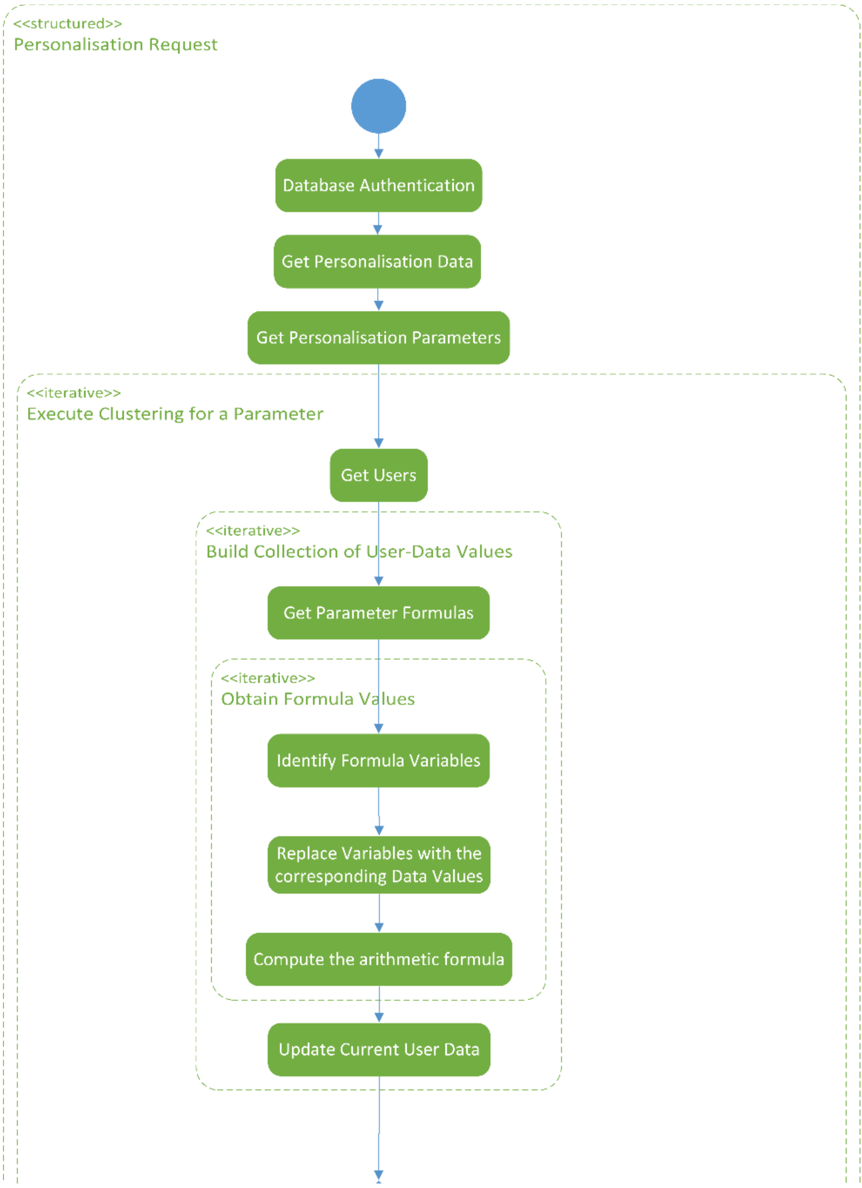


Figure 3.14 – Part one of the personalisation request algorithm flow.

Figure 3.14 illustrates the first part of the algorithm, in which its actions are exactly the same regardless of the use, or not, of contextual segmentation. Therefore, upon receiving the request from an application on behalf of a user, having her/his identification, the personalisation identification and context data attached to the user's request, the algorithm simply consults the parameters that are being used for the requested personalisation. Then, it parses the expressions in each parameter for every user that uses the application. It results in a matrix of users-values where each line represents a user and each column represents a data dimension. For instance, the clustering process will have three dimensions in case there are three expressions for a parameter. The matrix is used as input for the clustering process, which outputs the respective clusters for each user. The clusters have a direct correlation with the parameter options, meaning that the combination of clusters resulting from clustering every parameter used in the application will result in a single personalisation option. User profiles are generated by a machine learning module using the k-means clustering algorithm.

For a personalisation including context segmentation, the data structure of users with the corresponding parameter expression's values is populated in the same way as it was previously, but instead of sending it immediately to the clustering operation, the algorithm makes some modifications to the user data. For each context option to which the requesting application was assigned to (being alone or at work are examples of context options), retrieved data are conditioned by the user's current context instead of getting the normal user's interaction stream from the database (see right flow in Figure 3.15).

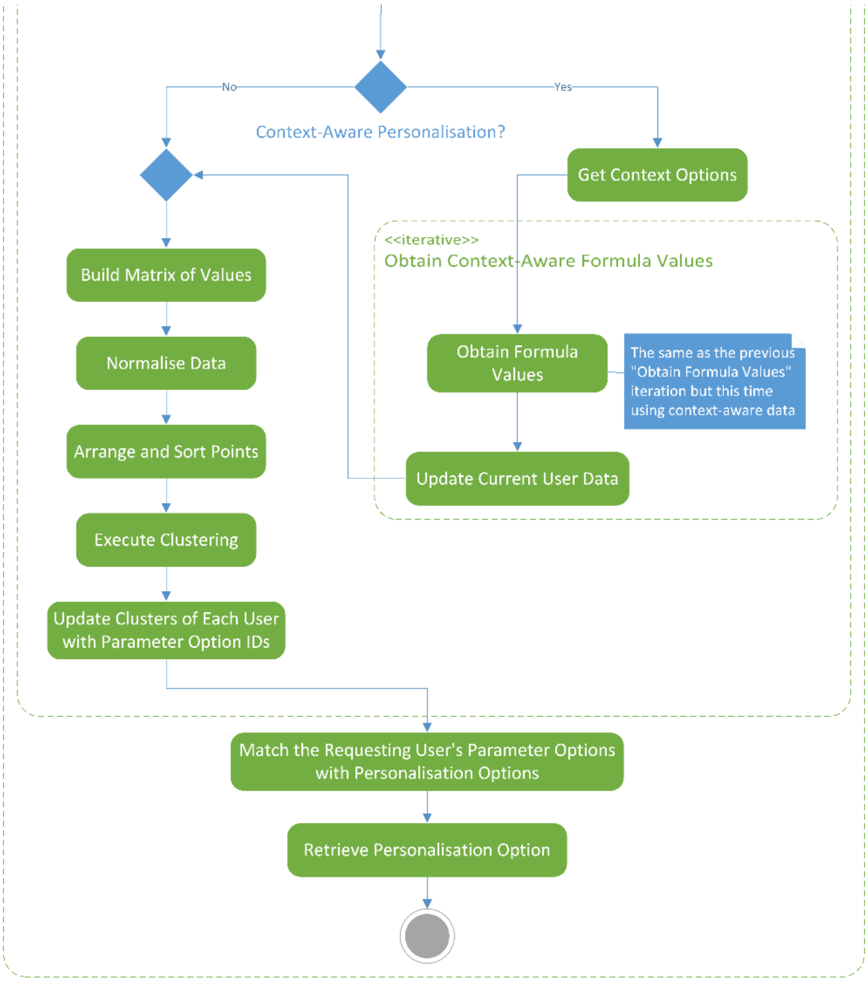


Figure 3.15 - Part two of the personalisation request algorithm flow.

In case the personalisation addresses any external service (it will be done as a post-algorithm filtering), the external service will be consulted and the personalisation option along with the external service's response will be retrieved to the user as a response to the personalisation request.

3.3.3. Personalisation API's Methods

The personalisation option request needs a framework that offers developers a way of managing effectively users, and their respective data, in order to be used with success in the personalisation of the applications. Therefore, the personalisation API provides other important methods, besides the one for personalisation requests, which are the following: `registerNewUser`, `resourceUpdate`, and `preferenceUpdate`.

The first one is responsible for associating a new user to an application. The program verifies if the user already exists and, if that is not the case, the user data will be registered in the CAPE's database, along with her/his association with the application. Thus, if the user is already registered with CAPE, only the second step is executed. It is important to state that a user may use more than one application. Right after the registration steps, the user's resource values are initiated with zero, and her/his preferences need to be initiated to their default values.

The method to update the user's resources values regarding interactions data - `resourceUpdate` - starts with the validation of the update type of the resource (e.g., incremental, addition). After that procedure, the current resource value is retrieved from the database and used to recalculate itself, if the type so requires, and finally updated in the database. Since the application may use context segmentation in one or more of its personalisation instances, it is also required to do the update of the resource values under the context in which the user interactions occurred.

A user's resources of the type preference are much more static and longer lasting than the interaction resources, which are highly dynamic. Nonetheless, those user's preferences will probably change with time so the framework needs to be able to handle their updates. Therefore, there is a method - `preferenceUpdate` - specifically designed for updating this type of resource. Its algorithm is simpler than the previous one since it consists in checking the preference type (e.g., multiple choice, plain text), updating the old value with the newer following the preference type directive.

In section 3.4.3, the set of methods provided by the personalisation API is addressed in more detail since it is extended by the platform.

3.4. The Cloud-based Platform

The Personalisation Platform for Multimodal Ubicomp Applications (P^2MUCA) was designed to facilitate even more developers in the process of applying personalisation to their applications.

3.4.1. Principles

This research work has a clear focus on people's daily lives, increasingly underpinned by a Ubicomp based world. Indeed, for what concerns personalisation, awareness of human factors is very important and it is the ultimate goal. Since the motivating challenge of this research is related to human factors, our model's usage strives for application's technology and device's platform independence.

It should be noted that users may interact with Ubicomp systems and applications through the use of devices powered by Android, iOS, Windows or other mobile operating systems. The modelling framework, CAPE, already addresses the requirement of the operating system and programming technology independence, providing a collection of web services based methods on top of the Java-based personalisation API. The latter is the only technology-dependent component, which is a programming technology, in this case. However, it is available as a Java library mainly for testing purposes by developers of Java-based applications, since our solution provides the platform that should be the pervasive widespread tool to be used by all developers, independently from the programming technologies.

The platform, which is based on the cloud, gives them a higher layer of abstraction than CAPE does, providing greater independence from the applications deployment technologies and devices operating platforms. P^2MUCA allows developers to give their end users a better experience through the use of a collection of personalisation tools that developers can rely on and to which end users can trust their data. To reach such goals, it was required to implement a robust and highly available solution that also presents mechanisms to put end users in control of their personal data. Therefore, P^2MUCA was designed to support two kinds of users:

1. Developers, which will have to register their client applications to use the service;
2. Applications' end users, which should allow applications to have access to their data, being able to manage their data used for personalisation purposes.

This dichotomy between developers and end-users is similar to what we have on platforms such as Facebook and Twitter. In platforms such as P^2MUCA, end-users must be able to authorise client applications to get access to their user profile data. P^2MUCA was then implemented as an HTTP-based service. It is justified by being a simpler service model that can be easier to be used by third-party applications that, this way, will only need to worry about sending HTTP requests. Additionally, it is possible to extend the current design to support WS-* services if that option proves to be advantageous to third-party developers that use the platform. Other important aspects taken into account in the service implementation are the mechanisms of authentication and authorisation. While developers should be able to get access to user data and the tools/services offered by the platform, users should be able to have control over which applications have access to their data. This aspect is very important since the personalisation framework may hold confidential data that must be protected from unauthorised access.

Several standards could be implemented in P^2MUCA to deal with authentication and authorisation. The latter is done with a simple yet effective login form on the P^2MUCA's interface with the user. On the other hand, given that the major problem here is authorisation, a robust mechanism such as OAuth 2.0[12] was selected for the authorisation implementation (IETF, 2012). Standards that are more complex could have been used, but it made sense to choose OAuth, considering the ease of use aspect and the goal of using the P^2MUCA's API in the open web and not in an enterprise environment. OAuth is also used by major Internet websites and applications.

An extension to Spring Security called Spring Security OAuth[13] was used to implement the server-side of the OAuth protocol. This library supports both OAuth 1.0 and 2.0, but only version 2.0 was used because it is simpler to work with and it provides a high level of security that is enough when used over HTTPS.

Regarding programming technology, the natural decision was to continue using Java-based tools to implement the platform and to ease the integration process of CAPE, since the latter is implemented using Java technology. Java EE (Enterprise Edition) presents the standard tools to implement web applications and services in Java. They are just a set of standards for multiple purposes, not being

[12] OAuth 2.0 - https://oauth.net/2/
[13] Spring Security OAuth - http://www.springsource.org/spring-security-oauth

tied to any particular implementation. However, GlassFish[14] is considered the reference implementation, so it was used to reach the requirements of P^2MUCA.

3.4.2. The Architecture of P^2MUCA

The platform's architecture is intended to be comprehensive to potentially cover various settings of Ubicomp scenarios and allow the time decrease in different implementations of personalised applications by third-party developers.

The platform has been designed to integrate two major components:

1. P^2MUCA website - a front-end for registering new applications and instantiating the underlying personalisation model for an application. The latter can be executed either by submitting the XUPIL file or by using the provided configuration user interface that makes the process easier. The website is also used as a user sign-in and authorisation point, allowing users to grant and revoke an application's access to their data;
2. P^2MUCA service - implements the HTTP service API protected with OAuth 2.0. It uses CAPE to support all of the provided API calls.

Figure 3.16 illustrates how both P^2MUCA's components depend on each other. For instance, the P^2MUCA website allows defining the instantiation of an application's personalisation using CAPE, but that same definition must be shared with the P^2MUCA service. Moreover, OAuth data must be shared between both, because the P^2MUCA website is responsible for creating applications' client credentials and access tokens, but the P^2MUCA service endpoint must use that same data to authorise each request made by the client applications. Therefore, two databases are considered, which are the following:

1. A database that stores the data about P2MUCA's users, applications, and OAuth credentials;
2. The CAPE database that is manipulated by the two components.

The service component offers an HTTP based API that is secured behind an OAuth 2.0 Resource Service. This Resource Service must have access to the P^2MUCA database since there are data that must be shared with the OAuth 2.0 Authorization Service running on the P^2MUCA website. The service is deployed as a Spring Security filter with the aid of the Spring Security OAuth library. Since the OAuth protocol protects data from unauthorised access, first, an application must get both an access and a refresh token in order to be able to use the personalisation API on behalf of a user. Therefore, these tokens can only be obtained by the application if the user authorises it. Afterwards, the access token can be used by a pre-determined amount of time to make API requests on behalf of the user, while the refresh token can be used to get a new access token if the original one expires. A user can always use the platform to revoke application access to his/her data. In this case, both the access and the refresh token become useless. This mechanism gives a user the express control over who can access his/her data.

There are three scenarios of use to take into account regarding the integration of OAuth 2.0 modules in the architecture. The scenarios are the following:

- Authorisation code grant flow. An application asks for permissions from the user using its client id, getting an authorisation code that can be exchanged for the access and refresh tokens by providing its client id and client secret. The refresh token can be used to get a new

[14] GlassFish - http://glassfish.java.net/

authorisation token if the access token expires after a given time. With this procedure, the user does not have to go through the authorisation page once again.

- Implicit grant flow. This procedure should be followed in cases where the client secret should not be used because its privacy is compromised (e.g., a browser-based application), providing an alternative simpler flow. This way, a short-lived access token is issued as part of the redirect URI while a companion refresh token is not provided. Once the access token expires, a new one must be requested which means that the user must, once again, be redirected to the authorisation URI. However, if the client application is already authorised and the user is still logged in, the user should be automatically redirected back to the application and the new access token will be delivered to it through the redirect URI.

- Client credentials grant flow. In this case, the client credentials (client id and client secret) can be exchanged for an access token that can be used to make requests on behalf of the client itself.

The goal of this cloud-based personalisation platform development is to facilitate the implementation of a personalised interaction across different Ubicomp applications. P²MUCA should work as our personalisation solution's ultimate interface with the developers, encapsulating the modelling framework and its underlying model. Therefore, the HTTP API module in the P²MUCA Service component (see Figure 3.16 - Architecture overview of P2MUCA.) is of high importance by abstracting the underlying implementation and only exposing operations the developers need, reducing their cognitive load. A developer must be registered with P²MUCA to get access to the services, being then able to register applications in development to start making personalisation requests using the HTTP-based API.

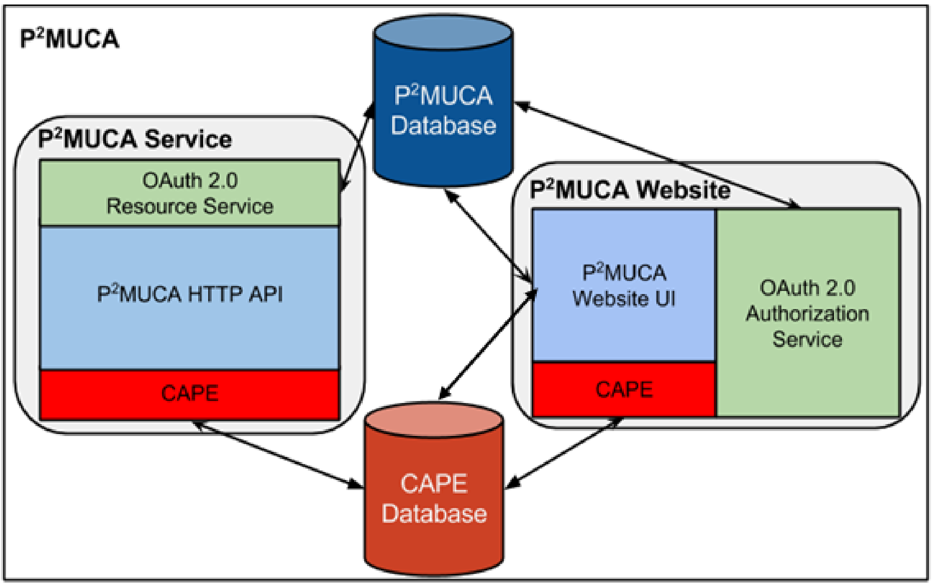


Figure 3.16 - Architecture overview of P²MUCA.

This way, the exposed Web API is a direct mapping of all the operations provided by CAPE, so, it appears on top of the CAPE module in the architecture schema. The HTTP API wraps CAPE's operations and exposes them through a simple HTTP endpoint secured with OAuth, which can be used by any application developer by issuing ordinary HTTP requests. As mentioned in section 3.3.1 - CAPE's Architecture, CAPE communicates with its inner database, which is highlighted in P²MUCA's architecture to be noted that it is again deployed as a platform component, although belonging to CAPE's architecture, conceptually. P²MUCA website also needs to access the CAPE database, besides accessing the P²MUCA database. When a developer is creating or editing instantiation settings for the personalisation of an application, then the Website module must use the CAPE configuration module to apply changes, which will be recorded in the CAPE database. This is done through the P²MUCA Website UI either by submitting the XUPIL file or by using a newly developed P²MUCA configurator (presented later in section 3.4.4). P²MUCA Website UI consists of a website for users and developers. Essentially, users can register, create applications and manage permissions given to third-party applications. The Website UI also needs to access the P²MUCA database to manipulate the OAuth data while creating or editing applications and every time users give or revoke application permissions.

Moreover, the P²MUCA database also stores user authentication information, such as usernames and passwords of the end users and third-party developers, so that they can authenticate with the website and manage their applications and permissions, respectively. The OAuth 2.0 Authentication Service is part of the P²MUCA website and it was implemented thanks to the Spring Security OAuth library, just like the Resource Service module. It is responsible for the authorisation endpoint and the token grant endpoint. The first is used when users are asked whether they authorise a specific application, and the second returns access and refresh tokens when one of the multiple grant types defined is used (IETF, 2012).

3.4.3. The HTTP-based Personalisation API

As previously presented in section 3.3.1, CAPE already included a first version of the personalisation API with a PHP-based frontend. However, it was mostly developed to work as a proof of concept to validate the personalisation model throughout the research work before implementing the personalisation platform. Therefore, P²MUCA wraps and extends the original CAPE's API in order to provide a more feature-rich service that can be publicly available, thus presenting a stronger focus on availability, ease of use and security. All operations/calls are protected using OAuth 2.0. As already stated, the P²MUCA service was developed as an HTTP-based web API. It mainly follows a remote procedure call pattern and not the standard RESTful service since the provided operations are not closely mappable by the latter approach. The complete set of supported API calls (operations) is shown in Table 3.1 and Table 3.3. The first Table presents the operations made available to developers for management purposes. On the one hand, the `validate` call will output both the application and the end user to which the access token is associated. On the other hand, the `addUser` call can be used by a developer to add end users (through their usernames and emails), associating them with the application that s/he is developing/managing.

Operation	Method	Input Params	Result (as a JSON object)
/validate	GET	access token	Application's *clientId* and end user's *username* for the given access token.
/addUser	POST	username	*status* property equal to success or error and a *message* property with the status message.
		email	

Table 3.1 – HTTP Personalisation API calls: management operations.

All calls results are outputted as JSON (JavaScript Object Notation) objects. For instance, in the case of a `validate` operation, the result will be the one presented in Snippet 3.2, which was produced by making the following call: "`https://service-p2muca.rhcloud.com/validate?access_token=c0c1fc49-87c3-4a95-b6a6-3e079e4701fe`".

```
{
        "clientId":" WeSync",
        "username":" user0"
}
```

Snippet 3.2 – JSON-encoded result for a validate call with a specific access_token.

An access token must be passed in all cases as a URL parameter with the key `access_token` or as an HTTP authorisation header with the keyword `Bearer` followed by the token. The access token must be acquired by using one of the already presented grant flows. It should be highlighted that all but one of the API calls are end user specific. The end user is identified based on that access token. The only exception is the aforementioned `addUser` API call for which an application client specific token should be used, i.e., a token should be obtained for the application itself through the client credentials authorisation grant flow. Thus, in this case, the access token must be an application specific token instead of one obtained from a user.

Table 3.3 presents all the calls that can be used actually to deal with the personalisation of an application. The first one, `getPersonalisation`, is used to request the personalisation option (which corresponds to the profile of the end user in question) for a specific personalisation instantiation.

Operation	Method	Input params	Output (JSON object)
/getPersonalisation	GET	personalisation context (optional)	*status* property equals to success or error and a *message* property with the status. If successful, the personalisation option is returned under the *result* property.
/getPreference	GET	preference	*status* property equals to success or error and a *message* property with the status. If successful, the resource value is returned under the *result* property.
/setPreference	POST	preference option	*status* property equals to success or error and a *message* property with the status.
/setPreferenceValue	POST	preference value	*status* property equals to success or error and a *message* property with the status.
/getResource	GET	resource context (optional)	*status* property equals to success or error and a *message* property with the status. If successful, the resource value is returned under the *result* property.
/setResource	POST	resource Value context (optional)	*status* property equals to success or error and a message property with the status.

Table 3.3 - HTTP Personalisation API calls: X-Users based operations.

The rest of the API can be divided into setters and getters operations for the resources values that have been chosen by the developers to be taken into account in the application's personalisations. There is a different way of dealing with preferences and interactions stream resources. Thus, we kept that distinction in this API and both resources types have dedicated calls. The setters are used to update them. Preferences have two different options for doing that since a preference is defined both by a semantic meaning and an enumerated type value. Therefore, its update can be done by using one or the other as the parameter in the call. On the other hand, an interaction stream resource can be updated within a specific context, which should be indicated too. Getters are operations that were not initially included in CAPE, and they can be used by a developer who needs to know the current value of a resource for an end user. This is a case of an extension point with respect to the CAPE's

API that was detected during the platform's development, which goal is to facilitate even more the work of developers.

Moreover, if context data is used to form a parameter, then a JSON object with key-value pairs (KVP) of string type containing data about a user's context should be added. Such context can be related to the current time for a user or the geographical coordinates of where s/he is, which are inherited from CAPE. Besides these two, other two additional context types were added with the P^{2}MUCA development, which are: `intervalName` and `valueName`. A context resource can be defined as having one of those types and can be passed in a context map, being a generic way of allowing third-party developers to define and manage new context types that are not built-in into the platform. The interval type allows a numeric interval with a '-' as a separator (e.g., 1-6). On the other hand, the value type just matches any passed string. An example of the value type would be the context with the `key` *place* and the `value` *office*, indicating that the user is currently at the office. Both options can be seen as meaning some kind of context that may make sense in the scope of the application. Snippet 3.4 illustrates an example of a JSON-encoded context map. It includes all of the supported types, but only for illustration purposes. A real application implementation would only use the necessary context types and values, i.e., the ones that would make sense in its scenario to reach the intended personalisations.

```
{
        time: "12:10",
        latitude: 38.708056, longitude: -9.138333,
        valueName: "office"
        intervalName: "3",
}
```

Snippet 3.4 – JSON-encoded context map as an input param for an API call.

3.4.4. The Web UI of P^{2}MUCA

The user interface of P2MUCA consists of a website for both end users and developers (Figure 3.17). Users can register (see Figure 3.18), create applications and manage permissions given to third-party applications. The two main features sets available to users are dedicated to the management of:

1) [P^{2}MUCA user as a developer] applications that are being personalised following the solution provided by the platform;
2) [P^{2}MUCA user as an end user] permissions given to third-party applications.

Both options are available to any user since there is currently no distinction between end users and third-party developers. This approach is intentional since a developer may also be an end user of applications, starting with the ones s/he develops. The only downside is that regular end users are presented with additional options that they may never use. However, this can be overcome in future versions of the Website UI if we decide to hide those advanced options for developers in the cases in which the user is not a developer. We may add a procedure that requires the user to objectively request the activation of the developer options under certain conditions.

It is important to note that the platform allows the user to associate a Facebook and/or Twitter account with her/his P^{2}MUCA account. Thus, the system supports login through Facebook and Twitter credentials. The registration process can also be triggered by using one of those external accounts, which means that some of the data will be automatically filled in the registration form and the new user will automatically have the external account associated with her/his profile. A confirmation e-mail is sent to the newly registered user with an activation link, just like in many other websites. That link must be followed to activate the account before the user can log in to the P²MUCA Website. This integration with commercial consolidated platforms that are considered state-of-the-art in their field promotes the agility, responsiveness, and ease of use of P²MUCA.

Figure 3.17 – P²MUCA Website UI: initial interface.

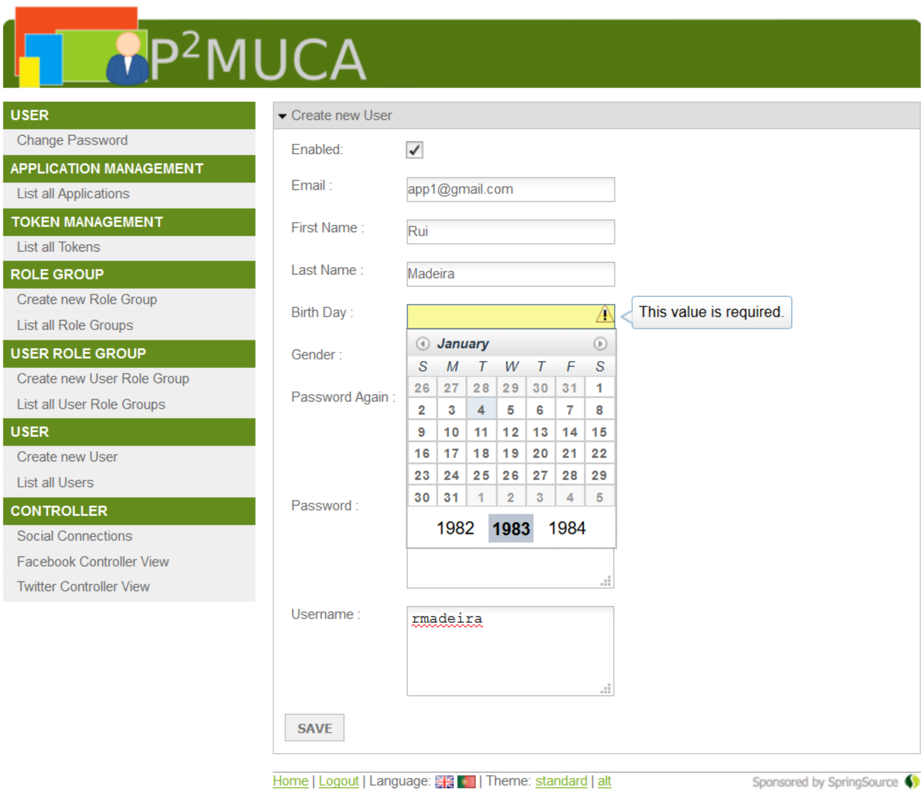


Figure 3.18 – P²MUCA Website UI: user registration interface.

3.5. Evaluation Methodology - Studies Overview

The implementation of a cloud-based platform that integrates a modelling framework driven by a generic model, with the corresponding DSL, for supporting the development of personalised applications is a very challenging endeavour. In addition to presenting a suitable design to engage potential users at all levels, the research requires careful evaluations in order to understand if the solution works as an adequate and efficient tool for personalisation purposes, not only towards developers, but also towards the end users.

It is important to note that, in the studies with the developers, we were not particularly interested in evaluating the performance issues of P²MUCA as a potential commercial solution, neither the language characteristics of X-Users/XUPIL, by following the three general criteria for languages evaluation (Sebesta, 2010), which are readability, writability and reliability. The goal of these criteria is to provide a consistent way of evaluating and comparing General-purpose Programming Languages (GPL), but some of them may be used to evaluate languages with restricted domains, such as DSLs. However, characteristics that serve those criteria, like "simplicity" or "expressivity", can convey a lot if assessed through user studies, along with other dimensions. From this perspective, DSLs can be regarded as similar to Human-Computer Interaction (HCI) as they are meant to close the gap between the domain experts and the solution computation-platforms (Barišić et al., 2011). Most of the requirements concerning the evaluation of UI are associated with usability (Barišić et al., 2011), which is defined by quality standards in terms of achieving the Quality in Use. Moreover, designing useful systems has long been cited as one of the primary goals of the user-centred design. Usefulness can be defined as "the extent to which a system's functions allow users to complete a set of tasks and achieve specific goals in a particular context of use" (Macdonald, 2012). Development solutions are tools to be used in a useful way by someone, by their end users. Thus, there is no reason why approaches usually applied to the domain of HCI for user experience evaluation should not be applied here too (Cardoso, 2016). Ultimately, the judgement of developers about their daily work tools is the most important aspect to take into account in our research. Therefore, we conducted user studies to evaluate the usage of our solution by developers, such as (Cardoso, 2016) did successfully in research works, each addressing particular aspects of the X-Users/XUPIL-CAPE-P²MUCA triad, assessing how the solution may respond and be useful to the developers' needs.

Besides the user studies with developers, others were conducted with potential end users in order to evaluate mainly the acceptance of the personalised prototypes by the end users. The evaluation process even included a study involving experts from an application's domain in order to understand how they look into, and may participate in, the personalisation process, but it included mainly studies with common end users, both in the laboratory and in the field. We followed techniques that are suggested in (Dix et al., 2003). Data were obtained through the observation of the users interacting with the prototypes and utilising query techniques, such as interviews and questionnaires.

Six prototypes were used to test the solution and support the evaluation of its components as a whole, on the one hand, and separately, on the other hand, at different moments of the studies. The first study was directed to the developer side, based on the LEY project, then the following three studies (presented in chapter 5) were made at the end user side and, finally, the cycle was closed with the last study being focused, once again, on the developers (Figure 3.19). The last one has a methodology similar to the first one, conducted with independent developers using a prototype developed by students and called 'Walker', in a first part, and then with the developers involved with us in the development of all prototypes used in the research, in a second part.

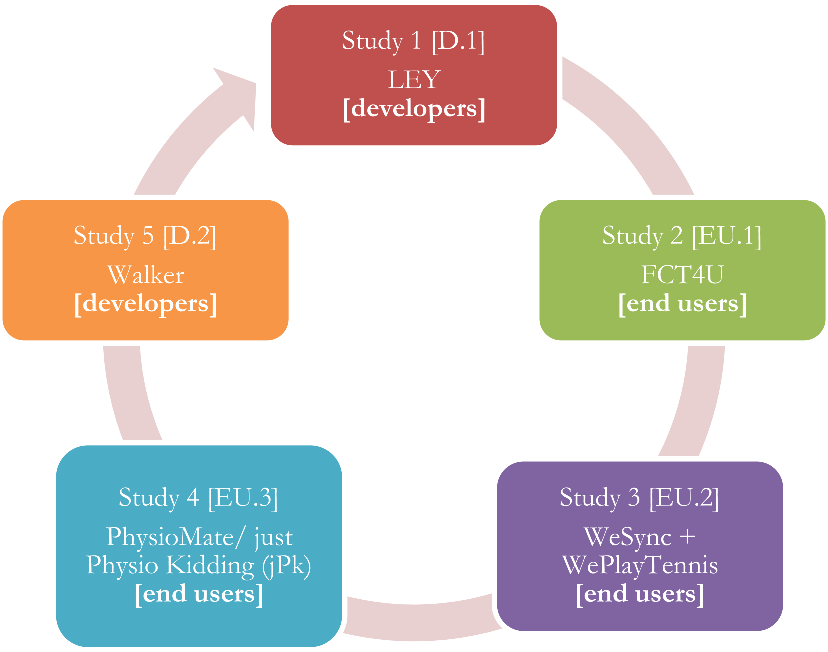


Figure 3.19 – Cycle of the evaluation process comprised of five studies.

The evaluation was organised in those five main stages that are summarised in Table 3.5, where we can find the prototypes and user types involved in each study and what was the assessment focus on each one. The first and the fifth studies (both described in Chapter 4) involved two phases of evaluation with developers (two entries in the table). The "All prototypes" entry refers to the aforementioned evaluation made by the developers that worked with us in all the prototypes.

Study #	Which prototype?	What User Type?		What to evaluate?					Experts involved?
	Implemented Application	Devs	End users	X-Users + XUPIL	CAPE	P^{2}MUCA	Context aware	User profile across apps	
1	LEY	X		X	X		X		
	Developers' apps			X			X		
2	FCT4U		X		X		X		
3	WeSync/ WePlayTennis		X		X			X	
4	PhysioMate/ "just Physio kidding" (jPk)		X		X				X
5	Walker	X		X	X	X	X		
	"All prototypes"			X	X	X			

Table 3.5 - Prototypes and users involved in the studies and what was primarily evaluated in each one.

3.6. Summary

This chapter presented the model-driven solution for the personalisation of the human-computer interaction across different applications towards a Ubicomp based world. The solution was designed to address the following problem.

The process of choosing the appropriate HCI, composed of content, interfaces, or services, for each user is still a critical challenge in many applications of Ubicomp scenarios. The challenge is critical since the process must consider not only what to offer, which is the composition of the final HCI, but also the aspects, features, and conditions that are important to know who the user is in a context of use of an application. Consequently, the offer of the right personalised experience is a challenge due to different issues, such as user interests, heterogeneous environments and devices, dynamic user behaviour and user privacy. Some important specific challenges are related to inferring relevant information about a user, aggregating and integrating such information effectively into long-term user models while achieving user model interoperability, and providing pervasive information access in a personalised manner.

The concept and the solution's main components were introduced at the beginning of the chapter in the Introduction (section 3.1). Our solution is driven by the proposal of a generic model to promote interoperability in the personalisation of applications regardless of the intended Ubicomp scenario (Section 3.2). The main goal is that the same model may be followed and instantiated by different applications, even if from different domains. The model is named X-Users and is the core component of the solution. In the same section was presented an XML-based description language, XUPIL, which was designed to be the standard format for the definition of the rules specifying the instantiation of a Ubicomp application's personalisations.

The second component presented in section 3.3 is the modelling framework, named Context-Aware based Personalisation Environment (CAPE), which encapsulates the X-Users model and integrates three main modules: Web services interface, personalisation API, and user profiling module. CAPE, and consequently the model, is provided by a cloud-based platform (P²MUCA) to support developers in the implementation of personalised solutions, considering different contexts of use, addressing dynamic user behaviour and user privacy, and aiming a high usability level (section 3.4).

Figure 3.20 illustrates how the concept was built from the inside out (or from the lower to the upper layer if thinking about the solution's architecture), with the model being the initial and central component. Then, it was encapsulated by CAPE, which was later integrated by P²MUCA in order to better support developers and provide end users with data management and privacy features.

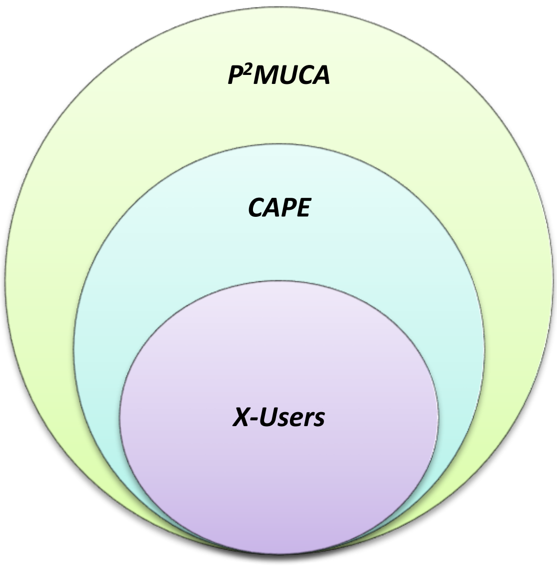


Figure 3.20 – Solution for HCI personalisation built from the inside (the model) out (the platform).

4. Experimenting the Personalisation Solution (Developers-driven Evaluation)

"Life is a gift. Experience is the beauty."

(Yogi Teas, 2017)

This chapter presents the conducted studies with developers in order to respond to the research questions raised previously. We have been involved in the development of different applications that worked as proof of concept prototypes in the studies throughout the different phases of the evaluation process, which was presented in section 3.5 of the previous chapter. The current chapter includes two main sections, each one presenting a conducted study with the corresponding prototype used to support it. They are the following:

- Study #1/D.1 – First attempt of applying the personalisation model [LEY – persuasive pervasive gaming on domestic energy consumption-awareness] (section 4.1);
- Study #5/D.2 – Asking the developers [Walker – people counting in indoor spaces] (section 4.2).

4.1. Study D.1 – First Attempt of Applying the Personalisation Model

This section contains the description of the study that has started right after the conclusion of the design of X-Users and its DSL, the XUPIL language. The first goal of this study was to have a first (preliminary) evaluation of both X-Users and XUPIL as a whole. The user study was directed to developers with different levels of experience, but it was very important to have participants with little experience in the field. XUPIL was designed to be simple, but as much as the concept allows it, thus we wanted to target these participants to assess if that goal was achieved. If the results of such an assessment are positive, proving the language and its underlying model are indeed simple, then they can be extrapolated to developers with more experience. The latter are expected to show greater ease due to their familiarity with development concepts and programming techniques.

The tests of this first study were conducted having the LEY prototype as a case study (described in section 4.1.1). At the time, we were participating in the development of the LEY prototype under the funded project DEAP[1], which goal was the development of pioneering environmental persuasive interfaces to motivate citizens positively to become more environmentally responsible in their everyday life (Romão et al., 2012). Therefore, it made all the sense to use LEY as the first proof of concept and to support the evaluation with the developers (in section 4.1.3). Although the study was primarily based on LEY, we also included a group of participants that could choose any other application they had developed in the near past or were currently developing (details presented in section 4.1.3). Additionally, being LEY a context-aware system, the integration of context was studied and evaluated in this first study.

[1] DEAP – Developing Environmental Awareness with Persuasive Systems: funded by FCT/MCTES under the grant PTDC/AAC-AMB/104834/2008.

At the base of the study with the developers was the definition of a set of personalisation instances to enhance LEY (presented in section 4.1.2), which allowed us to test both the model and the language and prove their viability. Additionally, through HTTP requests, LEY was able to submit data to the CAPE's server, and also send personalisation requests for each different personalisation instance, which allowed us to test and validate the solution's architecture and CAPE's user profiling module.

At the end of this section, we present the final remarks that summarise the results of the tests with the developers and how CAPE responded to the personalisation requests (section 4.1.4).

4.1.1. LEY: Persuasive Pervasive Gaming towards Domestic Energy Consumption-Awareness

LEY (Less energy Empowers You) is a persuasive pervasive-based serious game approach to help people understand domestic energy usage in order to foster better consumption habits (Madeira, Silva, et al., 2011). The mechanics of the game are based on real-time domestic energy consumption data, which are gathered using smart energy monitor devices.

Motivation

Over the years, it has been a challenge to find ways of educating people to have better habits, especially on domestic energy consumption. A lot of research was done on the topic of energy reduction. On the one hand, if monthly utility bills are ineffective at persuading for sustainable energy consumption habits (He and Greenberg, 2009), on the other hand, usual commercial off-the-shelf real-time energy monitor devices do not apply motivational techniques for energy savings. HCI research groups have been devoting a lot of effort in designing and developing systems focused on feedback technologies. Results show that technology-enabled consumption feedback can promote awareness and lead to energy savings (Woodruff and Mankoff, 2009). There are important factors for a successful behaviour change and the design as an interaction tool (Fogg, 2009) is an essential part of persuasive applications that work as behaviour change support systems (BCSS) (Oinas-Kukkonen, 2013). This way, some persuasive feedback systems appeared, with an interesting emphasis on serious game approaches. Power Agent (Bang et al., 2007) and Power Explorer (Bang et al., 2009) are casual games directed to the teenagers for encouraging their long-term behaviour change. The focus is on real-time feedback, using custom-built energy sensors and a mobile phone game. EnergyLife (Gamberini et al., 2011; Gamberini et al., 2012) is another system for households providing appliance-level data through a mobile application and feedback on the total energy consumption using an ambient display.

However, environmental sustainability involves efforts such as informing individuals' personal choices in consumption and behaviour (Woodruff and Mankoff, 2009). Moreover, He and Greenberg present a set of guidelines for a more motivational presentation of information, highlighting "People are more motivated to act when presented with vivid and personalised information" (He and Greenberg, 2009). Froehlich introduced ten design dimensions of feedback technology and one of them was related to the offering of highly personalised recommendations tailored to the sensed energy usage in the home (Froehlich, 2009). This personalisation factor cannot be explicitly found in the aforementioned projects, although the EnergyLife's authors are improving it in order to tailor the game to its users, which was an indication obtained with the conducted tests (Gamberini et al., 2011). We can find a serious game approach towards saving energy in an office environment, instead of the household, specifically designed for reduction of plug-loads in a mid-size commercial office setting (Orland et al., 2014). The authors refer to the use of a web-based game application, Energy Chickens, which should present a personalised aspect, but it is not clear how it is done since there is no evidence of personalisation techniques usage.

Motivated by the aforementioned, we expect that integrating personalisation into LEY will potentiate even more its goal of changing potential negative habits and supporting the positive ones to a whole new level. As presented before, a dominant driver for personalisation is to help people coping with information overload, but another one is the appearance of smart environments that sense and respond to their inhabitants, such as LEY. This pervasive computing approach can be more effective with a personalised game, matching the individual's current context, implicit behaviour and preferences.

System overview

The system consists of three main components (see Figure 4.1):

1. a sensor platform for real-time energy consumption monitoring,
2. the supporting server providing a Web-based information system (IS),
3. the mobile application (app).

The sensor platform uses the Current Cost EnviR[2] wireless home real-time energy monitor, but others can be used provided they serve the same measuring goals. The EnviR device connects its monitors directly to the web using a 'bridge', without needing a dedicated computer. We use the Xively IoT platform (formerly known by Pachube) as a fast and easy way of making the sensor data available in real-time for the mobile game. The web server and IS component is implemented in MySQL, PHP and Java technologies, based on the Spring MVC platform. The server processes the sensor data and stores them in a game database, which also contains all the game-related data. The user can connect to the community-based IS to obtain detailed information about, for instance, her game results, household's energy consumption, and friends that are using the system.

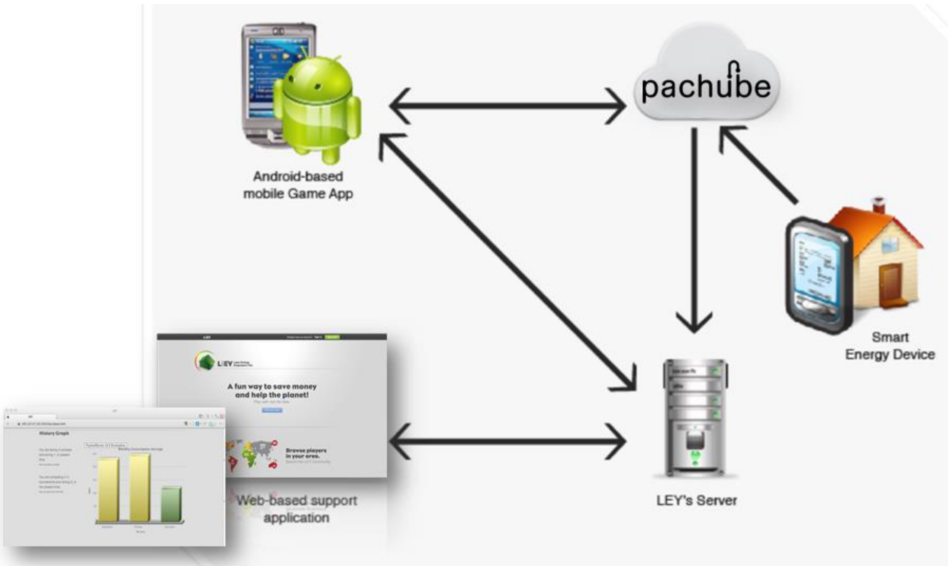


Figure 4.1 – LEY: The system architecture.

A first mobile game app is implemented in Java for the Android platform. It connects to the server to obtain and store the necessary data, but it can receive data directly from pachube to implement "wakening-up functions". These functions are launched when abnormal real-time sensor values appear or calendar events occur.

[2] http://www.currentcost.com/product-envir.html

Rationale and design

The application is considered as being a pervasive serious game as it implements game mechanics to entertain and engage users, being educational, context-aware and pro-active. It is mainly directed to an adult user, but it is also suited for teenagers. The game's user interface is based on the house represented by an avatar, which acts as the main game character. Each household's inhabitant that uses the game app can play with a different avatar. At the beginning of the game, each user's avatar is assigned with a profile, which is defined according to some factors, such as the housing typology, connected appliances and their average number by room, the number of inhabitants, and last year's consumption. The initial profile assignment allows houses with different typologies to compete between them in the LEY context. The first week's consumption values are used to help sharpen the profile.

The game presents a single mode and two secondary competition modes (combat and tournament). The first one corresponds to the status view, in which the user can find information immediately and intuitively (e.g., Figure 4.2). It intends to be easy to read with the visualisation of data represented by the graphical elements of the user interface in order to provide a fast overview of the user's status, which is composed of: a consumption level, the global LEY's score, and the house's avatar mood. Actually, the background represents the current weather conditions for the household's location.

The user's challenge is always to bring her/his house to the best consumption level and to obtain the best possible score over time. The consumption levels are defined according to the official energy efficiency rating, which presents the energy efficiency of dwellings on a scale of A (most efficient) to G. Therefore, a user competes against her/himself to achieve a level A for her/his virtual house, the avatar, while saving real energy as much as possible. Figure 4.2 illustrates a case in which the household is currently with a level C. On the other hand, the score value intends to be more comprehensive also taking into account variables in the context of the user interaction with LEY. Thus, the score is obtained through the following values: real energy consumption values; responses to regular quizzes; competitions results; and invitations of people to the game; among others.

Another important functionality is the avatar interaction with the user, showing its satisfaction level to the current player's consumption actions. For instance, the avatar mood is green if the consumption is below a predefined threshold value that is below the mean value for the current period (Figure 4.2), while yellow means it is around the mean value (Figure 4.3) and red is for a current consumption value above the mean value, which deserves a notification (Figure 4.4). The avatar appears in "wakening-up functions" (in the form of notifications) when occur, for instance, the aforementioned abnormal energy consumption values (Figure 4.6) and it also presents regular quizzes.

In the combat competition mode, a user can challenge any other user, entering into a one-on-one competition. This way, an environmental sustainability-based quiz will be launched to each one of the "fighters", which receive points according to the result. Unlike the combat mode, a tournament involves various users (between three and ten of them). A single user creates it and invites other participants to enter the competition during a predefined period of usually one month, while the tournament classification is being updated in real-time (Figure 4.5). After its conclusion, the final classification is obtained according to a score based on what we call 'the biggest loser' concept, which is based on the global score at the beginning of the competition ($score_{start}$) and the end of it ($score_{end}$), through the following expression: $(score_{end} - score_{start}) / score_{start}$.

At the end of a month, persuasive information related to the energy consumption results is displayed both in the game and in the IS area (a more detailed version). The user also has access to a statistics menu where s/he can consult diverse interaction data. The available statistics are the current energy score, the average energy consumption, number of played combats, number of rejected combats, number of played tournaments and number of rejected tournaments.

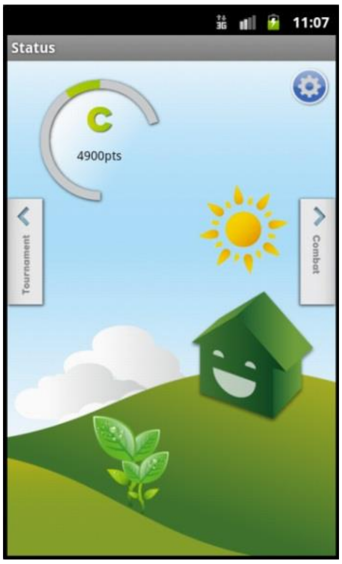


Figure 4.2 - LEY UI: status view (level=C, score=4900 pts, avatar mood=green).

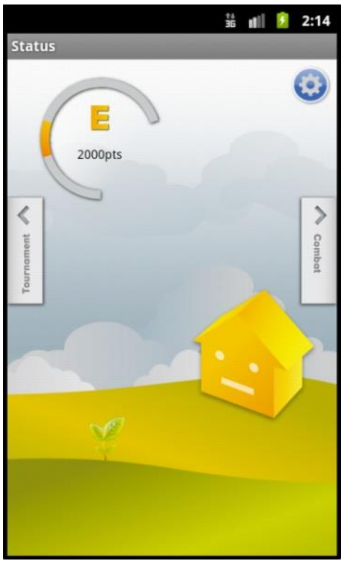


Figure 4.3 - LEY UI: status view (level=E, score=2000 pts, avatar mood=yellow).

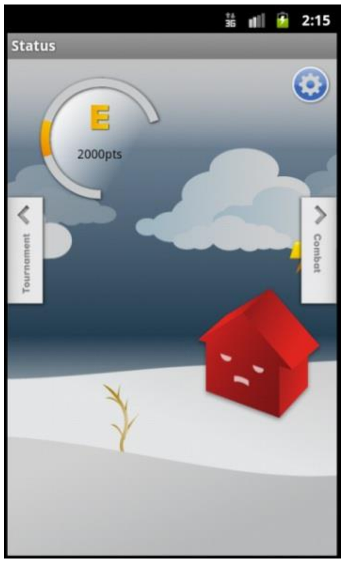


Figure 4.4 - LEY UI: status view (level=E, score=2000 pts, avatar mood=red).

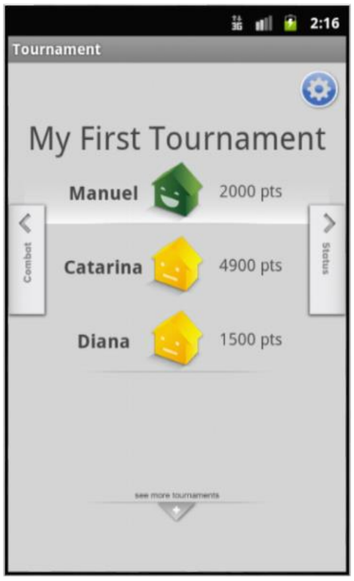


Figure 4.5 – LEY UI: tournament screen showing the classification ordered by the avatar mood).

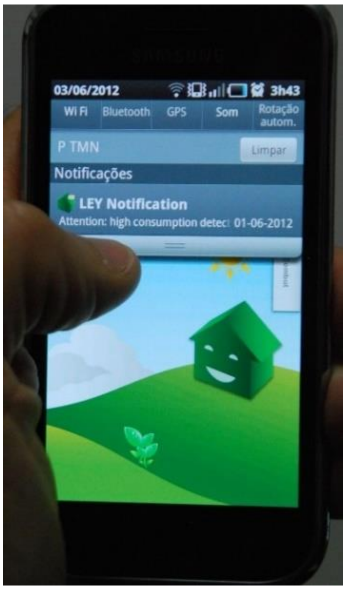


Figure 4.6 - LEY UI: a notification is received communicating that an abnormal high-consumption value has just been detected.

4.1.2. Personalisation Design for LEY

LEY was used for a first attempt of applying the personalisation model to one concrete case of a prototype that was being developed with a specific goal (Madeira, Vieira, et al., 2012), other than being used only for testing our solution. Therefore, after one first analysis of what LEY offers to its users, we found out several possibilities for personalisation based on collecting different useful data. For instance, in order to use the app, a user needs to log in and, usually, this may be a good resource for personalisations since users that constantly log in may present a different profile in comparison to users that log in once in a while. The former tend to be more interested and dedicated than the latter. However, considering only the number of logins may be not enough. There may be users that are constantly logging in and logging out and other users that rarely do so, but the latter might spend more time using LEY for each login session. This situation indicates that the time spent logged in can also be important for generating a certain user profile and identifying personas.

Another interesting interaction data resource is the number of times the user consults the statistics menu. If a user does that often, it probably implies that there is more dedication or interest on her/his LEY performance. If on the one hand the single-player mode provides data that can be potentially used for designing certain LEY's personas towards personalisations options, on the other hand, the competition modes provide other kinds of data that can be used to design other personas or improve the previous ones. Thus, it is possible to collect and use the number of combats and tournaments played by the user, the number of combats and tournaments rejected by the user, the category level of each user's opponents, how much time a user takes to complete a combat or a tournament, and many other interaction data.

After observing the personalisation possibilities, we decided to select the resources set described in Table 4.1 in order to advance with the instantiation of four different personalisations for the LEY's mobile app: P1) `status view background`; P2) `house avatar`; P3) `alert level`; and P4) `first screen`. The personalisation instances are described as follows and the XUPIL script that defines LEY's personalisation is presented in appendix A.1.

R	Resource	Description
Interaction stream resources		
1	**loginTime**	[sum] Total log time by the user.
2	**numberCombats**	[increment] Number of combats played by the user.
3	**numberTournaments**	[increment] Number of tournaments played by the user.
4	**numberRejectedCombats**	[increment] Number of combats rejected by the user.
5	**numberRejectedTournaments**	[increment] Number of tournaments rejected by the user.
6	**statisticsMenuCount**	[increment] Number of times the statistics menu was consulted.
Context resources		
7	**weather**	[external service] Current weather for the location of user's house.
8	**currentTemperature**	[interval] hot (26-40); warm (11-25); cold (-10-10)
9	**routineHoursOfAccess**	[interval] businessHours (09:00-17:00); leisureHours (17:00-9:00)
Preference resources		
10	**prefStatusView**	[question_MC] User chooses from: 0.5; 0.75; 1.0; 1.25; 1.5

Table 4.1 – LEY: resources data on which personalisations are based.

P1) status view background

This personalisation combines the usage of the current weather condition with how social a user is. In the LEY context, the social meaning is related to the contacts of the user and how s/he socialises with them through competitions. A group of the user's contacts will appear as the house's neighbourhood if the user is considered competitive, but it does not appear in case the user is non-competitive. The house's neighbourhood will indicate how the user's current energy consumption is doing against the average energy consumption of the user's contacts represented in the neighbourhood. We consider that if the user has a competitive profile, s/he will positively face the challenge of being under a peer pressure scenario.

Therefore, this personalisation has two basic scenarios, two personalisation options, resulting from the two personas that are represented here: 1) `non-competitive`; 2) `competitive`. They are determined using the parameter `competition_level` that uses the following interactions stream resources (expression (5)): `numberCombats`; `loginTime`; `numberTournaments`. We gave a higher weight to the tournaments parcel since we believe it represents better the social meaning.

$$0.4 * \left(\frac{numberCombats}{loginTime}\right) + 0.6 * \left(\frac{numberTournaments}{loginTime}\right) \tag{5}$$

Moreover, the external service of weather is used to change the background even more, but this time according to the real current weather condition that is felt in the house's region. Given the location of the user's house, it is possible to obtain a very detailed description of the current weather conditions. This can work as additional useful information because the energy consumption of a user is also related to the weather conditions. For instance, during a snowstorm in winter the users will tend to use warming devices to compensate the heat discrepancy, which increases energy consumption. With the inclusion of the weather context, we will have six personalisation options for the background instead of the two initially thought (see Table 4.2). They are obtained through the combination of the two profiles of the `competition_level` with three possible values obtained from the weather service.

		competition_level	
		competitive	**non-competitive**
external weather	**sun**	**neighbourhood sun**	**isolated sun**
	clouds	**neighbourhood clouds**	**isolated clouds**
	rain	**neighbourhood rain**	**isolated rain**

Table 4.2 - LEY: six personalisation options for the "status view background".

P2) house avatar

The house's avatar is the main indicator of the user's current energy consumption (the colour indicates it). It is probably the most-representative element of the interface. It is interesting to make it more expressive assuming the user can see it as representing her/his energy consumption behaviour. As it was referred previously, different users will have different interests and behaviours, which is why the house is a perfect candidate to express different personalities among the users. In this case, the user should be classified as `defensive`, `offensive` or `coward`, which are the three possible cluster profiles that represent the personas considered at this point. The computed user profile is used to adapt the appearance of the user's house icon with the inclusion of a hand shield, a sword or a white flag.

This personalisation uses the aforementioned parameter `competition_level` in combination with another parameter called `fearful_level`, which is based on the resources (expression (6)): `loginTime`, `numberRejectedCombats`, `numberCombats`, `numberTournaments` and `numberRejectedTournaments`. The parameter is defined with the two options (two clusters) that represent personas: 1) `non-fearful`; 2) `fearful`.

$$0.65 * \frac{\text{numberRejectedCombats}}{\left(\frac{numberCombats}{loginTime}\right)} + 0.35 * \frac{\text{numberRejected}Tournaments}{\left(\frac{numberTournaments}{loginTime}\right)} \tag{6}$$

The house avatar will be personalised according to one of the three personalisation options that are defined by the combinations of the options of the two parameters/profiles - `competition_level` and `fearful_level` – as illustrated in Table 4.3.

		competition_level	
		competitive	non-competitive
fearful_level	fearful	coward (fearful)	
	non-fearful	offensive	defensive

Table 4.3 - LEY: three personalisation options for the "house avatar".

P3) alert level of notifications

The LEY mobile app sends regular alerts to the user every time the actual energy consumption of her/his house exceeds a certain threshold related to the consumption mean in that period. Although it is a good idea to warn the users when this happens, it is also interesting to personalise those alert messages according to the behaviour of each one.

This personalisation instance integrates context segmentation to obtain more accurate results. It uses two context resources – `currentTemperature` and `routineHoursOfAccess` - as explained in the subsection *Personalisations* of section 3.2.2 and according to expression (3). The `currentTemperature` was specified in the personalisation with a weight of 0.4, while `routineHoursOfAccess` weights the remaining 0.6. The context has a total weight of 0.25 against 0.75 of resources data-independent of context, which means that the interaction resources should not be treated in the same way.

The personalisation uses two parameters, which are the `competition_level` and the `user_level`. The first one was already described in personalisation P1. The second one is used to describe how much a user uses the application. It defines two clusters to profile a user as 1) `basic`, or 2) `advanced`, based on the resources `loginTime` and `statisticsMenuCount` through the expression (7).

$$0.8 * loginTime + 0.2 * \text{ statisticsMenuCount} \tag{7}$$

The two parameters are combined (see Table 4.4) to personalise how often a user receives alerts from the system and how much persuasive those alerts should be. Concerning the user level, a more advanced user tends to: receive alert messages more often; have a smaller threshold of energy consumption alert; receive a more detailed alert message, providing some additional data such as the difference between the current energy consumption and the user's average. There is also another type of alerts, which concerns the competition level of a user. If the user was not in any competition for a period longer than a defined interval, then an alert is sent to inform her/him about this particular situation. The referred interval and the alert type depend on the user profile (competitive vs non-competitive).

		competition_level	
		competitive	non-competitive
user_level	basic	multi-basic	single-basic
	advanced	multi-advanced	single-advanced

Table 4.4 - LEY: four personalisation options for the "alert level of notifications".

P4) first screen

When a user enters the LEY mobile app s/he is faced with the authentication page and then goes to the main app's interface. This interface is divided into three screens: status, combat and tournament. The user can swipe between those three different screens, but maybe some users would like to open LEY and go straight to the desired screen. Maybe the analysis of how many combats and tournaments the user has done in the past will help understand if s/he is more interested in a specific screen.

Therefore, in this personalisation, there are three options according to the three main screens of the app. This personalisation instance is based on two parameters. The first one is a refinement of the competition_level, but this time with the inclusion of a preference resource called `prefStatusView`, which gives additional weight to the case in which the user prefers the status screen. This parameter (`user_profile`) is defined by expression (8) and indicates if the user would prefer to see the combat or tournament screen instead of the status, or the opposite, presenting two options/personas (clusters): 1) `status`; 2) `player`.

$$\frac{numberTournaments + numberCombats}{loginTime} * \left(\frac{1}{prefStatusView}\right) \quad (8)$$

Finally, a second parameter called `player_profile` indicates the tendency of the user concerning combats and tournaments. Expression (9) represents the parameter and there are two possible options: 1) `tournament`; 2) `combat`.

$$\frac{numberCombats}{numberTournaments} \quad (9)$$

The options of the two parameters are combined resulting in three personalisation options as illustrated in Table 4.5. If the user has a status profile, then it is irrelevant her/his option regarding the player profile. On the other hand, if the user is a player then it is necessary to get what has a higher impact on it, tournament or combat, to launch the corresponding screen.

		user_profile	
		status	player
player_profile	tournament	statusScreen	tournamentScreen
	combat		combatScreen

Table 4.5 - LEY: three personalisation options for the "first screen".

4.1.3. User Study with Developers

The first goal of this assessment moment was to have a first evaluation by independent developers of X-Users and XUPIL as the tool to instantiate it and help understanding better the model and its concept.

Design and participants

The evaluation process followed in the two studies (the current D.1 and the next D.2 in section 4.2.3) is presented in Figure 4.7. The process started with the participants' recruitment, where we analysed the volunteers to group them according to the requirements of the two parts of the study. The variables related to the user profiles that would lead to different results for different groups were controlled (Barišić et al., 2011). We invited developers to participate in this study with the requirement that they were not involved in any aspect of the development of LEY. Since the tests were supposed to be executed by application developers, it was assured that every participant was familiar with programming. As previously mentioned, the tests were comprised of two parts, based on:

1. the LEY prototype, in which participants were told to imagine themselves as LEY developers. The participants would implement the personalisation instances presented in the previous section.
2. the participants' own chosen applications, which they had developed in the near past or were currently developing, having to design possible personalisations for them.

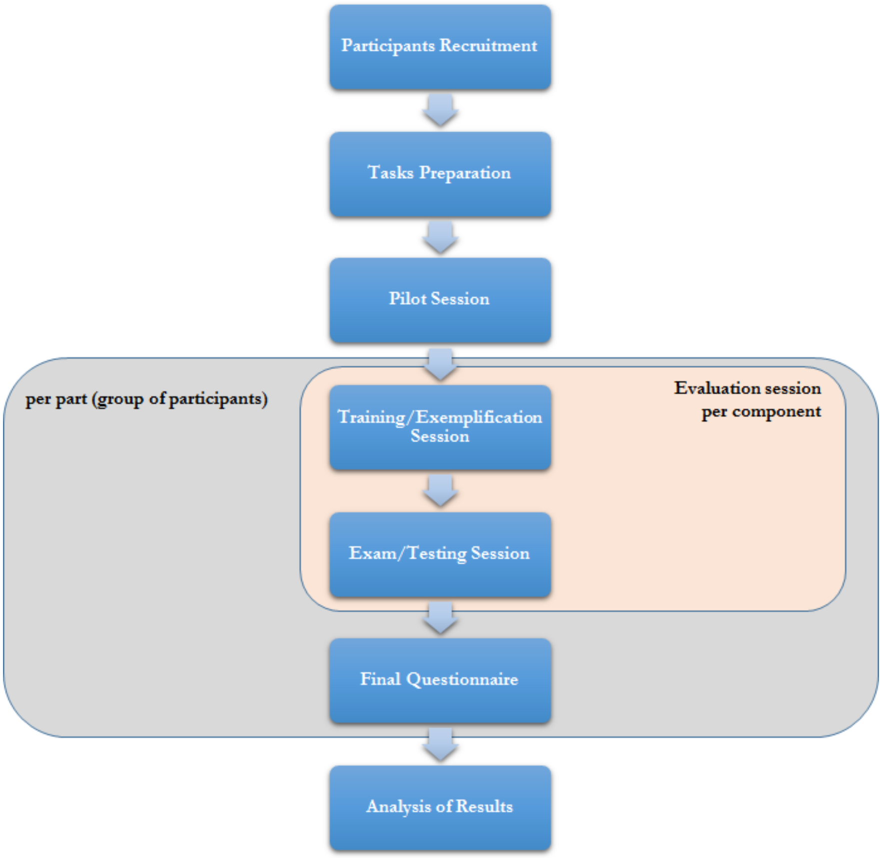


Figure 4.7 - The followed evaluation process steps (adapted from (Barišić et al., 2011)**).**

Regarding the first part of tests, we worked with ten volunteer participants who were junior software developers aged between 20 and 26 years ($\bar{x}$ = 22.9; σ = 2.02). They were from four different areas of the computer science domain (Figure 4.8) and presented different levels of experience (Figure 4.9), but only three of them had more than five years as at the date of the tests. All participants were students completing the master's thesis.

In the second part of the study, the tests involved only five computer programmers aged between 24 and 33 years ($\bar{x}$ = 28.2; σ = 3.31). Six individuals per subset of the population would be the minimum required for a controlled experiment according to (Johnson, 1992). However, typically, the most common usability method is a lab test requiring a relatively small number of participants, which is between four and ten (Tullis and Albert, 2013). It would be good to take a larger number, but it was not easy to find the appropriate participants who could volunteer by the time we wanted to conduct the study. Unlike the participants of the first part, most participants of this second one were working in the Multimodal Systems area (Figure 4.10), with experience in the HCI field, which implies some expertise with high-level development. They did not present many years of experience in their current areas of work (Figure 4.11), but they had more years of accumulated experience, also considering the past years in their previous areas. Still, all of them already had the master's degree.

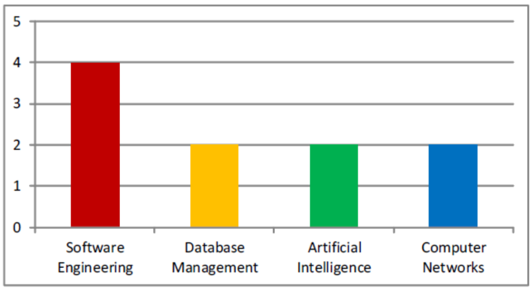


Figure 4.8 – Study D.1: Current areas of expertise of part 1's participants.

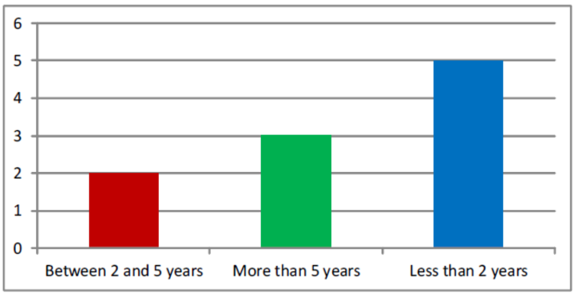


Figure 4.9 – Study D.1: Years of experience of part 1's participants in their current areas.

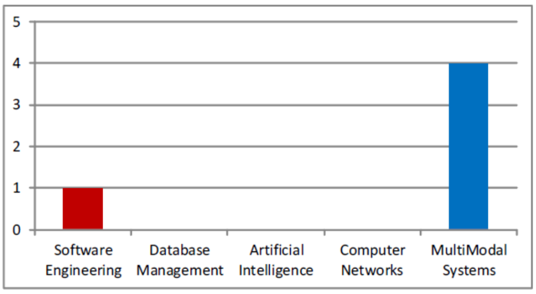


Figure 4.10 - Study D.1: Current areas of expertise of part 2's participants.

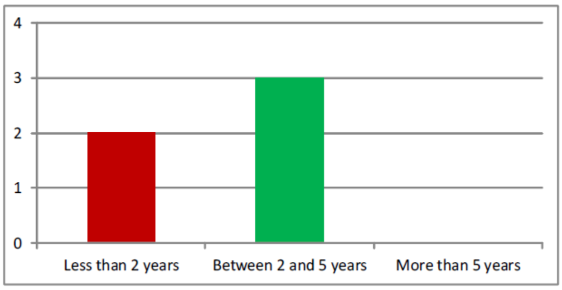


Figure 4.11 – Study D.1: Years of experience of part 2's participants in their current areas.

Regarding the task preparation step of the process, we organised the tests in the two parts, which took place in a lab at our University campus. First, we prepared everything to brief each participant, taking the chance to explain the concept behind X-Users, how to instantiate it through XUPIL, and the functionalities of CAPE, besides its prerequisites. In the second part, the applications chosen by each participant were very different from each other, concerning subjects such as sports news, multi-player fighting games, and augmented reality, for instance. This diversity illustrates that many applications from different domains can benefit from the integration of personalisation using the X-Users based solution. When confronted with the need to personalise the chosen application, by evaluating the reaction of those participants, it was clear that the task would need some time to get them with the appropriate mindset after the initial clash of leading with something new. Nevertheless, after a few minutes, the participants were able to come up with the design of aspects that could improve the application through the integration of personalisation aspects.

Afterwards, the participants would design the personalisations for the application used in the tests, according to the X-Users model, and implement the instantiation file in XUPIL. These were the tasks they would perform in the testing session step of the process. In the pilot session step, we simulated the tests with two individuals (colleagues that would not participate in the study) in order to verify the materials and questionnaire. This rehearsal was important to check that the time constraints and other possible external variables like proper equipment were controlled, and did not interfere with the results (Barišić et al., 2011).

After having everything tested, we proceeded to the evaluation session with each group in the corresponding part of the study. We conducted a training session where we introduced the concept and briefed the participants explaining all the components in test. We took the opportunity to train the participants through some examples. The exam session involved the execution of the aforementioned tasks with special attention to the implementation of the personalisation instances through sentence writing activities according to XUPIL. In order to evaluate unbiasedly, the users tested the same environment and as realistically as possible (Barišić et al., 2011). During the exam session, a researcher observed and took notes of the participants' activities.

Finally, after finishing all tasks of the exam, we asked the participants to fill in anonymously a questionnaire covering some important questions. They were asked for a debriefing in the form of that questionnaire with the goal of obtaining their qualitative perspective about the model-driven solution and XUPIL. The questionnaire was favoured over personal interviews, although the latter appeared informally while observing the testing process. The questionnaire started with a group of questions related to the participants' personal data to characterise them, based on age, area of expertise and experience in the area (data presented above). Next, the questions focused on evaluating our personalisation solution, including feedback about the personalisation concept, the modelling process and the ease of use of XUPIL. As presented in Table 4.6, participants were asked to rate multiple statements, using a five-point Likert-type scale, which ranged from strongly disagree (1) to strongly agree (5). The evaluation process ended with the analysis of the results, which are presented as follows.

Questions
Q1. Have you ever used any third-party approaches or tools to integrate personalisation into applications?
Q1b. In the positive case, which one(s)?
Q2. Do you think there should be more types of external context services?
Statements
The model-driven solution
S1. The core concept "personalisation – parameter – resource" of the model is easy to understand.
S2. The integration of 'context' for personalisation purposes is useful.
S3. Applying personalisation to the application is easy to design.
S4. I would be interested in using CAPE to provide personalisation for future applications.
Using XUPIL
S5. The personalisations instantiation with XUPIL is easy to understand.
S6. The personalisations instantiation with XUPIL is simple.
S7. The personalisations instantiation with XUPIL is easy to learn.
S8. The personalisations instantiation with XUPIL is expressive.

Table 4.6 – Study D.1: Questions and statements of the questionnaire.

Results and discussion of part one

In spite of the lack of experience personalising applications, it was observed that all participants successfully implemented the given tasks related to instantiate the personalisations requested for LEY using XUPIL. The results varied in complexity. We did not record the time that took each participant to write a syntactically and semantically correct personalisation. We registered that some developers took more time than others did. We preferred to give support to the tasks since they were using a new paradigm and following the X-Users model and its language for the first time, helping every time a participant was stuck at some points, mainly at the beginning.

It is noteworthy that all participants indicated that they had never used any tools or approaches to integrating personalisation into their applications or projects (ten in ten responded "no" to Q1). This indicates they have no means to compare our solution with third-party personalisation tools and frameworks. It was the first time they had to think about programming personalisation into an application, and not only programming features and interfaces to be used by the end users for customisation purposes.

Regarding the adoption of other external context services and types (Q2), the results are unclear since half of the participants thought that the current available services and context types are quite limiting, and the other half thought that there was not any other service that could improve the personalisation results. Among their suggestions, there is the adoption of the mobile camera to recognise the current user's surroundings. If applicable, it could be interesting to take into account different energy fares a user might have, which vary according to daytime, and also the usage of GPS coordinates to detect if a user is in any particular place such as her/his workplace, home or other places s/he might frequent in a regular way.

Figure 4.12 summarises the results obtained with the first four statements. It is possible to observe that the feedback regarding the core concept of the personalisation sub-model was extremely positive ($\bar{x}$ = 4.4; σ = 0.66). Almost all participants (othe nly one gave the neutral value of 3) found it easy to understand, with 50% of them strongly agreeing with statement 1 (S1). They considered it intuitive, which is very positive.

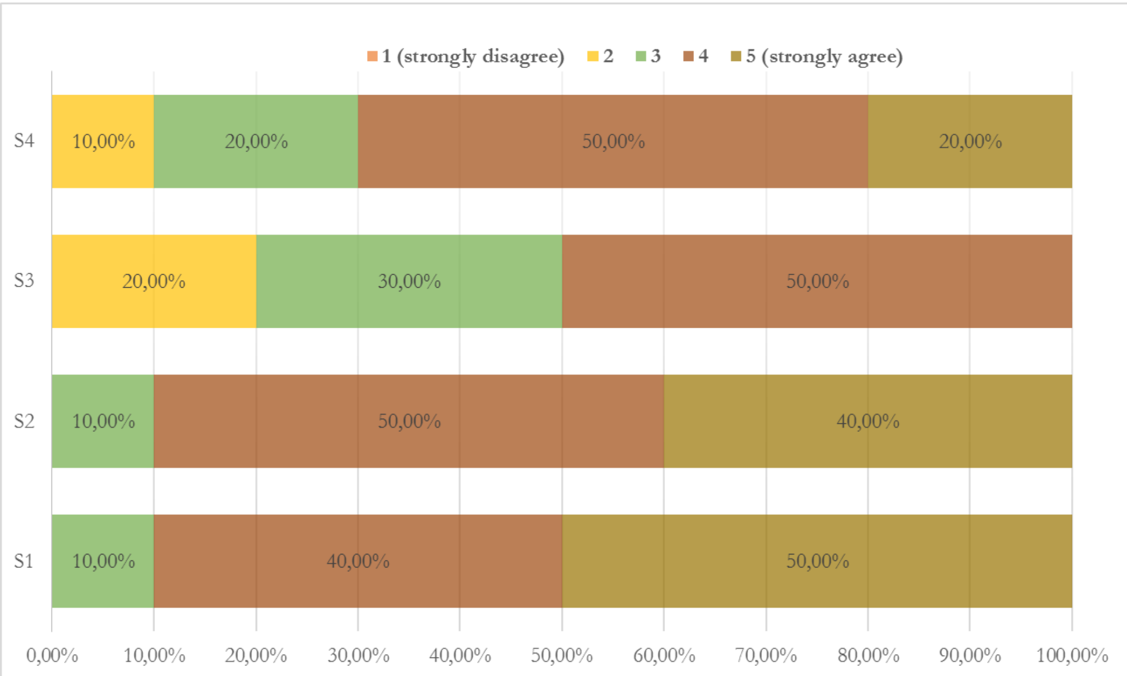


Figure 4.12 – Study D.1: Summary of part 1's results for S1-S4.

Another positive and very interesting result is the evidence that participants strongly accepted the notion of context in order to refine data used to profile the end users of the applications. This acceptance is clearly demonstrated in the results of S2 ($\bar{x}$ = 4.3; σ = 0.64). Nine of the ten participants accepted very well the concept of context integration and segmentation considering it useful. During the experiments, they referred that the usage of context offers a great potential for some specific applications.

The results obtained with S3 ("Applying personalisation to the application is easy to design") show that participants rated their practical personalisation task applied to LEY as reasonably easy, since the results were not as good as the previous ones, with a slightly lower mean value regarding the difficulty level ($\bar{x}$ = 3.3; σ = 0.78). It was expected according to the experience level of the participants and since they had to design personalisations, and some with complexity, for the first time and soon after learning how to do it. Nonetheless, 50% considered easy to design the personalisations and only 20% of participants (two of the ten) disagreed with that.

Finally, seven participants stated interest in using our solution shortly, with two of them being very interested. Only one participant did not manifest interest (rating it with a 2), but nonetheless leaving an open door for the possibility of coming to use it in the future. Therefore, S4 presented very positive results ($\bar{x}$ = 3.8; σ = 0.87) indicating that participants were looking forward to use CAPE in order to help personalising future applications.

The next four statements were focused on the evaluation of XUPIL since it was used by participants to instantiate the personalisations for LEY. Therefore, after doing it, we asked them to respond to those statements in order to understand if the language was suitable for its purpose. Figure 4.13 shows the results were globally positive. However, it is worth noting the tendency to be less positive when we move from S5 to S8.

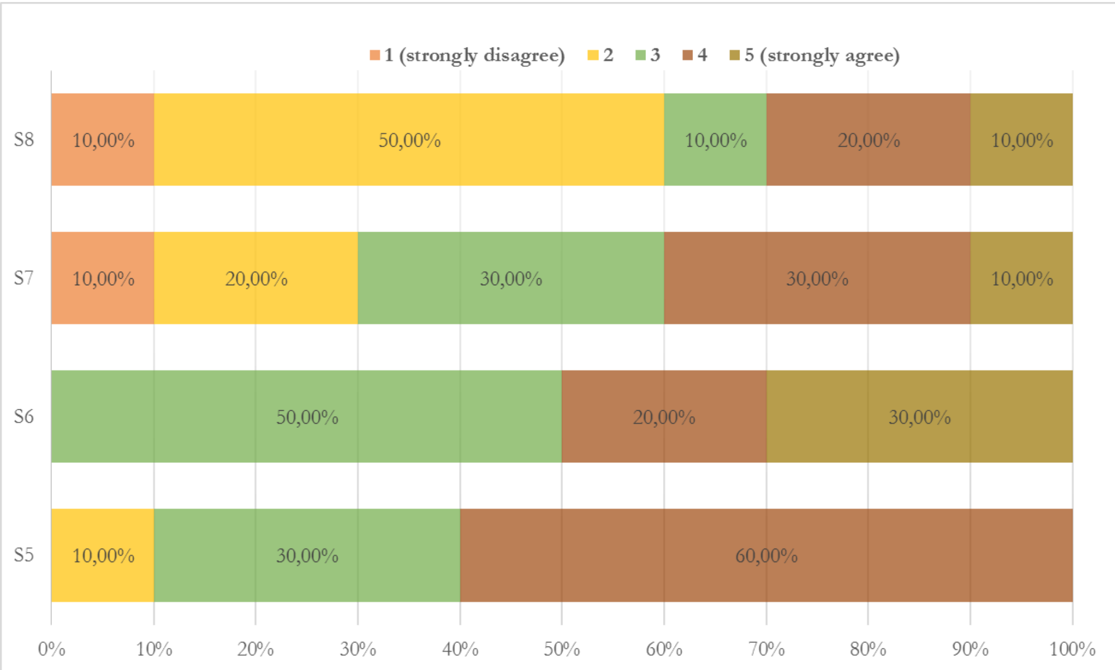


Figure 4.13 – Study D.1: Summary of part 1's results for S5-S8.

Indeed, participants indicated the personalisations instantiation with XUPIL was easy to understand ($\bar{x}$ = 3.5; σ = 0.67), which is very positive, in line with the result for understanding the core concept of the model (S1). 60% agreed with S5 and only one participant stated s/he had more difficulties in understanding how to use the language. 30% were neutral regarding S5. We suppose these results could be due to some inexperience of the participants regarding the use of XML-based languages since we observed some difficulties with regard to XML aspects. We even had better results for S6 ($\bar{x}$ = 3.8; σ = 0.87) with respondents finding the personalisations instantiation through XUPIL simple to accomplish. 30% strongly agreed with the statement and no one responded below 3. After understanding XUPIL, usually, participants could find it more simple to use, so the results increased from S5 to S6.

Nonetheless, globally, the personalisations instantiation with XUPIL was not so easy to learn ($\bar{x}$ = 3.1; σ = 1.14). Statement 7 shows a wide range of results with 30% not agreeing that XUPIL is easy to learn. On the other hand, 40% consider it easy to learn and the remaining are still neutral regarding the needed effort to master the language. This is a reasonable limitation when configuring an application's personalisation, implying that this issue had some influence on their experience.

We found out that participants were concerned with some lack of expressiveness in the process of the personalisations instantiation with XUPIL ($\overline{x}$ = 2.7; σ = 1.19), which can be a critical limitation. Five of them disagreed with S8 and one even strongly disagreed. It is possible to conclude from those results that some would believe it is not easy to write code that clearly expresses the intentions of the developer, mainly for a human reader. S8's results are more negative than S7's, which indicates they can learn it but still considering hard to write instantiations that easily express what is intended in order to be rapidly understood.

Results and discussion of part two

It was observed the developers involved in this second part were very enthusiastic concerning the possibilities of CAPE. They tried to design different personalisations, taking different paths, with context segmentation having a particular interest in being tested. After the implementation part, this second part of tests followed the same approach as the first one, consisting on filling the same questionnaire as in the previous part of tests, which provided some interesting conclusions.

Once again, this second part showed that no participant ever used any third-party tools or approaches to integrate personalisation into their applications (five in five responded "no" to Q1). They only had the experience of implementing isolated cases and very application-specific aspects of personalisation using their own-programmed user modelling techniques. It was nothing generic, global or standard that they could even use across different applications development.

Most participants understood perfectly well the core concept behind the personalisation sub-model. Figure 4.14 shows that only a single participant answered negatively to S1 ("The core concept "personalisation – parameter – resource" of the model is easy to understand."), and s/he justified the answer by saying that a graphic user interface would probably support better the understanding of the concept, making that connection more intuitive. Nonetheless, three of the five participants strongly agreed with S1, pushing up the overall result towards a positive perspective ($\overline{x}$ = 4.2; σ = 1.17). The result consolidated the notion that we had with regard to the well acceptance of the personalisation sub-model by developers.

The previous part of tests showed that participants strongly accepted the notion and integration of 'context' as a useful resource for additional information. This second part supports that conclusion, as Figure 4.14 illustrates with the results for S2 ($\overline{x}$ = 4.4; σ = 0.49). The developers involved in the study had the needed experience that allowed them to have the notion that context is of great importance for nowadays applications, agreeing with the context segmentation used by our solution. When asked about the integration of more external context services and types in CAPE, the results were not conclusive. On the first group of participants, we had a division in participants' opinions regarding Q2, with 50% who thought there should be more types of services integration, and 50% who thought the given types were enough. Unfortunately, on this second part of tests, the answer is still not clear with three in five responding "yes" to the question. Although there is a small tendency for a positive response, the number of participants is not enough to draw a strong conclusion, which is aligned with our expectations. Developers would need more time and experience implementing personalised Ubicomp applications in order to find out which context services they would need. It is our conviction that CAPE should integrate as many as possible.

The results obtained with S3 illustrates that participants found that applying personalisation to the application has a tendency to be easy to design. However, the results were not as positive as we would expect from developers used to work in the HCI field ($\overline{x}$ = 3.6; σ = 0.80). None of them disagreed with S3, but 60% had a neutral response, which can means that they needed more time to feel more confident with the usage of the solution. Moreover, for the evaluation they needed to come up with their own ideas for personalisations, having little time for the effect and feeling some pressure. In the first part, participants only needed to implement the personalisations that we suggested to them.

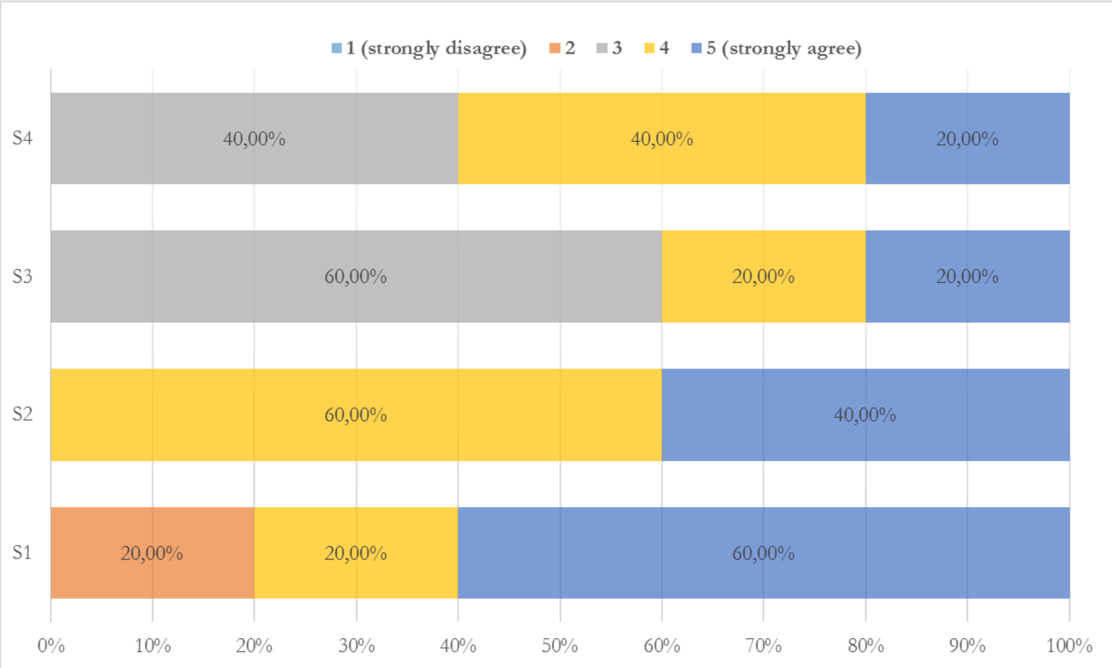


Figure 4.14 - Study D.1: Summary of part 2's results for S1-S4.

In the end, these developers manifested a real interest in using CAPE for future developments of personalised applications and systems ($\overline{x}$ = 3.8; σ = 0.75). Three participants clearly stated an interest in using our solution in the near future, with one of them being very interested. The result for S4 is fully aligned with the result for this statement in the first part with the other group of participants.

About the last four statements, Figure 4.15 shows a results scenario that is not much different from the part 1's scenario, but being a little bit better. The results are overall positive, being very good for statements S5 and S6 and a little negative for S8. Once again, it was confirmed that the personalisations instantiation with XUPIL is fairly easy to understand ($\overline{x}$ = 3.8; σ = 0.75). 60% of the participants thought it was easy, or much easy (one of them), to understand how to use XUPIL in order to define personalisations instantiations. 40% were neutral regarding this statement S5 and none disagreed, which is a good indication. We think these positive results are related to the experience these participants had regarding the use of XML-based languages.

The results for S6 decreased slightly ($\overline{x}$ = 3.6; σ = 0.80) since one participant that gave 4 to the previous statement had here a neutral position. We think this participant still thought the usage of XUPIL was not that simple even after understanding it. Summarising, 40% of the participants found the personalisations instantiation through XUPIL simple to accomplish and no one responded below 3, which is very promising.

Globally, the personalisations instantiation with XUPIL was not hard to learn ($\overline{x}$ = 3.6; σ = 1.02), according to the opinions of this group of participants in the study. Statement 7 shows a wider range of results than the previous ones, but only presents a value of 20% not agreeing that XUPIL is easy to learn. On the other hand, 40% consider the language, and its underlying model, easy to learn, and one participant even reported it as being very easy to learn.

Once again, we found out that these participants were also concerned with some lack of expressiveness in the process of the personalisations instantiation with XUPIL ($\bar{x}$ = 3.2; σ = 1.17). Although the results are similar to the results obtained in the previous part of tests, here we only have 40% of participants that disagreed with S8. No one strongly disagreed, which indicates these developers do not see some lack of expressiveness as being a so critical issue.

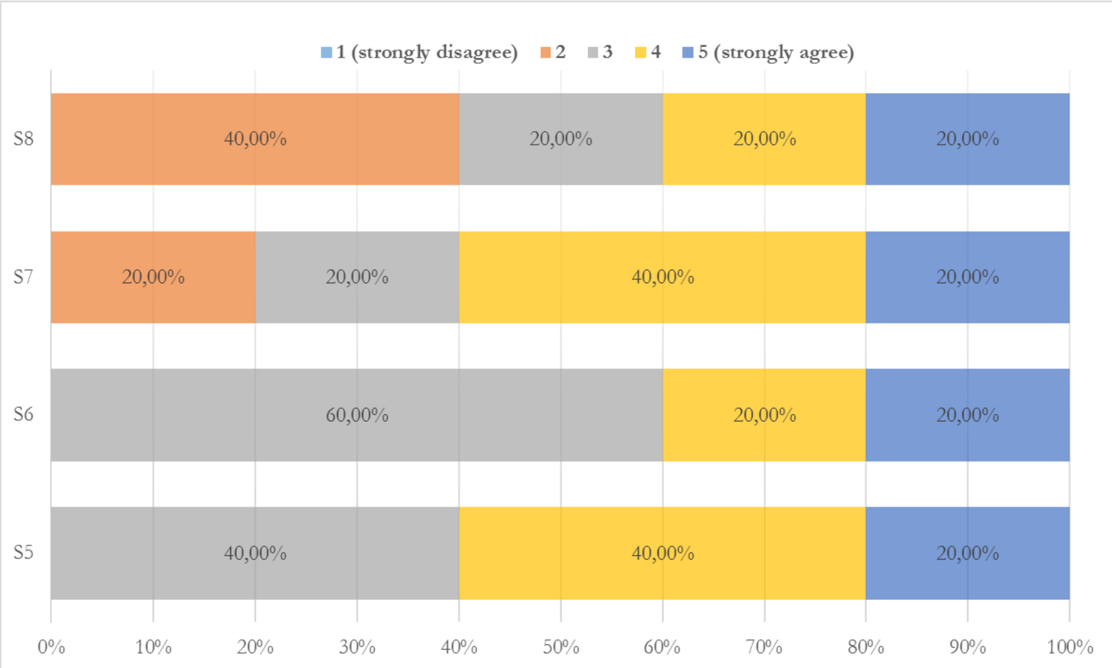


Figure 4.15 - Study D.1: Summary of part 2's results for S5-S8.

Regarding the comments, doubts and suggestions question, some participants expressed satisfaction regarding the usage of a language based on XML to configure their applications. It was also referred that CAPE needs to integrate more diversity regarding machine learning approaches in order to address different kinds of applications, for instance, focusing more on the recommendation of items.

4.1.4. Final Remarks

In section 4.1.2, we presented the definition of four personalisations for LEY, which is a context-aware pervasive application. This set of personalisations instantiations prove that the model can be used to personalise applications such as the mobile prototype of LEY. Moreover, it was possible to show how context can be used to segment important data, helping to fine-tune the user profiles. Therefore, they allowed us to test both the model and its language, proving their viability. Additionally, the personalised version of LEY allowed us to test and validate the CAPE's architecture and its user profiling module. These functional tests were important at the time because in the following study phase (see chapter 5) we would need to have a stable framework for interaction with end users.

Regarding the studies with developers, the results were very promising, showing that every developer was able to accomplish the task of personalising the requested application and most of them reported they would like to use our solution in the near future.

An interesting point taken from the tests is that the developers did not perceive the data normalisation issue. This situation happens when a parameter expression is structured in a way that results in data that is not balanced in the long term. For instance, if a parameter expression only measures the number of clicks in a certain menu, long time users will tend to have a larger value in comparison to newer users. The solution to this issue consists in creating an expression that represents the frequency of clicks per minute of measured login time. If necessary, we could go even further by dividing the previous expression by the number of days that have passed since the account was created. Maybe the participants were not aware they could test and play with different resources to reach that normalisation approach because they had not much time to think about how to obtain the parameters that would best represent the appropriate profiles. We suppose they would be aware of that possibility with time and experience of use. Study D.2 will support this hypothesis.

4.2. Study D.2 – Asking the Developers

Although appearing here as second in the sequence of the evaluation studies presentation, study D.2 was the last one to be conducted, which corresponded to the last task of this research work, not counting with the final writing of the dissertation. The plan had to be that one because a part of the study was designed to include the input from the developers involved with us in the development of the prototypes used in the remaining studies. Hence, studies described in the following chapter were conducted before this study D.2.

At this phase, our main focus was to obtain a last evaluation of the entire personalisation solution. Therefore, this study was focused on the evaluation of X-Users/XUPIL and CAPE, but it was also aimed at ascertaining if P²MUCA could provide the needed services responding to the developers' solicitations. As in the case of study D.1, part one of D.2 was directed to developers with different levels of experience. Once again, it was very important to include participants with little experience in the field.

One part of the study was based on the Walker[3] application, but the second part was based on the prototypes developed and used by our research team. Therefore, we involved in the study developers of those prototypes. The participation in the process of applying the personalisation model and using the platform in the implementation of those prototypes gave knowledge to developers that is important to collect and analyse. Moreover, this second part would be important to understand if developers that worked with us in the development of full-functional personalised prototypes, and during a longer period, would present significant differences in the evaluation of the solution compared to the first group. We hypothesised that it could happen, because using a solution like this one is not something simple to learn and master in the space of hours, or even days, for some lab tests, especially to use it comfortably and confidently.

4.2.1. Walker: People Counting in Indoor Spaces

At the base of the first part of the study was the Walker mobile app as a proof of concept prototype that would be personalised in the tests by the recruited participants. Among all the prototypes used in the studies, Walker has been the only one not developed by us in the context of this research work. Walker has been selected from a pool of prototypes implemented in a course at Escola Superior de Tecnologia de Setúbal of IPS (ESTSetúbal/IPS). The students of that course had to design and implement prototypes that had to integrate knowledge acquired in the subjects of IoT and mobile computing. The Walker prototype is an IoT-driven system that provides a mobile interface to track the number of people in indoor spaces, mainly directed to the ones responsible (owners, managers, or

[3] The concept design and the implementation of the non-personalised version of the Walker prototype was done by students Rúben Diogo and Diogo Banha, from "Curso Técnico Superior Profissional em Programação Web, Dispositivos e Aplicações Móveis" of ESTSetúbal/IPS.

supervisors) for the management of those spaces (Diogo and Banha, 2016). Although it has not been developed in the scope of our research, the Walker prototype, and the motivation behind it, will be briefly presented in the following subsections, following the approach taken for the remaining prototypes.

Introduction and motivation

Advances in ubiquitous sensing, communication, actuation and interaction technologies are allowing us to get closer to that smart world presented by Mark Weiser in (Weiser, 1991), in which computational intelligence is pervasively integrated within everyday physical environments. Continuous recognition of the presence of people, their identity, location, movement and activity patterns in real-time is a key challenge in the development of smart spaces towards personalised and adaptive services (Surie et al., 2013). Moreover, indoor tracking systems will be a foundation for the smart homes of the future, with applications such as: elderly and patient monitoring, activity recognition, and occupancy-driven lighting, heating, and cooling (Hnat et al., 2012).

The demand for automatic counting of people at event sites, buildings, or streets has been increasing due to several reasons (Fujisawa et al., 2013). Some surveillance applications require the detection and tracking of people to ensure security, safety, and site management (Liu et al., 2005). Crowd size estimation is an important task for both operational and security purposes (Ryan et al., 2015). The number of people, and even their distribution, in a public space can be used to gather business intelligence, such as, to ensure that normal operating conditions are maintained. A good example is counting customers who enter and exit from shops, which can be used for acquiring consumer shopping patterns and measuring shops' business performances. In outdoor spaces, for instance, it can be important to counting pedestrians at streets to effectively decide when to execute road construction and maintenance. The more traditional manual pedestrian counting presents some disadvantages since personnel expenses can be high and a simple task like counting people, but for a long time, can cause the workers' stress, introducing human errors (Fujisawa et al., 2013). As closed-circuit television (CCTV) becomes ubiquitous, it grows increasingly difficult for human operators to monitor all of the available data due to the sheer number of cameras installed (Ryan et al., 2015).

The solutions that are used to track and count people in indoor spaces can be roughly divided into three categories (Hnat et al., 2012): 1) tracking devices, 2) vision or audio systems, or 3) non-intrusive tracking. Among these, the most widely deployed methods to automatically count moving people use laser sensors (e.g., (Zhao and Shibasaki, 2005a; Zhao and Shibasaki, 2005b)) and infrared sensors (e.g., (Kazuhiko Hashimoto et al., 1997)). (Zhao and Shibasaki, 2005b) proposed a solution for counting people in real time using a network of single row laser range scanners. On the one hand, a sensor-based solution presents advantages regarding low cost and robustness against changes in the illumination conditions. On the other hand, a method like this one sometimes fails to count people correctly when they appear walking together and their heights are very similar, or if the heights do not fall within the presumed range (Fujisawa et al., 2013).

A solution using multiple infrared sensors can prove to be successful counting people moving in various directions, but the counting accuracy degrades considerably when an occlusion frequently occurs (Fujisawa et al., 2013), which can occur when many people are present. (K. Hashimoto et al., 1997) used an array of passive infrared (PIR) sensors arranged in a line to detect the number of people through a flight of stairs. In a previous work (Conrad and Johnsonbaugh, 1994), the centre scanlines of an image frame are used for a similar result. These examples of non-intrusive sensor-based approaches are used to detect people that pass through a confined area, not adapting very well to open spaces scenarios. Furthermore, any error at detection time will propagate indefinitely since each person is only detected at entrances and exits (Teixeira and Savvides, 2007). In a similar approach but with different technology, (Hnat et al., 2012) introduced Doorjamb as a non-intrusive room-level tracking system for homes. Doorjamb uses ultrasonic range finders mounted above each doorway, pointed downward to measure the distance to the person as s/he walks between rooms.

There are many works using video processing technologies as alternative methods for people tracking and counting, such as the cases of (Teixeira and Savvides, 2007), (Fujisawa et al., 2013) and (Yang et al., 2003). A different and interesting approach is the one from (Xu, 2013), which presented Crowd++, an unsupervised speaker counting technique through audio inference with smartphones to estimate the number of people in social hotspot places. A different way of counting people is through the tracking of their devices using wireless technologies, such as, WiFi and Bluetooth Low Energy (BLE). Sensors and device-based are often complementary solutions, where each technology has its challenges and benefits. BLE beacons[4] primary goal is to send push notifications (Location Marketing). Nowadays, there are different commercial solutions using these beacons, which are transmitters of BT low energy radio waves signals. The devices work indoors, which makes this technology ideal for communicating with customers. Beacons allow the customer's application to find if it is close (in proximity) to a specific location such as display or aisle. Therefore, they ended up being a favourite tech for location marketing services.

However, it is very hard to find reliable systems that also provide mobile interfaces that allow the responsible for the space to know how many people are over there, at any time and from anywhere. Moreover, we were not able to find any interesting solution providing personalised interfaces and services for this kind of systems. Therefore, we chose Walker for the studies based on this fact and on the previously acquired knowledge that people counting systems are important towards Ubicomp scenarios. Besides that, Walker was a stable and simple prototype that could be used easily.

System overview

The system behind the Walker concept is comprised of three main components (see Figure 4.16), just like in the case of LEY, which are the following (from left to right):

1. the sensing platform, which is mainly composed of two PIR sensors, a microcontroller and a WiFi module (esp8266),
2. the web server that supports MySQL and PHP,
3. the Walker mobile app, which is the interface with the end user.

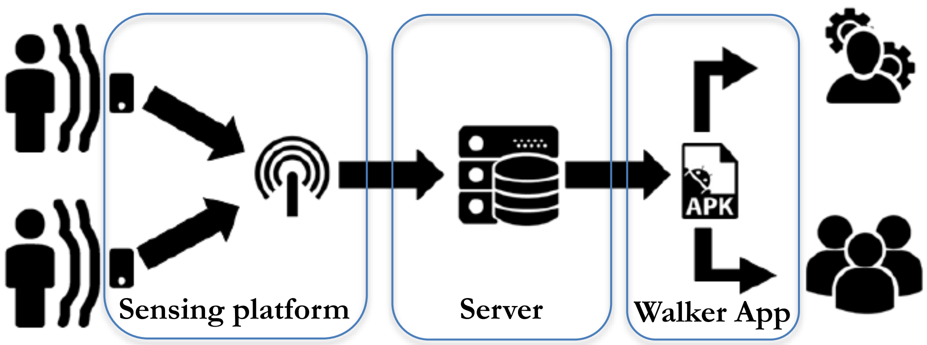


Figure 4.16 – Walker: the system architecture (Diogo and Banha, 2016).

The sensing platform provides the two PIR sensors, where one is responsible for tracking every entry and the other for every exit. Therefore, it is possible to know how many people have entered or left the monitored space/place in a given period. The difference between the two counters corresponds

[4] An example of BLE beacons is the solution from estimote: https://estimote.com/products/.

to the total number of people in the place, currently or in a given period. The sensors are connected to a microcontroller in which their programming is done. This subsystem is connected to the esp8266 module in order to have wireless communication to the web server. The web server provides the MySQL database and a PHP-based services module for connection with the client of the system, which is the mobile app that is implemented in Java for the Android platform.

Figure 4.17 shows the flowchart of how data flows in the Walker system. Every time one of the PIR sensors detects the motion of a person, data is processed in the microcontroller and the information regarding that event is showed in an on-site display that is connected to the system. The web server receives the sensor data in 10-second intervals and stores them in the database. This way, the central repository can be considered as being, as much as 10s allows it, up to date. The Walker mobile app requests the necessary data through the PHP-based services that query the database directly in the server. The app receives data structured as JSON objects, which are parsed conveniently to be stored in the local database of the app.

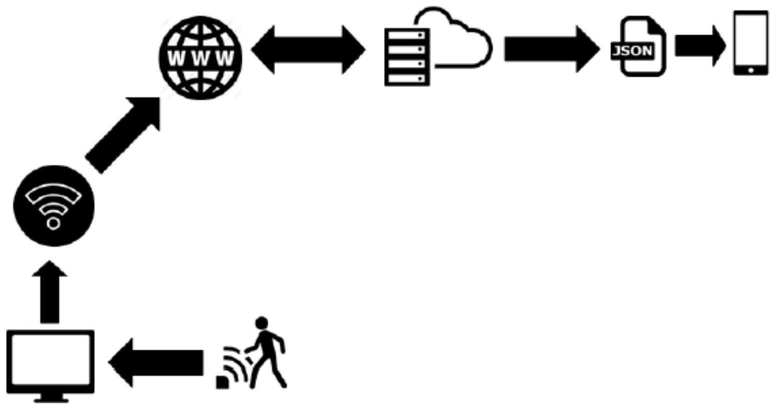


Figure 4.17 – Walker: how it works (Diogo and Banha, 2016).

Rationale and design

The Walker app is the mobile interface for the IoT system, which is a simple prototype that is limited to a few functionalities, for now. It is designed mainly for the owner or administrator/manager of places (indoor spaces with meaning). This user may be responsible for several places; thus, it is possible to add all the places to her/his list, provided the system is installed in those indoor spaces. However, "clients" of the place can also be users of the app. Therefore, the app was designed taking into account two types of users: administrator and client.

The app provides a login screen (Figure 4.18) since different users with several places can use it. The mobile app allows its users, regardless the type, to have access, at any time and from anywhere, to a screen where all their places are listed, showing how many people are in each one of those places (Figure 4.19). If the user is an administrator, then the places are the ones s/he managing. The administrator area gives access to analytics regarding the progression in the number of visitors throughout the day, week, among other data. Walker may help the administrator to: understand the behaviour of the space's users, which can be customers of a store; discover patterns in the visitors' behaviour (e.g., in context of weather); detect the space's peak times in a day, week and month; target and evaluate marketing in-space activities more precisely; compare results of different spaces; and even optimise the space's staff based on real needs.

On the other hand, knowing the number of people in a place may help a client to decide where to go next because, for her/him, some places may be more crowded than others. Nonetheless, a notification will be sent to the user if a space in her/his list is almost full. The threshold can be defined by the user, who will see the list of favourites places in her/his area (Figure 4.20), which s/he has bookmarked for a specific reason and according to the information that is provided for each place.

The administrator can add places and edit their information whenever s/he needs to do so. An important feature is that a space must be added having necessarily a location (address) and a city to which it belongs. The list of places of a user is initially shown by city, to which s/he belongs, and then by type of space. Thus, the current city of the user is stored. There are other filters that can be applied when the places are listed, such as, (not) crowded places, type of space, and brand to which it belongs. The app detects when the user is in the area of one of the places included in the system, since s/he may be inside the space, and then asks the user if s/he is really inside to "check-in" (feature similar to the one found in commercial apps such as Foursquare[5] and Swarm[6]). If that is the case, then locations (space and user) will be registered as matching and the user will be able to execute functionalities related to the space, such as posting photos. A search functionality lets the user searches for nearby places.

Figure 4.18 - Walker UI: login screen.

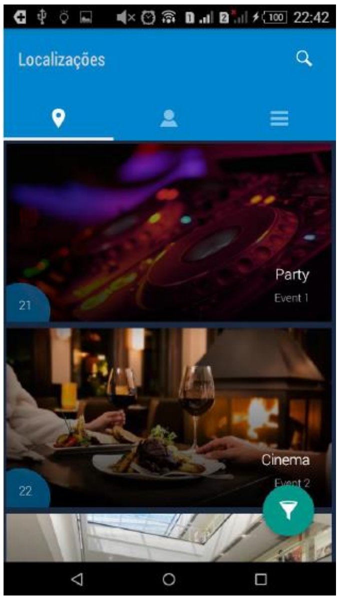


Figure 4.19 – Walker UI: locations/places screen.

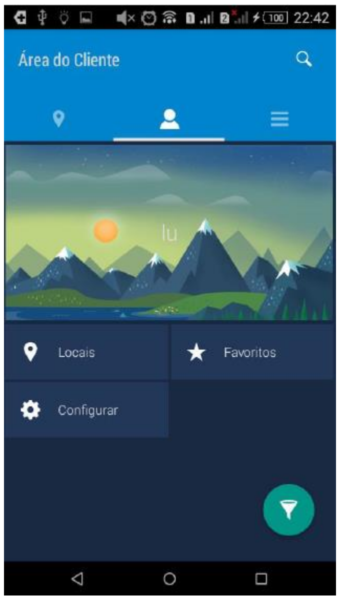


Figure 4.20 - Walker UI: client user area.

4.2.2. Personalisation Design for Walker

We were interested in using Walker as an independent prototype to conduct the evaluation on top of it. Therefore, at the base of the study was the definition of a small set of personalisation instances for the mobile app. It was not relevant to find out if these would work fine since we were not going to test the results of Walker's personalisation with end users, being only an experiment. Nonetheless, it was important that it would make sense to the participants to help them understand the personalisation process and its importance. After studying Walker, we found out a few possibilities for personalisation while looking into what the system could provide regarding interesting data to be treated as useful resources. The process was not easy since the mobile app was still a limited prototype, not having many functionalities that could be used for personalisation matters. Still, that limitation was positive too, since we did not want a complex prototype and the fact it was working was another important point. For this particular study, we decided to select the resources described in Table 4.7.

[5] Foursquare - https://foursquare.com/
[6] Swarm - https://www.swarmapp.com/

Once again, just like with LEY, a user needs to log in to be recognised by the system and to be able to use the app. Thus, two resources related to login can be used. The first one is the number of logins (`countLogins`), and the other is the time that the user has been logged in since s/he started using the app (`loginTime`). Users that constantly log in may present a different profile in comparison to users that log in occasionally. There may be users that are constantly logging in and logging out but not doing much with the app. Others may spend a lot of time with the app running, but not doing much too. Therefore, the two resources may work better in conjunction. Other data that can be used as a resource to combine with the previous ones is the number of usage days, i.e., the number of days that have passed since the user account was created (`numberUsageDays`). The latter can be used for data normalisation purposes in order to balance the other two resources in the long term. This way, we are able to create an expression that represents the average of both total login time and number of logins per day of usage. Moreover, another interesting resource is the number of active days (`numberActiveDays`), which is not to the number of sessions or logins. For this particular resource, each user is counted just one time, regardless of if s/he uses the app once per day or hundreds of times per day.

R	Resource	Description
Interaction stream resources		
1	**loginTime**	[sum] Total log time by the user.
2	**countLogins**	[increment] Number of logins made by the user.
3	**numberUsageDays**	[sum] Number of days since the user account was created.
4	**numberActiveDays**	[increment] Number of days the user was active.
5	**numberPostedPhotos**	[sum] Number of photos posted by the user.
6	**countSearches**	[increment] Number of searches made by the user.
7	**countBookmarkedPlaces**	[sum] Number of places bookmarked as favourites by the user.
8	**totalAverageCrowdedDays**	[replace] Sum of the average crowded days per week for all bookmarked places of the user.
Context resources		
9	**userLocationToSpace**	[value] user is inside or outside the space: in; out.
Preference resources		
10	**userType**	[demographics] the type of user: administrator; client.

Table 4.7 – Walker: resources data on which personalisations are based.

We selected more data related to interactions with the app, which are the number of photos posted by the user (`numberPostedPhotos`), the number of searches the user made (`countSearches`), and the number of places bookmarked as favourites by the user (`countBookmarkedPlaces`). All these resources combined may show how interactive is a user in the Walker's context. We could be even more rigorous storing all the interactions with the app's widgets, but that procedure was not necessary for this study. We would be making the process of personalisation for Walker unnecessarily complex and more laborious.

Furthermore, we wanted to include different types of resources. In order to do so, we thought of using the type of user to fine-tune profiles since it is a characteristic that naturally differentiates the personas that will use Walker. This resource is used as a preference (`userType`) since it can only have two possible values, which are related to the account type selected by the user when s/he created the account. Context is also used to indicate if a user is inside or outside a place (`userLocation-ToSpace`), with the values of 1 and 0, respectively. As such, the other interaction stream resources are stored separately for those values, besides the global values.

On the one hand, some resources were chosen because the participants in the study could easily gather them through programming, probably not having difficulties to understand them. On the other hand, others appeared after coming up with what to personalise to conduct the studies. The personalisation instances are described as follows (a possible XUPIL script defining Walkers personalisation is presented in appendix A.6).

P1) crowded places information

This first personalisation instance was designed with the intention of being simple and straightforward to help the study's participants entering into the computing concept they were facing for the first time. That was the primary intention with its definition. Making some sense was the second one.

Therefore, a simple parameter was created to define the personalisation of information related to the relation between bookmarked places and how crowded they can be. Expression (10) shows the definition of the parameter `crowded_places_level`, based on two resources: `countBookmarkedPlaces` and `totalAverageCrowdedDays`. It indicates the average of crowded days in a week per bookmarked place. The value of `totalAverageCrowdedDays` should be updated in P²MUCA at the end of each week. In order to better understand this value, let us suppose the user has two places in her/his list. The first one gets crowded, on average, two days per week. The second one never gets crowded. Thus, the sum is equal to two, which is the value that should be updated on our platform. Ideally, the app should replace the value on our platform once a week (at the beginning or end).

$$\left(\frac{totalAverageCrowdedDays}{countBookmarkedPlaces}\right) \tag{10}$$

In order to keep this personalisation instance simple, we considered only two options (two clusters) corresponding to two personas (see Table 4.8), regarding this particular behaviour. Thus, if the user gets a high value it may reflect the user tends for places that get crowded easily, maybe because s/he likes and bookmarks places that are popular. There can be other reasons. Information and type of notifications can be adapted according to this profile, also depending on the type of place. Administrators will analyse what to adapt in terms of information according to the places they are managing.

crowded_places_level	*Personalisation option*
high	crowdedTendency
low	nonCrowdedTendency

Table 4.8 - Walker: two personalisation options for the "crowded places information".

P2) notifications level

This personalisation combines the preference resource `userType` with the interaction resources that can denote how interested the user is in using Walker. Users can receive regular notifications that can be adapted according to the user profile. If the user denotes a higher interest for the app, then s/he will receive fewer notifications and with specific content about the favourite (if a client) or supervised (if an administrator) places. In the opposite case, the user will receive a much higher number of notifications with content that sometimes will be more generic, addressing what the user can gain with the app usage, and other times will be directed to questions that are more specific. This is only an example of an approach for the notifications in order to test the required personalisation for the study. Others approaches could be tested and followed.

This personalisation offers three basic scenarios, i.e., three personalisation options (see Table 4.9), resulting from three personas representing three different levels of interest for the app: 1) `low`; 2) `average;` 3) `high`. They are determined using the parameter `interest_level` that uses the expression (11) in its definition.

$$0.4 * \left(\frac{loginTime}{numberUsageDays}\right) + 0.4 * \left(\frac{countLogins}{numberUsageDays}\right) * fctr + 0.2 * userType * fctr \quad (11)$$

The user type value enters with a lower weight in comparison to the resources that denote interest by the user. This decision was taken to strengthen the value of the administrator user in the result. Even if the administrator uses the app less, it is assumed that s/he has a natural interest in using it enough to manage her/his places. The `userType` was defined with a value of 1 to the client and 2 to the administrator. The averages regarding the login time and logins count per day are treated here in an equivalent way, having the same weight in the calculation. However, the average `countLogins` and `userType` should be multiplied by a factor `fctr` (studied according to the average values of `loginTime`) to obtain values in the same order of magnitude as `loginTime`. Therefore, this expression is only a first approach that would need to be tested and tuned throughout a long-term evaluation, not only to find that factor, but also to find the right combination between the resources.

interest_level	*Personalisation option*
high	interestedUser
average	regularUser
low	unconcernedUser

Table 4.9 - Walker: three personalisation options for the "notifications level".

Moreover, since this personalisation is mostly based on the user behaviour regarding interest for using the app, context segmentation was also used to take into account where that user behaviour was happening, if it was inside or outside places registered in the app. The context resource named `userLocationToSpace` was used here. As such, the interaction stream resources are stored separately for the two values, in addition to the global values being stored. The resources values indexed to the context values have a weight of 50% in the calculation of the parameter values. This way, the personalisation option returned when the user is inside a place, no matter which one, may be different from the one returned when s/he is outside the places. For instance, a person may use the app a lot when s/he is inside a place, but almost never when s/he is not in any of the places.

P3) badges are bonus

The personalisation of Walker brought the possibility of new features. One of them is related to the interactive profile of the users, based on the aforementioned interaction resources. It was intended to reward users with badges that could mean receiving bonuses in certain places. These bonuses can be vouchers with discounts, offers, etc., depending on each place that adheres to the initiative. In addition, to make this personalisation instance more interesting and somewhat more complex than P1, it was decided to combine the interactive profile with the profile relative to the level of interest of a user.

The new parameter is called `interactive_profile`, which is based on the resources (expression (12)): `numberActiveDays`, `numberPostedPhotos`, `countSearches`, and `countBookmarkedPlaces`. The parameter is defined with the two options (two clusters) that represent personas in terms of interactive profile: 1) `high`; 2) `low`.

$$\left(\frac{0.5 * numberPostedPhotos + 0.2 * countSearches + 0.3 * countBookmarkedPlaces}{numberActiveDays}\right) \quad (12)$$

According to Table 4.10, the personalisation option returned for this instance P3 would be one of four: `no badges`, `badges`, `badgesBonus`, or `bonus plus`. It is clear that the first one implies the user will not have any badge, nor bonus, and the last one means s/he will receive an extra bonus, since s/he presents both much interest and high interactive behaviour. In the middle, the second one means s/he is on the right path, receiving badges to prove it, and the third option means that with badges come "real awards".

<table>
<tr><th colspan="2" rowspan="2"></th><th colspan="2">interactive_profile</th></tr>
<tr><th>low</th><th>high</th></tr>
<tr><th rowspan="3">interest_level</th><th>low</th><td>no badges</td><td>badgesBonus</td></tr>
<tr><th>average</th><td>badges</td><td rowspan="2">bonus plus</td></tr>
<tr><th>high</th><td>badgesBonus</td></tr>
</table>

Table 4.10 - Walker: four personalisation options for the "badges are bonus".

4.2.3. User Study with Developers

This second assessment moment with developers was focused on evaluating both XUPIL as the DSL for X-Users and P²MUCA as the overall solution for personalising applications and systems.

Design and participants

As previously mentioned, the tests were organised in two parts, based on:

1. The Walker application – Recruited participants had to implement the personalisation of Walker according to the design presented in the previous section. As the tests were supposed to be executed by application developers, it was assured that every participant had programming skills, particularly in Java language, regardless the years of experience in the field.
2. Other studies' prototypes - Developers had the experience of working with us in the prototypes' personalisation used by our research team in the studies. In further discussions, these second part's participants will also be referred under the designation of experts, because of the longer and more solid experience with P²MUCA in comparison to the remaining.

Regarding the first part of the study, we needed a sample of participants with significant experience in development of applications or, at least, a good part of them would present several years of experience. To accomplish that requirement, we invited students and professors from the informatics area, which were recruited at FCT/UNL and, mainly, ESTSetúbal/IPS, and three informatics engineers from startup companies. In total, we worked with 21 volunteers (4 female; 17 male), with ages ranging from 19 to 49 years old ($\bar{x}$ = 29.1; σ = 9.46), and development experience ranging from one to 30 years ($\bar{x}$ = 10.6; σ = 8.23). Regarding education, 38% of participants were bachelor's students, 24% were informatics engineers and 38% had post-graduate degrees.

In the second part of the study, the tests involved six developers (all male) aged between 24 and 47 years ($\bar{x}$ = 30.3; σ = 7.74), which results in years of experience ranging from 7 to 29 ($\bar{x}$ = 12.5; σ = 7.54). They were involved in the development of the prototypes according to the following distribution: one with LEY; one with FCT4U; one with WeSync/WePlayTennis; and two with PhysioMate/jPk. The FCT4U's developer has been involved in the implementation of P²MUCA, but not in the concept, principles and model behind it. S/he implemented and deployed the cloud-based platform under our guidance and supervision, integrating CAPE and deploying the personalisation API, which gave her/him a deeper and more critical knowledge. We recruited another developer who was involved with us in the development of the project BroiStu, which is focused on delivering a mobile health solution to cope with people who stutter (PWS) (Madeira et al., 2013; Demarin et al., 2015).

The evaluation process of this study's part one followed the approach and steps applied to study D.1 and already detailed in the corresponding section 4.1.3. At this phase, we only present the differences in the design of the process and describe particularities of the current study. The tests took place in a lab at ESTSetubal/IPS, with a researcher (a human monitor) observing and taking notes, so that data, such as, completion times and number of syntactic and semantic errors, could be collected. First, we briefed each participant, taking the chance to explain the concept behind X-Users and teach how to instantiate it using XUPIL for the effect. We also trained the participants using some examples. Furthermore, we described the P²MUCA's UI for registering applications and explained the P²MUCA's API for programming personalisation requests. We provided participants with a tutorial explaining and exemplifying the API's use to help them in the task of programming the requests to P²MUCA in the Walker app (Figure 4.22). We prepared the Android project of Walker to include a preprogrammed class with semi-implemented methods to deal with the personalisation calls to the platform and provided participants with a document describing the required personalisations (presented in the previous section).

Afterwards, the participants were asked to perform individually the following main tasks:

1. Conceive and design the three personalisation instances for the Walker prototype, one by one, using XUPIL in a single document (we kept track of the required time and number of errors). To accomplish it, the necessary resources and parameters would be defined, unless they already existed due to a previous definition, being reused. Whenever justified, the human monitor would guide and respond to doubts (initially, we gave them five to ten minutes to conceive and structure the initial ideas).
2. Register the app in the platform's UI and submit the XUPIL document with the personalisation definition (before that, the human monitor would clean up the remaining errors, if needed). Each participant had to name the document's file according to her/his number of participation in the study. For instance, participant 10 would be *walker_p10.xupil*.
3. Finish the Android project's class, programming the methods for updating resources and requesting the personalisation option for each instance.

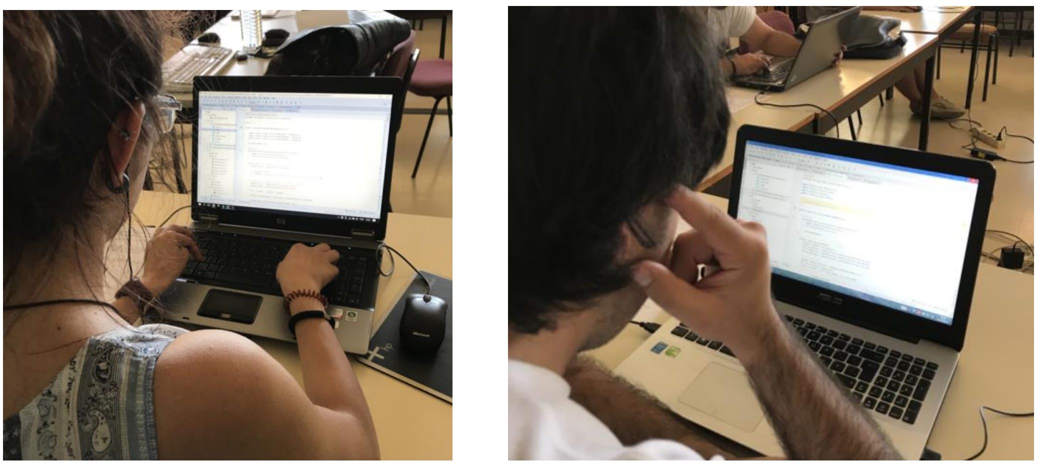


Figure 4.21 – Study D.2: One participant in the training session (left) and another in the testing session (right), with the tutorial to the left.

Finally, after having everything tested to their satisfaction, we asked the participants to fill in anonymously a questionnaire (Table 4.11) that was an updated version of the one used in study D.1. Personal interviews were also used informally while observing the testing session. The questionnaire started with a group of questions related to the participants' personal data to characterise them, including age, genre and years of experience. Next, from the questionnaire used in study D.1, we maintained the two open questions addressing knowledge about other tools for personalisation and ideas for types of external context services. We also kept the Likert-type scale statements, but this time ranged from strongly disagree (1) to strongly agree (7), focused on evaluating the important core aspects: S1) the personalisation principle's concept, S2) the integration of context, S3) the ease of use of personalising an application, in general terms, and S4) the interest in using our solution in the future.

Open Questions
Q1. Have you ever used any third-party approaches or tools to integrate personalisation into applications?
Q1b. In the positive case, which one(s)?
Q2. Do you think there should be more types of external context services?
Statements [7-point Likert-type scale, from strongly disagree (1) to strongly agree (7)]
The model-driven solution
S1. The core concept "personalisation – parameter – resource" of the model is easy to understand.
S2. The integration of 'context' for personalisation purposes is useful.
S3. Applying personalisation to the application is easy to design.
S4. I would be interested in using P²MUCA to provide personalisation for future applications.
Semantic Differential Scales
How do you evaluate XUPIL?
Pair1. Inexpressive - 1 2 3 4 5 6 7 – Expressive
Pair2. Complex - 1 2 3 4 5 6 7 - Simple
Pair3. Hard to Learn - 1 2 3 4 5 6 7 – Easy to Learn
Pair4. Hard to Read - 1 2 3 4 5 6 7 – Easy to Read
Pair5. Hard to Write - 1 2 3 4 5 6 7 – Easy to Write
Pair6. Hard to Understand - 1 2 3 4 5 6 7 – Easy to Understand
How do you evaluate personalising an application using the solution provided as P²MUCA?
Pair1. Uninteresting - 1 2 3 4 5 6 7 - Interesting
Pair2. Complex - 1 2 3 4 5 6 7 - Simple
Pair3. Inappropriate - 1 2 3 4 5 6 7 - Appropriate
Pair4. Hard to Program - 1 2 3 4 5 6 7 – Easy to Program
Pair5. Hard to Understand - 1 2 3 4 5 6 7 – Easy to Understand

Table 4.11 – Study D.2: Questions, statements and SDS in the questionnaire.

As presented in Table 4.11, the questionnaire had two final questions. These questions were answered by means of two multidimensional seven-point Semantic Differential Scales (SDS) (Osgood et al., 1975) featuring six and five pairs of bipolar adjectives, respectively, which have been derived from Sebesta's criteria for language evaluation (see section 3.5). We changed the study D.1's Likert-type scale statements regarding the use of XUPIL by the first SDS-based question as we considered it would be an improvement in obtaining a semantically richer response from participants, also according to other research works that had success following this approach. Finally, the last question was a new one since in this second developers-driven study we also tested the platform and its underlying modelling framework.

Results and discussion of part one

As previously stated, participants had to perform an experiment comprised of three main tasks. We gave support to the tasks since they were using a new approach for the first time, helping every time a participant was stuck at some points, mainly at the beginning. For the first task, the human monitor tracked and recorded the number of syntactic and semantic errors and the time it took each participant to complete the definition of a personalisation instance (described in section 4.2.2), from their options to the parameters, through the necessary resources. These results are in Table 4.12. We did not record identical data for the remaining tasks since the second one was relatively short and directed only to making them use the platform's UI, whilst the third one was somewhat dependent on their skills with Java language and a few of them had some limitations. Thus, we only wanted participants to use and assess the personalisation API in the context of a real application, to make them see how it works and what it provides with each call. It also worked as a proof of P²MUCA's performance, demonstrating that it was working and already responding to the submitted file. It was observed that all participants successfully implemented the given tasks. Naturally, some participants did not need any help and others needed it both to correct some errors and to advance at some points of the experiment.

Personalisation	Syntactic Errors			Semantic Errors			Time (seconds)		
	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max
P1	0.92	0	5	0.95	1	6	261	182	527
P2	0.78	0	7	1.02	1	8	316	225	643
P3	0.59	0	3	0.63	0	5	204	155	489

Table 4.12 - Study D.2: Results of experiment's task one (writing three personalisation instances).

Regarding the first task, Notepad++[7] was the editor used for writing the personalisations' definitions. All participants finished the XUPIL file with success, but the result varied in terms of readability, which was very interesting and expectable too. The result also varied in terms of time, needed to conclude each personalisation instance, and number of errors, as can be seen from the data presented in Table 4.12. The recorded data may help us to find up if XUPIL is a writable language, as well as being readable, which is implied. First, it is important to note that the complexity of P1 was the lower between the three, to work like a "warm-up" exercise. Participants only had to define two personalisation options, one parameter with two options and based on a simple expression, which used only two interaction resources. Then, in the second one, they had to come up with three personalisation options, create a new parameter, this time with three options, using four resources and having a more complex expression in semantic terms. Moreover, one of the resources was a preference and they had to cope with the integration of context, which increased the complexity of P2 in relation to P1. Lastly, P3 had an additional element since they had to write another parameter with two options and combine it with the one defined for P2, so reusing the latter. They had to come up with the definition of four personalisation options based on the combination of the options of the two parameters, which would result in six possibilities.

The three requested personalisations have systematically registered a lower mean number of both syntactic and semantic errors. Moreover, less than one syntactic error per participant, on average, is a positive number that is according to our expectations. It may indicate that XUPIL is easy to understand and learn. On the one hand, the language has a short vocabulary, and, on the other hand, the

[7] https://notepad-plus-plus.org/

majority of participants had several years of experience dealing with XML-based languages. The higher number of semantic errors per participant, on average, indicates that, here and there, participants had more difficulties in understanding and expressing well specific elements of the personalisation and defining the correct sentences (e.g., erroneous parameters' expressions, wrong order for the parameter options). We still consider the average number of semantic errors as being low and very positive. However, being higher than the average number of syntactic errors, and summing both, may indicate that XUPIL is harder to write than to understand and read. The correlation of results may indicate that globally XUPIL may be seen as not being so simple in the beginning, due to its underlying concept and model, which presents some complexity. However, the number of errors decreased from P1 to P3, becoming much less frequent, or non-existent, depending on the participant, proof of a learning curve for understanding XUPIL. Another positive indicator is the time taken by participants to produce the personalisation instances, both semantically and syntactically valid, even if with some help, which shows that XUPIL is simple enough to be mastered quickly.

These conclusions are further supported by the fact that, regardless their programming experience, this was the first contact these developers ever had with XUPIL and its underlying model. Designing the personalisation of the application, thinking about each instance and its options, the parameters and needed resources is really time-consuming and has some complexity, but afterwards, having all structured and prepared, the writing of the XUPIL document can be done easily and fairly quickly. Additionally, participants could, and should, use already defined parameters and resources in a new personalisation definition, reuse and adapt parts of code already written, and use the usual programming strategies, such as, copy/paste. It was definitely interesting to see how each participant wrote XUPIL code using their own style and indentations, some applying "CamelCase" notation for the names of the personalisation options, for instance, while others adapted and applied other conventions.

The questionnaire's results are described as follows. Once again, similarly to study D.1, all participants indicated that they have never used any third-party solution (response to Q1), specific tools or platforms, to support them in the process of applying personalisation to any application or project in whose development they have been involved. Participants have no way to compare our solution with other third-party personalisation tools and platforms. Three of them, all presenting more than ten years of experience in programming, post-graduate studies and habits of research, implemented application-specific aspects of personalisation in isolated projects, nothing transverse and easily reusable, using their own-programmed user modelling techniques. It was nothing generic, global or standard that they could even use across different applications. Therefore, for almost all the participants, the current experiment was the first time they had to think about programming personalisation into an application.

As for Q2, 17 participants would like to have more types of external context services and we also obtained five suggestions. Nonetheless, there are interesting suggestions. Three undergraduate students suggested the integration of platforms of IoT services where users may have data related to the objects and devices of the physical world with which they interact on a daily basis. Another participant added a more specific subject, which was a service that provides data about the connected fabric used in sports garments. We can put it in the same category of the previous suggestion, but, in this case, directed for a more personal context. Indeed, there are already platforms that manage those particular data and provide the services that P^{2}MUCA can integrate to be easily used as context, or even as demographics, in the personalisation of an application. Lastly, another idea was the integration of services that provide access to today's live marine and sailing weather forecast. It would be a specific case within the external weather services, which are already integrated.

In line with study D.1, the overall result of the four Likert-type scale statements is indeed positive as can be seen in Figure 4.22. As a very important point, it seems the majority of participants understood well the core concept behind the personalisation sub-model since 18 developers responded positively to S1 ("The core concept "personalisation – parameter – resource" of the model is easy to understand."). From the remaining three, only one participant answered negatively with three points (high negative). The result is very positive because, from our experience, it is not easy to make a new and somewhat complex concept understandable for others quickly. Naturally, the fact that the developers participated in a complete experiment, from designing the Walker personalisation, to using XUPIL and finally programming the requests to P²MUCA, allowed them to have a better understanding of the concept. Four of them even strongly agreed with S1, pushing up the overall result towards a clear positive result ($\bar{x}$ = 5.52; σ = 1.05). The result consolidated our opinion that the core concept behind the personalisation sub-model is well accepted by developers, even if a few of them have some doubts.

The first study with developers showed us that the integration of 'context' is really considered as being useful for personalisation purposes. The same happened in this second study since the results for S2 ($\bar{x}$ = 6.24; σ = 0.68) support that conclusion. The developers involved in the study also agreed with the context segmentation notion used by our solution. Moreover, S2 presents the best results among the four statements with an average above six points. On the opposite side, the results obtained with S3 are the worse ($\bar{x}$ = 5.04; σ = 1.09), illustrating that applying personalisation to one application is not so easy to design. This means the process can be hard in some parts, as we will observe with the results of the SDS-based questions. It is noteworthy that the two participants that disagreed a little (3 points) and three of the four that rated S3 with 4 points are undergraduate students with little experience. Only one participant with more than ten years of experience did not agree with S3, rating it with 4 points too.

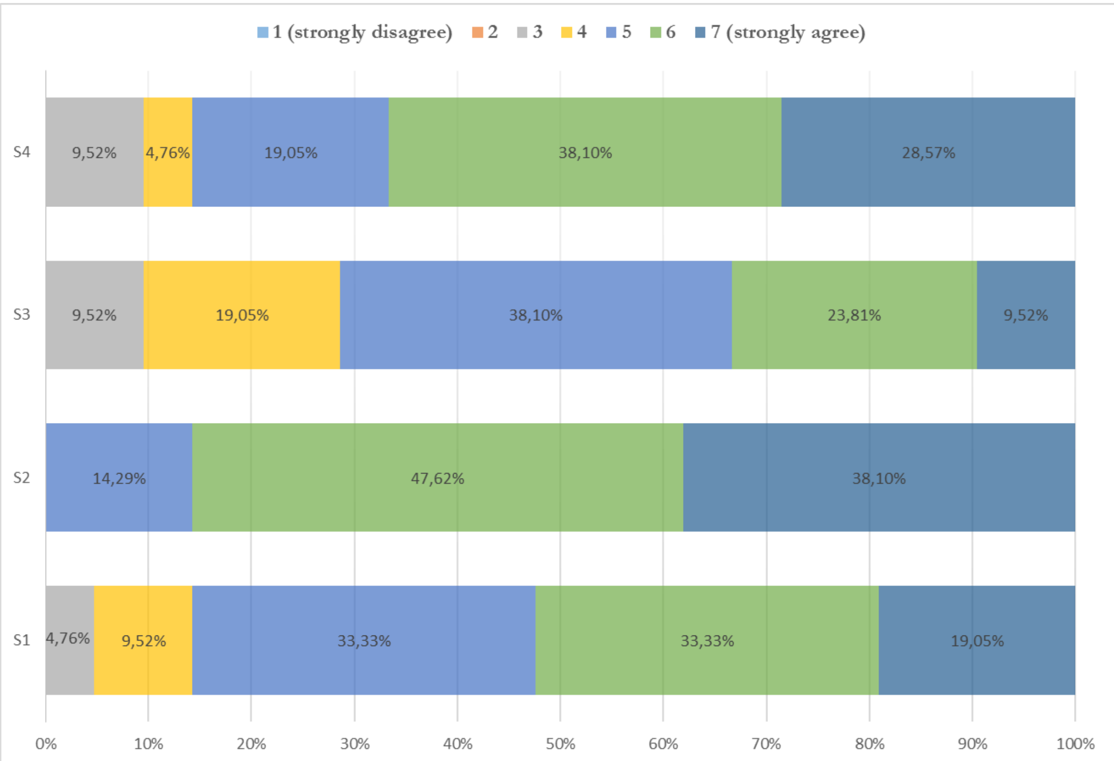


Figure 4.22 - Study D.2: Summary of part 1's results for Likert-type scale statements S1-S4.

The overall result for S4 is fully aligned with the result of this statement in the first study. The large majority of the developers involved in the study manifested a real interest in using our solution in the future ($\overline{x}$ = 3.8; σ = 0.75). Two participants disagreed with the statement, making it clear that they were not particularly interested in using it after the experiment. They have been consistent since they were the ones giving three points to the previous statement. Again, it is important to state that these were two of the younger participants, with one and two years of programming experience, which may indicate they still do not have the needed maturity to deal with the topics in evaluation. Another one did not disagree nor agree.

The results obtained through the SDS-based question "How do you evaluate XUPIL?" are very encouraging too (see Figure 4.23), revealing that the majority of our participants found XUPIL to be fairly easy to understand and read, besides being expressive. Globally, we can also extract that although not being considered very simple, the language is somewhat easy to learn. In spite the overall positive results, participants were left with the perception that the language is not so easy to write, what makes it the property with less positive ratings.

Inexpressive	0 0,00%	0 0,00%	1 4,76%	3 14,29%	5 23,81%	8 38,10%	4 19,05%	**Expressive**
Complex	0 0,00%	1 4,76%	1 4,76%	3 14,29%	9 42,86%	5 23,81%	2 9,52%	**Simple**
Hard to Learn	0 0,00%	1 4,76%	1 4,76%	4 19,05%	5 23,81%	8 38,10%	2 9,52%	**Easy to Learn**
Hard to Read	0 0,00%	0 0,00%	2 9,52%	1 4,76%	5 23,81%	9 42,86%	4 19,05%	**Easy to Read**
Hard to Write	0 0,00%	2 9,52%	2 9,52%	4 19,05%	8 38,10%	3 14,29%	2 9,52%	**Easy to Write**
Hard to Understand	0 0,00%	0 0,00%	1 4,76%	2 9,52%	5 23,81%	7 33,33%	6 28,57%	**Easy to Understand**

Figure 4.23 - Study D.2: Summary of part 1's results for the SDS-based question "How do you evaluate XUPIL?".

The participants with little programming experience who disagreed with statements S1 and S3 assigned 2 and 3 points to the pair 'Complex|Simple', which is really an indicator of the difficulties they had when designing a personalisation and in using XUPIL. It is interesting to observe that the overall results ($\overline{x}$ = 5.0; σ = 1.17) for this property are in line with the results of S3, which may indicate that the notion participants had regarding the simplicity of XUPIL had contributed strongly to the evaluation of that statement S3. Only 5 participants did not rated the pair with a score above 4 points. We wanted to design a language that would be as simple as possible taking into account the inherent difficulty in expressing the concept of a generic model for personalisation which has in itself some natural complexity. The fact that participants considered the language expressive ($\overline{x}$ = 5.5; σ = 1.10) illustrates they understood that the language has the necessary balance in the pair complexity/simplicity to reach the appropriate level of expressiveness. Moreover, generally, all participants considered that XUPIL is easier to understand ($\overline{x}$ = 5.7; σ = 1.12) than it is simple.

As for the question "How do you evaluate personalising an application using P²MUCA?", Figure 4.24 shows the results are even better than the ones obtained with the previous question. The developers involved in the study felt that the whole concept and how it was provided through the platform and its tools was really interesting. We really observed that attitude while conducting the tests. Participants became more enthusiastic as they progressed in learning the various aspects of the solution provided by P²MUCA. Participants considered the whole process of personalising an application through our solution as being relatively simple, but not many of them considered it as being very simple. Only one thought the whole process was really very simple. Nonetheless, only two rated the pair "Complex|Simple" with values below 4 points, which is positive.

Uninteresting	**0** 0,00%	**0** 0,00%	**0** 0,00%	**3** 14,29%	**4** 19,05%	**8** 38,10%	**6** 28,57%	**Interesting**
Complex	**0** 0,00%	**1** 4,76%	**1** 4,76%	**3** 14,29%	**10** 47,62%	**5** 23,81%	**1** 4,76%	**Simple**
Inappropriate	**0** 0,00%	**0** 0,00%	**1** 4,76%	**1** 4,76%	**5** 23,81%	**7** 33,33%	**7** 33,33%	**Appropriate**
Hard to Program	**0** 0,00%	**0** 0,00%	**1** 4,76%	**2** 9,52%	**5** 23,81%	**7** 33,33%	**6** 28,57%	**Easy to Program**
Hard to Understand	**0** 0,00%	**0** 0,00%	**0** 0,00%	**3** 14,29%	**8** 38,10%	**6** 28,57%	**4** 19,05%	**Easy to Understand**

Figure 4.24 - Study D.2: Summary of part 1's results for the SDS-based question "How do you evaluate personalising an application using P^2MUCA?".

The results are very positive when we focus on the pair related to be appropriate, which is in line with the fact they considered it interesting and with their high interest in using the solution in the future. Several participants related that appropriateness to the use of XUPIL and its expressiveness towards representing correctly the model for personalisation. It is noteworthy that the participants considered easy to program the personalisation of an application. Several weighted in this property both the utilisation of XUPIL and the programming of the requests to P^2MUCA using the HTTP-based API. Others only considered here at this point the API and its use inside the Java class they had to finalise. Indeed, we observed some ease in the programming of the requests. The first clash was hard, but after a few minutes, and a little help from us, participants quickly learned how to use the API effectively. Therefore, it was easy to program by the majority of them. Finally, we suppose all the aforementioned points counted to the very positive results obtained with the property of being easy to understand. We may conclude that these participants understood well (some more quickly and easily than others) what was involved in the whole process of personalising an application using P^2MUCA.

Results and discussion of part two

The six participants in this second part of the study were considered as being the experts in comparison to any other that only participated in a lab experiment. The involvement in the personalisation of a prototype over several weeks gave the corresponding developer solid and deeper expertise about the X-Users concept and the usage of its DSL, besides being comfortable and confident with programming the requests to P^2MUCA. They had the necessary time to learn calmly and master, as much as possible accordingly to the project and prototype, the entire solution in all its aspects and components. Moreover, knowing very well the application they were about to personalise would be an important advantage since it was expectable they would not have any shortcoming related to the application. For instance, on the other hand, part one's participants had the additional effort of having to enter into a concept and application they did not implement, which was an extra variable that may have conditioned some participants, especially those with less experience. Naturally, we applied the same questionnaire (presented in Table 4.11) to part two's participants only after they have been involved in using the solution provided by P^2MUCA. The questionnaire's results are described as follows.

In response to Q1, these six developers also indicated that they have never used any third-party tool or platform focused on the personalisation of applications. They all referred they were unaware of the existence of such integrated solutions to guide and support in applying personalisation. However, one of them has integrated social networks and ad networks into applications and websites, which ended up reflecting the users' preferences. Therefore, five in six participants had with us their first experience thinking about and programming personalisation aspects applied to an application. As for Q2, five participants stated they would like to have more types of external context services. One of them suggested the integration of traffic context services and any other service that provides data about the physical world in which the users live. Other recommended the integration of services related to nature, sea and weather conditions for outdoor sports practitioners that use several mobile applications. Three of them thought that the best approach would be to learn from what each developer needs for her/his project and encapsulate the most common needs as external context services.

The overall result of the four Likert-type scale statements is once again positive as can be observed in the chart presented in Figure 4.25. We opted to visualise the responses using a radar (spider) chart as a way of comparing the experts' scores in the form of a two-dimensional chart. There are only six participants and it was important to observe quickly the level of agreement of each one of them to the statements, which can also be quickly compared.

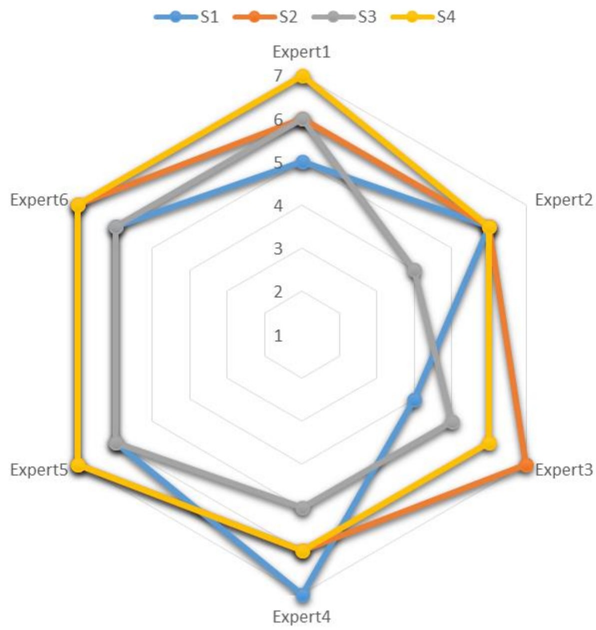


Figure 4.25 - Study D.2: Summary of part 2's results for Likert-type scale statements S1-S4.

For instance, statement S1 got results that we believe are very positive and interesting ($\overline{x}$ = 5.7; σ = 0.94), since they came from the participants who had a deeper critical perspective, with the necessary time to absorb the core concept "personalisation – parameter – resource" of the model. It is true that only one of the experts strongly agreed with S1, however we only had one with a neutral position. Three ratings of 6 and one rating of 5 are really very encouraging to us. We observed that Expert3 (WeSync prototype) had some initial difficulties understanding the concept when s/he was faced with the model for the first time. One reason for this may be related to the fact that s/he has had less time, in relation to others, to learn the concept before s/he has begun to use it. The team had to implement the WeSync personalisation in a very short time to respond to a tight schedule of experiments. Another reason may be the fact that thinking about and designing personalisation for WeSync was different from the rest as we can see in section 5.2, needing to put an extra effort in understanding well the model (this expert also rated S3 with "only" 4 points). On the other hand, Expert4 worked on the personalisation of "just Physio kidding" for several weeks, having had different sessions with end users which helped her/him to absorb calmly all the core characteristics of the model. These two cases demonstrate that some particularities of each prototype development may influence not only this, but all the results, which was expectable and of our interest. Prototype development-dependent analysis enables us to draw richer conclusions that complement the experiments-based studies.

The responses to statements S2 and S4 are indeed very positive as we can observe experts agreed or strongly agreed with both statements. Half of them rated the statements with a 7 and the other half with 6 ($\bar{x}$ = 6.5; σ = 0.50). These are strong indicators of how they look into the way context is integrated in the model and the opportunity of using our solution in future personalisations of new projects.

Finally, it is noteworthy that the results obtained with S3 are once again the worse from the four statements ($\bar{x}$ = 5.3; σ = 0.75). They are still positive, indeed, as none of the experts disagreed with the statement "Applying personalisation to the application is easy to design.", but one of them has a neutral position and there is not a single strongly agreement. However, three of them rated it with 6 points. We suppose the experience that each one had with the corresponding prototype development and its particular personalisation implementation process was decisive to the result of this statement. For instance, Expert2 had to cope with several details regarding the implementation of a personalised version of FCT4U, which design process had been somewhat tough at some points. Furthermore, Expert3 rated S3 with 4 points since thinking about the personalisation design for WeSync had to be done differently from the rest, as mentioned above, and this developer had some initial difficulties in understanding it according to the model. Afterwards, s/he mastered the language with very positive results.

Regarding the results of question "How do you evaluate XUPIL?", Figure 4.26 is self-explanatory, illustrating the global experts' subjective perception of and affective reactions to the language, based on the six pairs of properties.

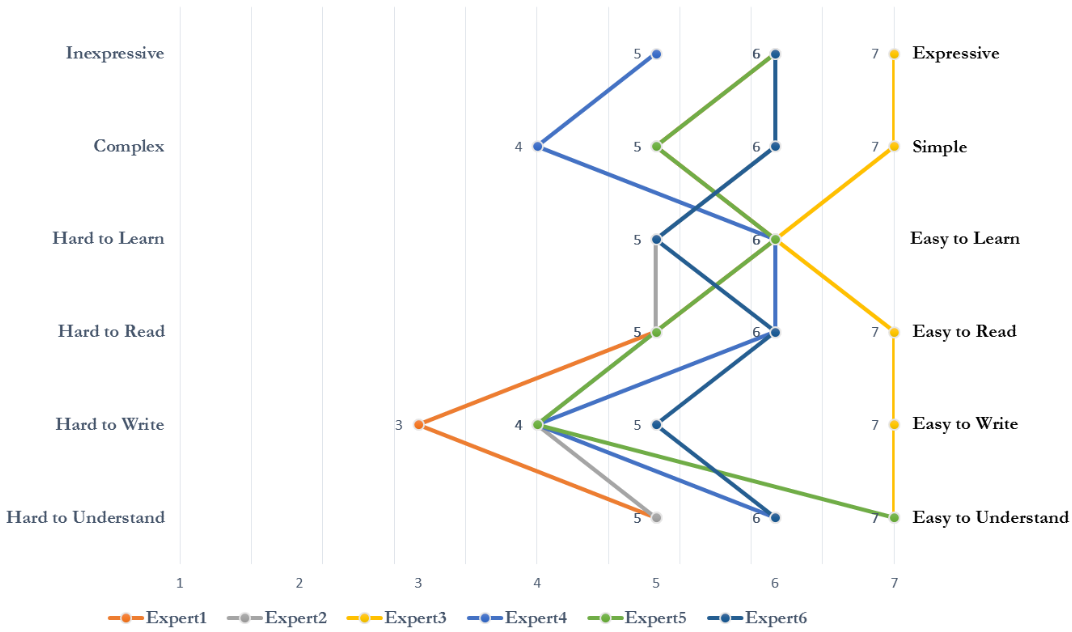


Figure 4.26 - Study D.2: Summary of part 2's results for the SDS-based question "How do you evaluate XUPIL?".

Summarising the results, the global figure shows the responses are positioned on the right side of the scale, which means experts rated the most important aspects of XUPIL positively. Most of the responses are situated between 5 and 6 points. According to these experts, it is clear that XUPIL is considered as being expressive (four of them rated it with 6 points) and easy to understand (two rated this property with 6 points and two rated it with 7). Moreover, we can also conclude with some confidence that the experience gathered by the developers while personalising the prototypes made them consider XUPIL as being quite easy to understand, learn and read. The responses are very similar regarding these three properties.

XUPIL was also considered as being simple, despite the rating of 4 points by Expert4. Her/his assessment of the language is that although being easy to understand and learn, it is not so simple. According to her/him, a developer needs some time to get into it as it has some complexity. Thus, it is not hard to understand and learn, but requires some time to be easily read by a developer. However, a developer will have some difficulties in writing it, according to Expert4. It takes us to the property of being easy to write. The perspective of the experts is that XUPIL presents some difficulties to the developers when it comes to writing definitions of personalisations, which is in line with the tests of part one. At this point, Expert3 even evaluated XUPIL as being a little hard to write and the property is the only one with an average below 5 points. However, on the other hand, we have the case of Expert3, who gave 7 points to how easy is to write with XUPIL. We assume that the case of this expert is somewhat special because, after demonstrating initial difficulties in designing the WeSync personalisation, s/he was very pleased to use XUPIL, showing particular ease and enthusiasm. Maybe her/his scores are somewhat biased by that feeling of satisfaction.

Finally, Figure 4.27 shows the general picture for the last question regarding the use of P²MUCA. We had to leave Expert2 outside this assessment moment since s/he could be very biased after her/his involvement in the implementation of the platform. Therefore, we only have the responses of five participants in this phase of the evaluation, but with significant meaning due to their long-term use of the platform and its underlying solution. According to them, we can assume without doubts that our solution is interesting and appropriate for the goal of supporting developers in the personalisation of applications. Although not as simple as we would like it to be, mainly due to the inherent complexity of the concept, the overall process of personalising an application using P²MUCA is easy to understand.

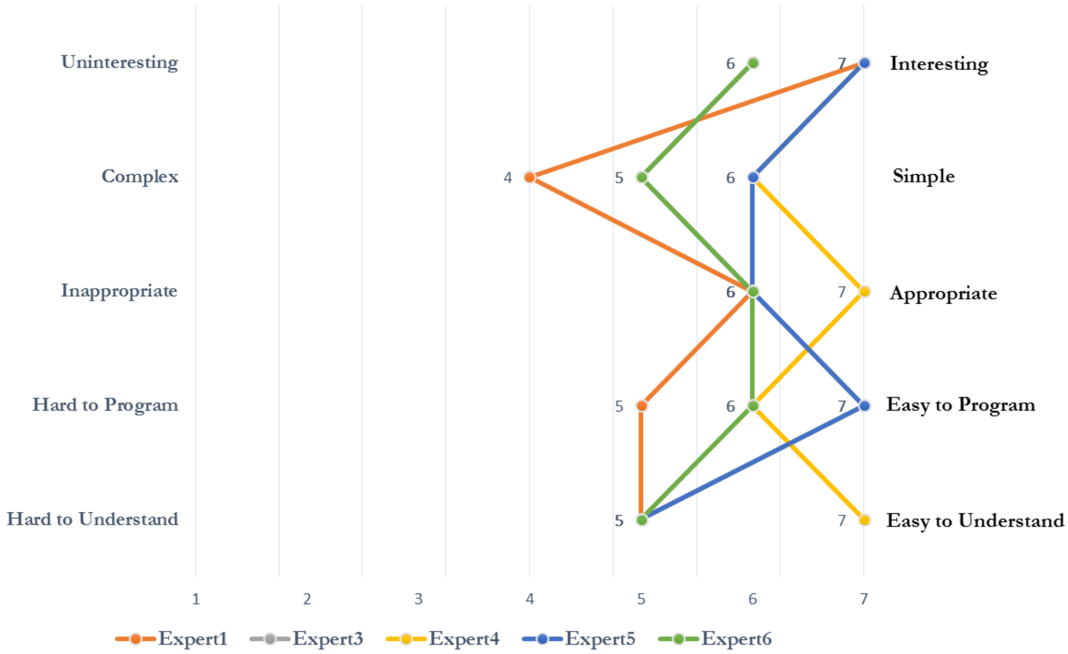


Figure 4.27 - Study D.2: Summary of part 2's results for the SDS-based question "How do you evaluate personalising an application using P²MUCA?".

Finally, it is important to note that, regardless the language used in each prototype, the positive results for the Pair4 ("Hard/Easy to Program") are very interesting and promising. Two experts assigned 7 points to this pair (Expert3 assigned to the last two pairs the same values as Expert5 and to the first three exactly the same as Expert6). If in the first part of the study we had all the participants using a Java-based prototype, in this second part we had prototypes based on three different languages: LEY (Expert1) and BroiStu (Expert6) have been developed in Java, PhysioMate (Expert5) and jPk (Expert4) in C#/.Net and WeSync (Expert3) in Objective-C. We may assume that providing an HTTP-based personalisation API has been a good decision, working like a charm as it was possible to observe from the work of these developers.

4.2.4. Final Remarks

The focus of this study D.2 was the evaluation of the entire personalisation solution, from the general X-Users concept, instantiated by XUPIL, to the P²MUCA platform, which provides the needed personalisation services to the third-party developers. The study was directed to developers with different levels of experience, but it had to include participants with extensive development experience. It was a requirement. Once again, it was interesting to include participants with little experience in the field. One part of the study was based on an experiment conducted with invited developers, while the second part was focused on getting the evaluation from the developers who worked with us on the prototypes. With the participation of those prototypes' developers, which we called experts, we obtained valuable insight from those who have used the solution in the development of full-functional personalised prototypes for several weeks, or months, in two particular cases.

The responses obtained with the first four statements of the questionnaire were very positive and in line with study D.1. The participants of this second study also welcomed our solution enthusiastically, demonstrated an interest in using it, as well as understood and agreed to the model's core concept and context integration. Globally, participants with few years of experience, who were still undergraduate students, presented more doubts regarding some aspects, more difficulties in dealing with the concept and less interest in using the solution in the future. On the other hand, the most experienced participants were much interested in using the solution. This result has been supported by the second part of the study, in which the experts showed even more interest for the solution, based on the very positive evaluation they did to the first set of general statements.

It is noteworthy that for the question related to the specific evaluation of XUPIL, in this second study, we replaced the set of four Likert-type scale statements in study D.1 by a question composed of six bipolar scales in order to get a more in-depth insight on the way participants look into XUPIL. We decided to measure the participants' subjective perception of and affective reactions to important properties of the language and, implicitly, of the model. The results are very encouraging too, revealing that the majority of our participants found XUPIL to be reasonably easy to understand and read, besides being expressive. Globally, we may also conclude that in spite of not being considered very simple, the language is somewhat natural to learn. As a shortcoming, the perception of participants is that the language is not so easy to write. The results are even better if we consider that we did not provide any visual editor that could support the writing of the Walker personalisation during the tests. It means participants had no XUPIL-based auto-completion support and syntax, or even semantic, errors were not reported until model validation by the human monitor who was conducting the tests.

Regarding the evaluation of P²MUCA as a whole, we may draw the conclusion that it is interesting and appropriate for the goal of supporting developers in the personalisation of applications. Although not as simple as we would like it to be, mainly due to the inherent complexity of the concept, the overall process of personalising an application using P²MUCA is easy to understand, learn and program.

The experts, who worked with us during a more extended period, did not present significant differences in the results of the evaluation, compared to the first group. We hypothesised that it could happen. The utilisation of P²MUCA and its underlying model is not something simple to learn and master in the space of hours, or even days, for some lab tests, especially to use it comfortably and confidently. However, the evaluation conducted in the lab revealed interesting positive results, above what we were expecting. Therefore, the evaluation of the experts did not present significant differences, but consolidated both the results of study D.1 and the first part of this second study. Moreover, it worked as an important reference, which helped us to realise how positive has been the part one's evaluation.

Lastly, it is important to note that a high number of participants did perceive the data normalisation issue, which was something that we could not observe in the first study. Maybe this time the participants were able to be aware of this strategy since we gave them more time to prepare themselves for the tests. Additionally, for this second study, we gave them a tutorial detailing the required personalisation, which included examples.

4.3. Summary

This chapter presented the conducted studies with developers in order to respond to the research questions raised previously, particularly $\mathbf{RQ_D}$, but $\mathbf{RQ_A}$ and $\mathbf{RQ_B}$ are implicitly responded.

We have been involved in the development of different applications that worked as proof of concept prototypes in the studies throughout the different phases of the development and evaluation processes. This involvement allowed us to prepare and conduct two main moments of experiments with developers, which were the following:

- Study D.1 (section 4.1) – It was the first attempt of applying the personalisation model, just after finishing its definition and language, XUPIL, which defines the personalisation of an application or system. LEY was used as proof of concept. Being a persuasive pervasive gaming on domestic energy consumption-awareness, the integration of context has been used to thoroughly test the response of the model to cases like this one. LEY was also used to conduct the first study with independent developers.
- Study D.2 (section 4.2) – Its main focus was to obtain the last evaluation of the entire personalisation solution with developers. In the first part, an experiment was conducted with independent developers and using a third-party application called Walker. For the second part, we invited the developers that collaborated with us in the development of prototypes used by this research work. We needed to obtain a deeper insight from their somewhat long-term experience with the solution.

Study D.1 was meant to be as short and quick as possible as we wanted to draw some conclusions in order to improve and fine-tune some aspects of the solution. It was based on the assessment of participants who revealed to have a few years of experience. On the other hand, as a final evaluation, study D.2 was meant to last longer than the first one. Thus, it included a thorough and longer experiment in the lab and the evaluation of the experts. It included participants with several years of experience and post-graduate degrees.

The results of the two conducted studies, both divided into two parts (sub-studies), support that the X-Users/XUPIL-CAPE-P²MUCA triad is a valid solution to the generic personalisation of applications. Moreover, at least to the participants in the studies, there are no tools or platforms that can be used in the same way as P²MUCA, being easy to use, simple to follow, and to be used across applications. There are different alternatives, but these developers do not know them, either because they are academic and closed to public use, or because they are too complex, or because they are less widespread and these participants never demonstrated the compelling interest to find them.

Finally, a thorough study with a greater number of developers using P²MUCA for several weeks or months would be the next step in order to make a final conclusive evaluation that, certainly, would help us refine some aspects of the solution presented in this research work.

5. Studying the Personalisation Solution (End Users-driven Evaluation)

"Develop the power of listening."

(Yogi Teas, 2017)

This chapter presents the conducted studies with end users in order to respond to the research questions raised previously. As previously mentioned, we have been involved in the development of different applications that worked as proof of concept prototypes in the studies throughout the different phases of the evaluation process. Each one of the remaining sections present a conducted study supported by the corresponding prototype. The studies are the following:

- Study #2/EU.1 – The personalisation platform in action for the first time [FCT4U – when private mobile displays meet public situated displays] (section 5.1);
- Study #3/EU.2 – The new application already knows the user [WeSync + WePlayTennis – adaptation to TV delays according to the behaviour of the user] (section 5.2);
- Study #4/EU.3 – Involving end users in the personalisation design [PhysioMate/justPhysioKidding – pervasive physical rehabilitation based on NUI and serious gaming] (section 5.3).

5.1. Study EU.1 – The Personalisation Platform in Action for the First Time

The study presented in this section was conducted having the FCT4U system (see section 5.1.1) as case study. It started right after the conclusion of the deployment of P^{2}MUCA. This time, the user study was directed to end users since we wanted to assess if the system could actually work, building appropriated user profiles based on real end users interacting with a Ubicomp installation like FCT4U. In a first phase, the system gathered data about the users interacting with FCT4U. Then, we analysed those data in order to know the users and to figure out who were the potential personas that could be involved with FCT4U. We defined a set of personalisations instances through XUPIL. Then, in a third phase, end users were probed with a personalised version of FCT4U. This way, it was possible to assess if personalisation was working according to what we were expecting. This study allowed us to track X-Users modelling in action in a realistic environment and to gather live interaction data streams with which we were able to make fine adjustments to the general personalisation modelling.

This study also worked as a first evaluation of P^{2}MUCA. However, since XUPIL is the language that CAPE "understands" and this one is encapsulated by P^{2}MUCA, the assessment of the latter is always intertwined with the assessment of the formers. Thus, this study worked as a field (although FCT4U was installed in a lab, end users were people working in that space) evaluation of the solution's performance as a whole.

Finally, just like in the case of LEY, FCT4U is also a context-aware system and, therefore, the integration of context in the personalisation model is once again validated.

5.1.1. FCT4U: When Private Mobile Displays Meet Public Situated Displays

FCT4U (Santos et al., 2013) stands for FCT (Faculdade de Ciências e Tecnologia), plus 4U, which is a short and informal way of saying "for you". It denotes how the system aims to provide personalised information towards each Faculty's member that interacts with it.

Motivation

Now, more than ever, we have an opportunity to realise an essential long-standing vision of Ubicomp, which is the seamless integration of mobile and fixed infrastructures in order to help support everyday tasks (Want, 2013). It is a real substantial opportunity to take advantage of the well-known ubiquity of personal mobile devices and the fact that situated displays appear more and more in our everyday spaces, having a deployment that is now less expensive.

Those digital displays are becoming increasingly common in different types of companies, public spaces, and even residences, existing in both indoor and outdoor spaces and being of varying size, shape, form, and purpose (Clinch, 2013). Public situated displays are invading the urban spaces (Kostakos and Ojala, 2013) and we are getting closer to the futuristic vision of Spielberg's Minority Report[8] film. Recent advances in display technologies with the consequent falling hardware prices originated the proliferation of (large) public displays (PD), with a significant number of them being even connected to the Internet. New types of display technologies are appearing especially in urban environments (Kostakos and Ojala, 2013).

Situated displays can have an important impact on personal and social behaviour, providing interesting design considerations and challenges. While there is a growing body of research exploring this area (e.g., (José et al., 2008; Davies et al., 2009; Alt et al., 2011; Coutinho et al., 2016)), it remains a complex and normally system-specific effort, due to the lack of generalisations in the association between Ubicomp applications and display resources. In the real world, they have been used mainly for digital signage (Want and Schilit, 2012) and have been also very focused on broadcasting advertising content, for commercial purposes. In fact, advertising is an important factor behind the deployment of public displays since it helps to support the technology to maintain it sustainable (José and Cardoso, 2011).

However, we also envision that complementing digital signage with the addition of interactivity will be essential shortly, since it makes a situated display with digital sign function more useful (Want and Schilit, 2012). The user can interact with public situated displays through touchscreens, controllers or similar objects, gestures with cameras and depth cams, and mobile devices (e.g., smartphones, tablets). Nowadays, these devices are essential tools for our daily living, being almost universally connectable and appearing in nearly all locations.

Bluetooth (BT) has been used to communicate with nearby displays since the early projects on this topic. For instance, Instant Places (José et al., 2008) used BT as a way to sense the presence of individuals near a public display to enable a certain degree of interaction afterwards. A similar system is the e-Campus system (Davies et al., 2009), in which researchers used SMS and BT device names to interact and even to customise the displays. BluScreen (Sharifi et al., 2006) also used BT-enabled mobile devices to influence a public display by detecting the audience around it, which had only a passive interaction with the system.

Therefore, mobile devices can have an important role to play in systems where situated displays networks want to enable new and more engaging user experiences, providing brief encounters of people with information that is relevant for their specific situation. However, letting them have a

[8] Spielberg's Minority Report (2002) on IMDb: http://www.imdb.com/title/tt0181689/

more active role in the system behaviour, with control and more personalisation features. Personalised content will reflect the users' interests and needs, which will be a substantial advantage, and mobile devices can provide the more obvious and direct mechanisms to obtain the preferences and learn the user profile of the users. The traditional digital signage model should support innovative new approaches and content tailored to nearby viewers, which are its ultimate users. For instance, End User Programming (EUP) can be used to empower users with the ability to adapt pervasive public displays to their continuously evolving requirements (Turchi and Malizia, 2015). However, the challenge is how to open public displays to user-generated content, which involve different risks, while being able to efficiently support conformity with the place and display owner expectations (Coutinho and José, 2016).

The personalisation of public displays can be achieved through the BT capabilities of the mobile devices, as seen in (José et al., 2008) and (Davies et al., 2009). Another work presented details of a meaningful attempt at a system that integrates pervasive display networks and mobile devices to support display personalisation (Davies et al., 2014). They described a series of usage models and design goals for display personalisation and presented Tacita, a system that supports those models and goals. Tacita (Kubitza et al., 2013) is comprised of four components in which one of them is an Android application that allows viewers to define content preferences. The first key design decision was to support personalisation by having displays broadcasting their capabilities to mobile devices, rather than having mobile devices broadcasting user preferences or personal identifiers. This approach provides a basic level of scalability and user privacy since mobile viewers are not observable by the infrastructure. A different and interesting approach to follow for our further work is the one by (Kurdyukova et al., 2012), which addresses automatic personalisation based on group context. The system recognises the group of spectators in front of a public display, based on their disposition and gender. It does not require input from the spectator side, neither for training nor for real-time content adaptation.

System overview

The FCT4U system was designed as a mobile and ubiquitous distributed system, which was composed of two main components:

- A mobile application built for Android mobile devices;
- A central node responsible for aggregating all the information from the mobile devices and for powering the public display.

This components division was established because the system is built around a PD installation and its users need a way to interact with that PD. The mobile application was developed to complement the public display experience, while allowing to provide input to the display, adding the needed active interaction mode to the system.

A critical issue of this system is the communication between the mobile devices and the public display. That communication link would have to be bi-directional to allow the public display application to push information to the mobile devices. Multiple solutions were initially considered for that purpose. The usage of BT was put aside due to its technical limitations. Direct communication through TCP sockets was also disregarded since FCT's wireless network blocks direct communication between connected devices. With the network limitations in mind, it came the idea of creating an intermediate service that could be used as a messaging hub between the mobile devices and the public display. This component would not need to be placed in the same place as the public display application, so it is hosted in a publicly accessible network. The system uses WebSockets[9] to avoid running a pure TCP-based server on a non-standard port, which many networks block, and to support web-based clients. It offers a bi-directional communication channel over a TCP connection to be used by web-based applications, but it can be used in other types of applications.

The architecture of FCT4U is designed to accommodate several mobile devices users interacting with a centralised engine placed on the PD installation (see Figure 5.1). Thanks to the bi-directional link provided by the Communication Hub using WebSockets, user input, preferences and other data can be transmitted in real-time from the mobile devices to the systems' FCT4U Engine.

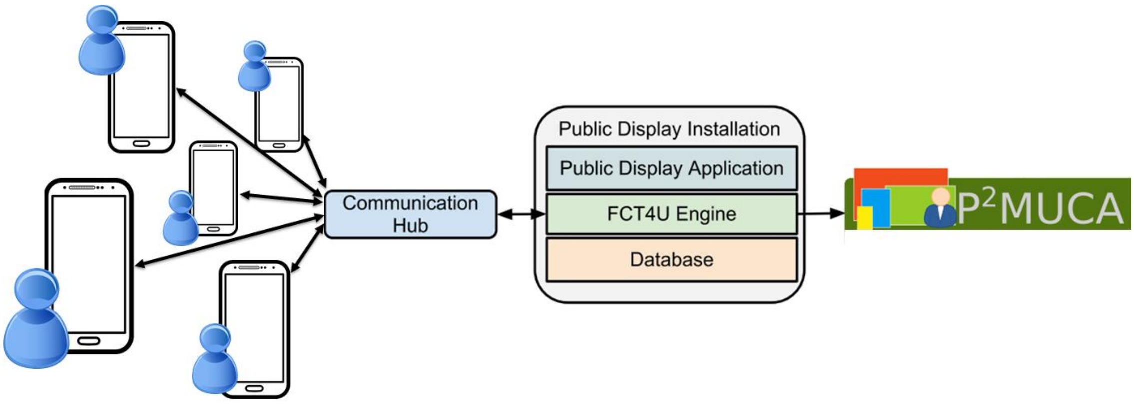


Figure 5.1 – FCT4U: architecture overview.

On the other hand, the engine feeds the Public Display Application with the pre-processed and pre-selected information according to the user profile built from user's preferences, interactions history and context. Such user profiles are built with the assistance of P²MUCA, which provides the personalisation features to FCT4U. Those profiles are also important to show the appropriate information to the user on his smaller, but private screen, of the mobile device, while keeping the ability to control the experience on the larger public display. The engine is currently running alongside the Public Display Application itself; however, most of it is entirely decoupled from the presentation layer and can be separated from it to power multiple displays if the need arises. The engine is backed by the (MySQL) Database using JPA[10] (Java Persistence API) as an abstraction layer. See (Santos et al., 2013) for more details about the deployment of FCT4U.

Rationale and design

The FCT4U system was designed around an interactive public display, which can be seen and used (shared) by several people, while proactively presenting personalised information under continuously changing context conditions. From a user experience point of view, the FCT4U system aimed to provide a public display based on the following principles:

1. Provide information tailored to users:
    a. in the surrounding space;
    b. taking care of privacy according to the context;

[9] The WebSocket Protocol (RFC 6455) by the Internet Engineering Task Force: http://tools.ietf.org/html/rfc6455.

[10] Introduction to the Java Persistence API (in the Java EE 6 Tutorial): https://docs.oracle.com/javaee/6/tutorial/doc/bnbpz.html

   c. sharing the PD with three users, at most, to maintain a well balanced amount of screen real estate;

   d. with and without the need of proactivity from users (active and passive users).

2. Expanding information:

   a. syncing between the mobile display and the public display;

   b. using the larger display to expand and enhance (mainly non-private) information.

3. Sharing information/interaction between users:

   a. sending short messages to others (140 characters limit just like Twitter);

   b. expanding mobile information in order to share it with users around the PD.

There is sensitive information that may only be shown on the mobile device's screen, or it can be shown on the public large screen if the system determines that the user is alone, or among friends, depending on the information's degree of privacy. The PD presents generic information when there are no users. If the number of users is between one and three, then the information is personalised for each one, appearing in a corresponding column. Figure 5.2 illustrates a sample of the information that can appear on the PD when three users are simultaneously interacting with it. Each piece of information in a column is programmed as a widget for each specific type of information. There are currently six widgets available: *map*; *weather*; *lunch menus*; *news*; *messages*; and *friends*.

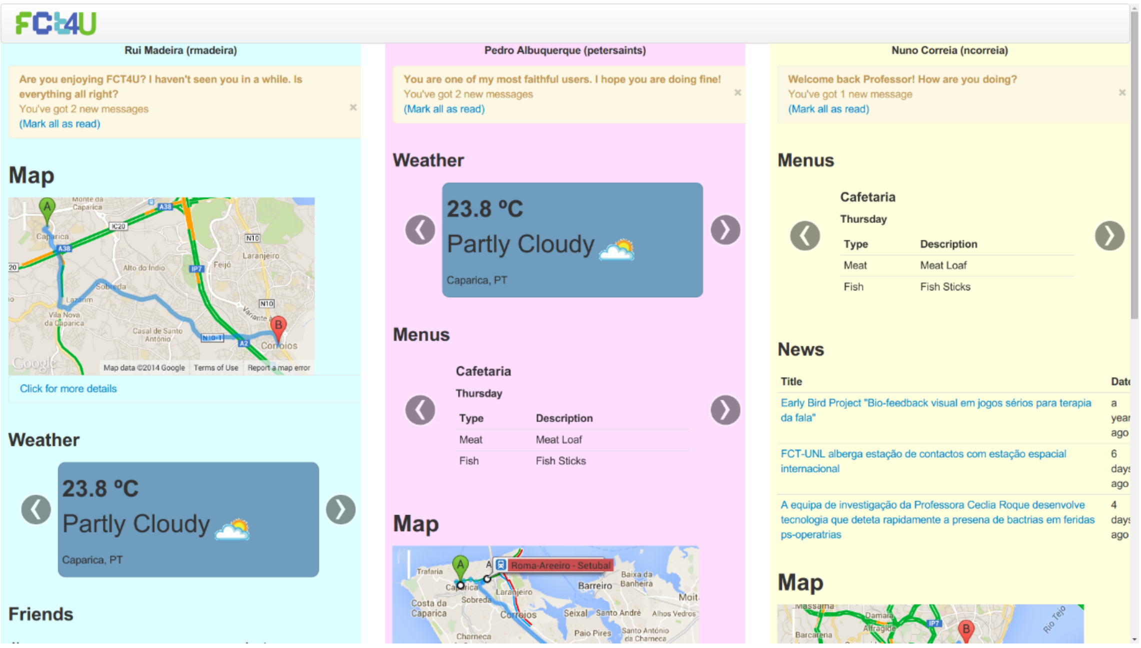


Figure 5.2 – FCT4U's PD UI: screen shared by three users (each one getting personalised information).

The *map widget* has the objective of giving the user information about where s/he is and how to go to where s/he might want to be next. For example, it shows the map centred over the place the user is (the campus) and her/his home. The *weather widget* simply shows the current weather in places that may interest the user. Those places are her/his home location, workplace and the campus itself. The user can use the arrows or swipe her/his fingers to move between different places. A *lunch menus widget* was built to show information about the menu for the current day on multiple bars and restaurants of the Faculty campus. Similarly to the weather widget, the users can move between different bars and restaurants by swiping over the widget or using the arrow buttons.

There is also the *news widget*, which can show news with potential interest to the user. It shows the news title and how much time has passed since it was published. However, if someone clicks on the news title (directly on a touch screen or through the mobile application), it will show a QR Code (the link to the news source) on the public display. Without personalisation, the widget shows information from the Faculty's website feed. The *messages widget* shows personal messages sent by friends. Within FCT4U, a friend is someone that uses the system and is a user's Facebook (it can be extended to other associations) friend. The Facebook profile picture is shown next to the name of the message's sender and the blue button lets the users mark a message as read. The mobile application must be used to select the friend to whom the user wants to send a message and to type the message content (maximum of 140 characters). Besides the widget, an alert bar is shown at the top of a user column with the number of unread messages and a link to mark messages as read. Moreover, the *friends widget* allows users to know if some friends have passed recently nearby the public display's spot. If so, it will show the names of the user's friends, their Facebook's profile picture and how much time has passed since they have been detected.

Some of the widgets, e.g., map and lunch menus, can use location data to provide segmented information in order to be more useful. Favourite locations can be set through user preferences. However, to make the application smarter and more dynamic, a location tracking service was built-in into the mobile application. The service continuously sends the user location to the FCT4U Engine for storage and later analysis.

On the mobile device of a single user appears the information of the column that corresponds to that user (Figure 5.3). The mobile device also works as a command; while interacting with the mobile display on the user's device, the public display is updated in real-time, thus allowing the user to scroll and change the information shown (Figure 5.4). It removes the need for a touchscreen, reducing the cost of the system while providing a user dedicated screen to show the more sensitive private information. In fact, the mobile application is supposed to be the primary user input of the FCT4U system. The interaction of FCT4U also works the other way around; the public display is a touchscreen enabled device so interactions with it will leave the display in a state that will be replicated to the mobile display. The mobile application also extends the public display features by allowing to view the information while away from the public display installation.

The system tries to infer the user's favourite places, e.g., home and work, based on the gathered location. A clustering algorithm was applied to identify places from the raw location data. The location data is also used to try to determine automatically whether someone uses public transports. In order to accomplish this goal, the system counts the number of times a user has been close to the major public transport hub near the campus. If it passes a certain threshold and the user has no explicit preference, then it is assumed that s/he uses public transportation.

A major component of the FCT4U system is indeed the mobile application. It presents two additional important features, which are user registration and detection. The user registration process allows the system to identify each user and to start targeting content for that specific user. FCT4U uses P^2MUCA's credentials as their own, through the use of OAuth 2.0, with an extended token validation endpoint to use OAuth, not only for authorisation, but also for authentication. When a user goes through the authentication process, the application also gathers the BT device address and associates it to the user. This allows the BT discovery routine running on the public display application to identify to which user belongs a detected device.

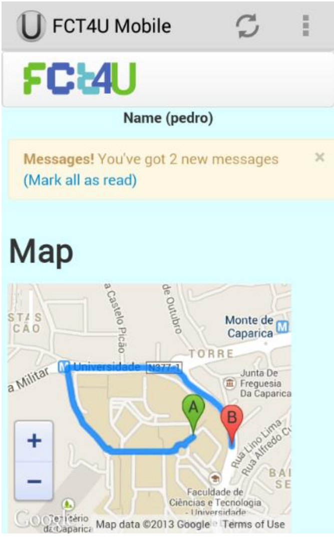


Figure 5.3 - FCT4U mobile UI: information sample.

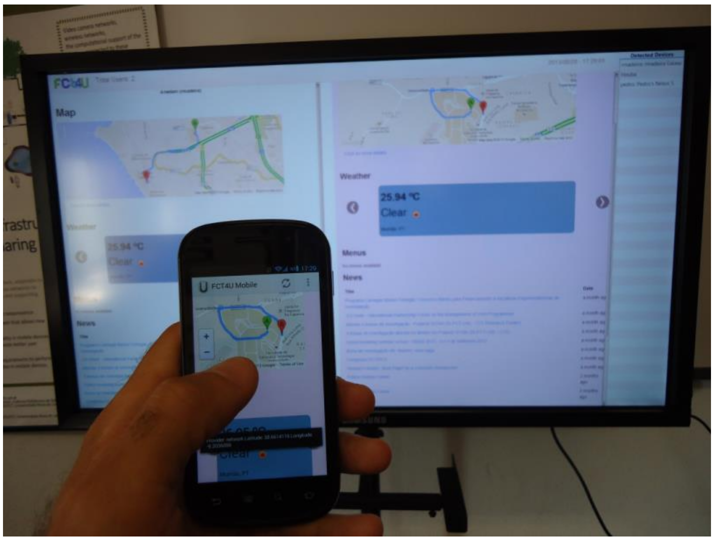


Figure 5.4 – FCT4U: the direct interaction between a user's device and the PD.

5.1.2. Personalisation Design for FCT4U

Personalisation was applied to FCT4U in order to study how better to enrich the user experience, but coping with many potential risks that appear from sharing the public display with users. In our case, with the possibility for a user to leave messages to others, there is the feared appropriation of the display for presenting offensive content, but risks go far beyond that one (Coutinho and José, 2016). Thus, when personalising the public display system, it is necessary to take into account the different risks involved (Coutinho and José, 2016). Additionally, interactions should be adaptable between public and private, depending upon what the user wants to share, without unduly compromising the user's privacy, which results in design implications for personalising public displays (Davies et al., 2014; Parker et al., 2016). Moreover, there is the need for contextual personalisation based upon user activities and objectives at the time of the interaction (Parker et al., 2016).

The system was then first tested among potential end users in order to explore, uncover, and understand personalisation needs of those end users. After observing the first personalisation possibilities, we decided to select the resources set described in Table 5.1. Those resources were selected after watching the users' interaction with the system and analysing gathered interaction data. Some of those resources count the number of times a user uses the application or interacts with a given widget, how much scrolling is done or how often the public display detects the user. Those resources are then used as the basis to create the usual parameters, giving different weights and establishing relationships between them.

A first set of four personalisations for the FCT4U system was established: P1) `map mode`; P2) `privacy mode`; P3) `greetings mode`; and P4) `view order`. The FCT4U's personalisation instances are described as follows (the XUPIL script defining the entire personalisation is in appendix A.2).

P1) map mode

This personalisation is used to determine what kind of information to show on the Map widget. This personalisation is based on a user preference resource - `transportType` - whether the user uses public transit or private transportation (i.e., a car). It also takes into consideration geo-fencing data (`nearPublicTransportHubs`), provided by FCT4U, on how many times a user is detected near the most important public transit hub nearby the faculty's campus. Moreover, the fact that a user is a teacher or a student (`userType`) also helps refining the parameter by giving a larger value to teachers, knowing they are more likely to use a car. Expression (13) defines the parameter used by this personalisation, which is `transportProfile`.

R	Resource	Description
Interaction stream resources		
1	**detectionCount**	[increment] Number of times a user is detected near the PD.
2	**uses**	[increment] Number of times a user launches the mobile application.
3	**interactionCount**	[increment] Number of times a user interacts with any widget.
4	**messagesSent**	[increment] Number of messages sent by a user.
5	**messagesReceived**	[increment] Number of messages received by a user.
6	**messagesRead**	[sum] Number of messages read by a user.
7	**newsClicked**	[increment] Number of news clicked by a user.
8	**newsViewed**	[increment] Number of news viewed by a user.
9	**weatherInteractions**	[increment] Number of interactions with the Weather widget.
10	**menuInteractions**	[increment] Number of interactions with the Lunch Menus widget.
11	**mapInteractions**	[increment] Number of interactions with the Map widget.
12	**mapDetailInteractions**	[increment] Number of times that the Map widget details are toggled.
13	**nearPublicTransportHubs**	[sum] Number of times a user is detected near public transport hubs.
Context resources		
14	**lunchInterval**	[hour interval] current time: before lunch (8h-14h); after lunch (14-8h).
15	**companion**	[value] user is: alone; accompanied; accompanied only by friends.
Preference resources		
16	**userType**	[demographics] the type of user: student; teacher.
17	**transportType**	[question_MC] User's preferred transport type: public; private.

Table 5.1 – FCT4U's resources (data on which personalisations are based on).

$$0.7 * transportType + 0.2 * nearPublicTransportHubs + 0.1 * userType \qquad (13)$$

We defined two possible option values for this parameter, which are directly translated into two personalisation options (Table 5.2): `privateTransport` or `publicTransport`. Users that fall into the former option will have road and traffic information available at the map widget, whereas the latter one will make the widget show the route between current place and home by using public transports, along with proper public transport schedules.

transportProfile	***Personalisation option***
private	privateTransport
public	publicTransport

Table 5.2 - FCT4U: two personalisation options for the "map mode".

P2) privacy mode

This personalisation tries to determine if the user is in a sensitive privacy situation. It takes into account parameters regarding the type of user - `userType` (student: less privacy sensitive; teacher: more privacy sensitive) - and the transport type – `transportType` (public: less privacy sensitive; private: more privacy sensitive). For instance, in the case of a professor it may be undesirable to show her/his home location fully zoomed in by default, so to avoid students knowing exactly where s/he lives.

If indicated in the API call, the personalisation also may use contextual post-filtering by receiving context information from FCT4U regarding the user company (alone or accompanied with friends: less privacy sensitive; accompanied with strangers: more privacy sensitive). A user's company is determined based on other devices detected nearby and to whom they belong. Since Facebook integration is already taken into account for some widgets, it is also used to classify if the detected devices (at the same moment) belong to a friend or not.

Besides using the `transportProfile` parameter explained in the previous personalisation, it also uses the new parameter called `userTypeProfile`. This parameter applies two preferences resources, which are the aforementioned `transportType` and `userType`, which are weighted using the expression (14).

$$0.9 * userType + 0.1 * transportType \quad (14)$$

Lower values will correspond to students and higher values to teachers. Table 5.3 presents the four basic personalisation options based on the combinations of the options for the two parameters. However, since context is here used (as an external context) for post-filtering the results, then we will have 16 personalisation options instead of the former four. Besides these four, each one of the four will be combined with the three possible values for the context (alone; accompanied; accompanied only by friends). If context is not used in the personalisation request then we may have, for instance, the option `studentPrivate`; however, if context is declared then the option will be, for instance, `studentPrivateAccompanied`. Afterwards, the information displayed on the PD is adapted accordingly to the option obtained for the profile of the user and to the context in terms of companion.

		userTypeProfile	
		student	**teacher**
transportProfile	**private**	studentPrivate	teacherPrivate
	public	studentPublic	teacherPublic

Table 5.3 – FCT4U: four basic personalisation options for the "privacy mode".

P3) greetings mode

This personalisation takes into account how often and how much someone uses the application in order to determine if a user is inactive, active or an average user. It also takes into account whether the user is a teacher or a student. It does not power any useful feature, but it may be helpful to give users some feedback about their activities. Therefore, FCT4U has different greetings modes when presenting information to users. It uses the `userTypeProfile` parameter that was already described. `userActiveness` is another parameter, which indicates how active a user is, being defined by expression (15) as follows.

$$0.5 * uses + 0.3 * interactionCount + 0.2 * detectionCount \quad (15)$$

The `uses` resource counts how many times the application was launched by the user; `interactionCount` keeps track of the user's total number of interactions; and `detectionCount` counts the number of times that a given user was detected near the PD installation. The combination of the two parameters creates six personalisation options (see Table 5.4), according to the user activeness (`low`, `medium`, or `high`) level and type (`student`, or `teacher`). The personalisation options are returned to FCT4U, which will adapt the greetings presented to the users on the top of their columns (see Figure 5.2).

		userTypeProfile	
		student	**teacher**
userActiveness	**low**	inactiveStudent	inactiveTeacher
	medium	usualStudent	usualTeacher
	high	activeStudent	activeTeacher

Table 5.4 – FCT4U: six personalisation options for the "greetings mode".

P4) view order

This personalisation focuses on providing the more relevant information first. It uses multiple parameters based on interaction resources that count how much does a user interact with each widget. Each of those parameters breaks users into those that use a widget more than the average and the ones who do not (therefore, two profiles for each other). The parameters (with each corresponding expression) are the following: `communicative` (16); `menusAholic` (17); `newsAholic` (18); `weatherAholic` (19); and `mapAholic` (20). For each one of them, the two options (profiles) names are assigned according to the name of the parameter. For example, in the `communicative` case we have: `communicative`; and `notCommunicative`.

$$0.2 * messagesRead + 0.3 * messagesReceived + 0.5 * messagesSent \quad (16)$$

$$1.0 * menuInteractions \quad (17)$$

$$0.7 * newsViewed + 0.3 * newsClicked \quad (18)$$

$$1.0 * weatherInteractions \quad (19)$$

$$0.6 * mapInteractions + 0.4 * mapDetailInteractions \quad (20)$$

By combining all those possibilities, a series of personalisation options is defined which allows the application to know which widgets tend to be the most preferred by each user. Given that information, it is straightforward to adapt the interface to show the widgets a user uses the most first followed by the ones s/he uses less. It is noteworthy to refer that the combination of the parameters' options beginning with not (i.e. `notCommunicative`, `notMapAholic`, and so on) results in the personalisation option `none`, meaning a persona who does not have a clear preference for specific widgets over others.

Additionally, the companion resource is taken into account in options that have the `communicative` parameter set to `communicative`, so that the system can decide if it should show messages on the public display, due to their private nature. In order to take account of cases in which the companion context is not available, options without it are also considered such as in the case of the personalisation P2 (privacy mode).

Moreover, since this personalisation is mostly based on user interaction habits, context segmentation was also used to take into account which widgets are more common depending on the time of day. The day was broken into before lunch (BL) [08:00-14:00] and after lunch (AL) [14:00-08:00] using a context resource. As such, the interaction stream resources are stored separately for those intervals, besides the global values. The resources values indexed to the context values will weight 50% in the calculation of the parameter values. This way, the personalisation option returned for BL may be different than the one returned for AL. For instance, if someone uses the Lunch Menus widget a lot during the morning, but not so much during the afternoon, the system may put the person into different usage profiles depending on the time of day.

The name of each personalisation option is representative of the combination on which it is based, and it is interpreted by FCT4U to reorder the widgets shown on the web UI.

5.1.3. User Study with End Users

This study would be the first one towards having the P²MUCA platform responding to requests from a Ubicomp application (a context-aware system, in this case). Therefore, the study was planned to include two phases, using:

1. FCT4U without activation of personalisation features to:
   - conduct users tests to evaluate the FCT4U's concept and usability, and guide the system's personalisation design;
   - gather initial data about the users interacting with FCT4U.
2. a personalised version of FCT4U in order to:
   - conduct user tests to understand if applying personalisation can improve the user interaction with the system;
   - evaluate the response of P²MUCA.

Design and participants

These tests also took place in a computer science lab at our University campus (see Figure 5.4 and Figure 5.5), but in this case because we needed to have the installation of the PD in a place that we could control and supervise. On the one hand, we needed to have a full installation that stood stable for some weeks and, on the other hand, it was important for security measures since we needed to protect the involved equipment. A 46-inch touchscreen was used for the public display installation and different Android-based smartphones and tablets were used to run the mobile application.

We recruited volunteers that could be potential end users of the system in the near future and asked them to use and test it. First, we briefed each participant, taking the chance to explain the concept behind FCT4U, its main functionalities, the different interactions that were included, and what we intended to obtain with the user study. In the first moments of interaction with FCT4U, the tests were conducted with researchers observing and taking notes.

In phase 1, we conducted a first user testing session in which participants described their thoughts, observations, and actions as they interacted with the first FCT4U prototype (Santos et al., 2013). We were concerned with the evaluation of the idea and concept behind FCT4U; if participants believed this is a useful system in terms of supporting and helping their daily lives in the Faculty campus. We were primarily focused on evaluating the interaction modes and the pieces of information made available without using personalisation. The first phase had a duration of 15 days, and it was also used to collect interactions' data to build the participants' user profiles just before applying personalisation to FCT4U. Afterwards, we initiated the second phase and, after one week, a second user testing session was held with the same participants to evaluate FCT4U with personalisation activated (described in the previous section).

The following procedure was the same for the two sessions of evaluation. Questionnaires were favoured over personal interviews, although the latter appeared informally while observing the testing tasks. Thus, we asked the participants to fill in anonymously a questionnaire covering some important questions concerning our work. The questionnaire started with a group of questions to characterise the participants regarding age, degree and role in the institution. Other questions were included to get feedback on the experience of the participants with mobile devices and public displays installations.

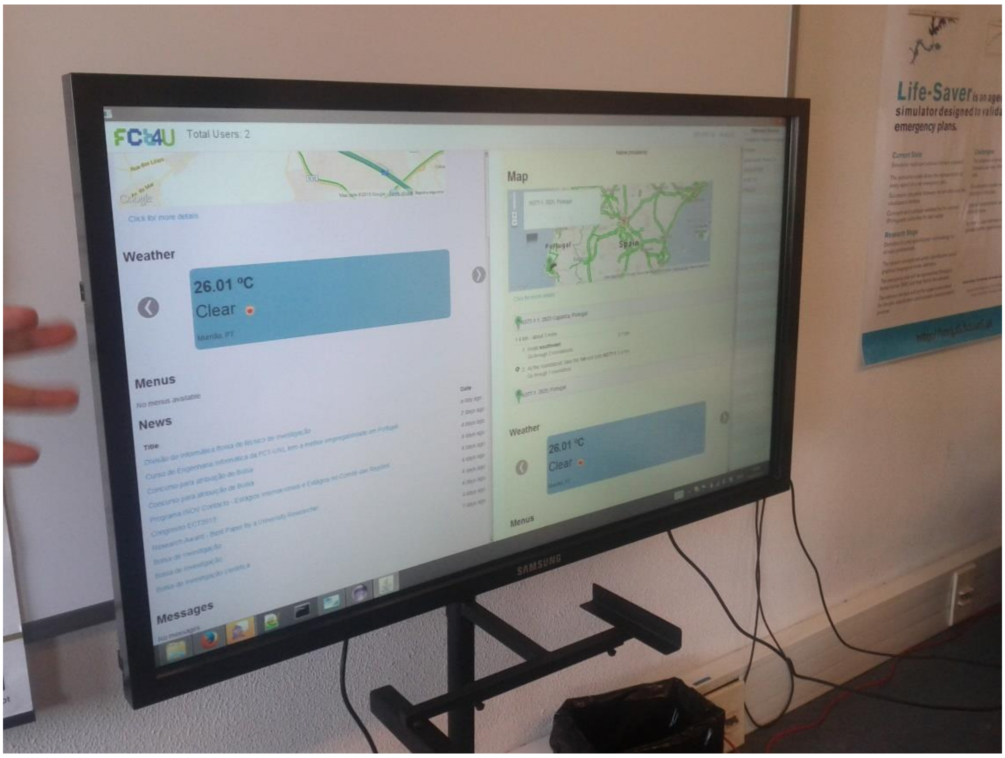


Figure 5.5 – FCT4U: the lab installation of the public display system.

Then, in the following part, the questionnaire focused on evaluating the user experience with FCT4U, taking into account three scenarios:

1. passive interaction by just being detected by the system and seeing information in the PD;
2. active interaction using a mobile device running the FCT4U's Android application;
3. active interaction using the touchscreen modality of the public display.

Participants were asked to rate multiple statements, using a seven-point Likert-type scale, which ranged from strongly disagree (1) to strongly agree (7). Statements regarding usability aspects were based on the SUS questionnaire (Brooke, 1996; Brooke, 2013). Table 5.5 presents only the main statements that are of interest to this particular research. The questionnaire had more statements in order to cover all the interaction and usability aspects that were important to evaluate the global design of FCT4U. However, here we are mainly interested in statements that allow us to evaluate the effect of personalisation on the user experience.

We asked people that were working in our research centre on a daily basis to interact with the system. Eleven volunteer users did it for a period of 15 days. Nine of them, aged between 26 and 39 years ($\bar{x}$ = 30.5; σ = 3.90), took part in the whole user study responding to both phases' questionnaires. Two of them responded only to the second one. Two of the participants were female. All participants had a master's degree, except one who was a researcher/professor with PhD degree.

Statements
S1. I have experience when it comes to interacting with public displays.
The passive interaction
S2. The system responds quickly to my physical presence showing information directed to me in the public display.
S3. I am satisfied with the information that appears without the need of an active interaction on my part with the display.
S4. I am satisfied with the order in which the information widgets appear in the display.
S5. This system is useful when I am in a hurry and have no time to stop for a while, having the potential to show important information to me.
S6. The passive mode of the FCT4U system saves me time when I use it.
S7. The passive mode of the FCT4U system can save me even more time when I use it if the information and the widgets order are adapted to my profile.
Personalisation issues
S8. I want to have a more personalised experience with the use of FCT4U.
S9. I do not mind sharing my current location to have a more personalised experience based on where I am at certain times.
S10. I do not mind that FCT4U infers where i usually have lunch in order to present me personalised lunch menus.
S11. I want a privacy mechanism in FCT4U to control what information related to me appears in the public display.
The whole FCT4U system
S12. The FCT4U system is easy to use.
S13. The FCT4U system is user friendly.
S14. The FCT4U system requires the fewest steps possible to accomplish what I want to do with it.
S15. With the use of the whole FCT4U system I can recover from mistakes quickly and easily.
S16. The FCT4U system is suitable to help enhance its users' activities on campus through a pervasive way.
S17. The FCT4U system does everything I would expect it to do.
S18. I feel very confident using the FCT4U system.
S19. I would recommend the FCT4U system to a friend or colleague.
S20. I would like to use this FCT4U system frequently.

Table 5.5 - Study EU.1: Questions and statements of the questionnaire.

Regarding technologic aspects, only one of them did not use a smartphone as a personal device. Five of the nine had an Android-based smartphone and another one had an Android-based tablet. Therefore, almost 70% of the participants could use the current system in their daily living. After talking with the participants, and from the questionnaire results, it got clear that they were already used to have contact with display digital signage, but only three already had contact with interactive installations (see Figure 5.6). This was somewhat expected since participants were mainly from a research group focused on interactive and multimedia applications, thus, presenting some experience (S1: $\overline{x}$ = 4.4; σ = 1.34) in the subject that could benefit the evaluation process.

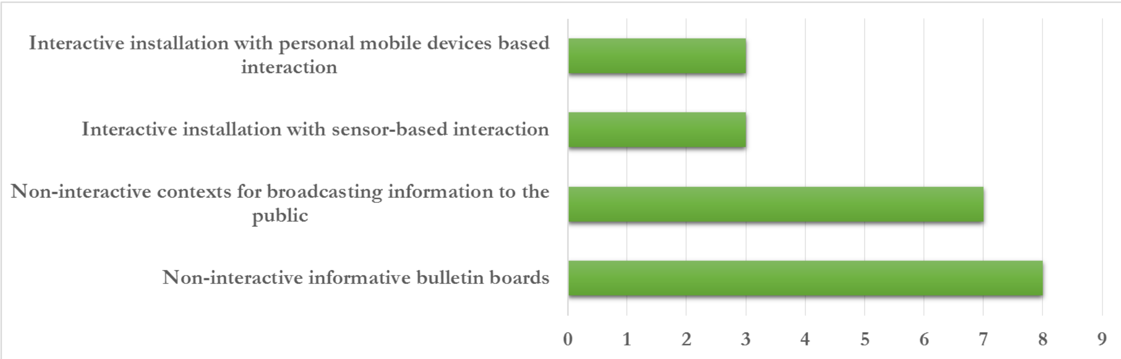


Figure 5.6 - Study EU.1: Number of participants / public display installation type.

Results of the user profiling for personalisations

CAPE collected enough data to build relevant user profiles after 15 days of use, surpassing the usual cold start issue. At that moment, personalisation requests resulted in the personalisation options by participant (user) seen in Table 5.6. The results show that our approach was able to categorise the users in a wide array of personalisation options mainly based on the user's interactions, sometimes done under certain context conditions, according to the different resources that have been selected to create the parameters, which define the different user profiles.

The `view order` personalisation is the most complex and interesting due to a wide range of personalisation options that it presents and the integration of context segmentation. For instance, we noticed that users were most active before lunch (BL). We may draw this conclusion supported by the raw data that were gathered, but the results in Table 5.6 corroborated it since in all cases, except for user 10, the personalisation options obtained for the `view order` BL are the same as for the `view order` without context segmentation (global results). It means interactions BL were in a considerably higher number, thus weighing more in the calculation of the global values. Also interesting is the fact that the lunch menus widget is very popular during the time interval of BL, as it was expected according to our previous user study (Santos et al., 2013). The usage drops significantly AL, which results in the fact that, aside from user 10, the lunch menus widget is no longer actively used. Moreover, the personalisation options for the `view order (AL)` also show that, globally, users were usually less active interacting with the widgets in that period of the day. Five users even got the option of `none`, meaning the activity dealing with the widgets was much reduced or even none. They will be presented with the global `view order,` which is the default, corresponding to a non-contextual situation regarding the period of the day.

The `greetings mode` indicated three users were the most active ones (7, 8 and 9) and, in contrast, there were other three (5, 6 and 11) that were labelled as less active (`inactive,` in terms of the name we gave in the personalisation definition). Eight of the users will be presented with a `map mode` directed to the public transports information. It is an expectable result since this personalisation is much based in a preference stated by users, which shall be in accordance with their practices.

Finally, in both `privacy mode` and `greeting mode` we consider if a user is a teacher or a student. This is tricky to evaluate since some participants are researchers that could be considered as being both. For instance, a researcher that is a Ph.D. student and also a teacher. Therefore, the results will be in agreement with what the participants declared to be, since the parameter used to categorise them is mainly based on a preference.

User	Personalisations					
	Map Mode	**Privacy Mode**	**Greetings Mode**	**View Order**	**View Order (BL)**	**View Order (AL)**
1	privateTransport	studentPrivate	usualStudent	weatherMenus Map	weatherMenus Map	none
2	publicTransport	studentPublic	usualStudent	communicative Menus	communicative Menus	communicative
3	publicTransport	studentPublic	usualStudent	weatherMenus News Map	weatherMenus NewsMap	news
4	publicTransport	studentPublic	usualStudent	weather Communicative Menus	weather Communicative Menus	communicative
5	publicTransport	studentPublic	inactiveStudent	map	map	none
6	publicTransport	studentPublic	inactiveStudent	weatherMenus News	weatherMenus News	news
7	publicTransport	studentPublic	activeStudent	weather Communicative MenusNews	weather Communicative MenusNews	news
8	publicTransport	teacherPublic	activeTeacher	menusMap	MenusMap	none
9	publicTransport	teacherPrivate	activeTeacher	weatherMenus Map	weatherMenus Map	none
10	privateTransport	teacherPrivate	usualTeacher	communicative MenusMap	communicative Map	Communicative **Menus**Map
11	privateTransport	teacherPrivate	inactiveTeacher	weatherMenus	weatherMenus	none

Table 5.6 – FCT4U: Personalisation options obtained per user participating in the tests.

With the activation of personalisation, FCT4U should respond to those personalisation options received from P²MUCA, processing them to adapt its interface accordingly by switching elements, hiding parts of the user interface, providing extra/different information, etc., according to the personalisations defined in section 5.1.2.

Results and discussion of the user tests

In this particular research work, we are using only the results from statements (S2-S7) regarding the first of the three scenarios considered in the tests. Here, we analyse the results for that scenario, in which the user is passive, since it presents the elements needed to evaluate the system without and with personalisation from the participants' perspective. Figure 5.7 presents a summary of phase 1's results (FCT4U without personalisation), whilst Figure 5.8 summarises the results obtained in phase 2 (FCT4U with personalisation).

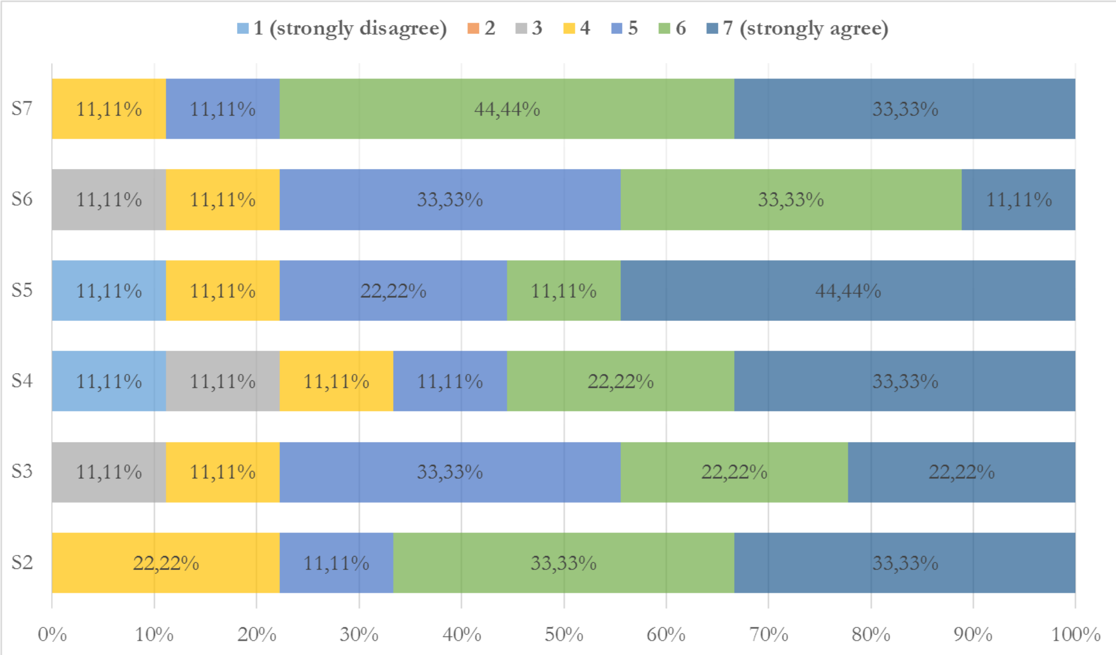


Figure 5.7 – Study EU.1: Summary of phase 1's results (FCT4U without personalisation) for S2-S7.

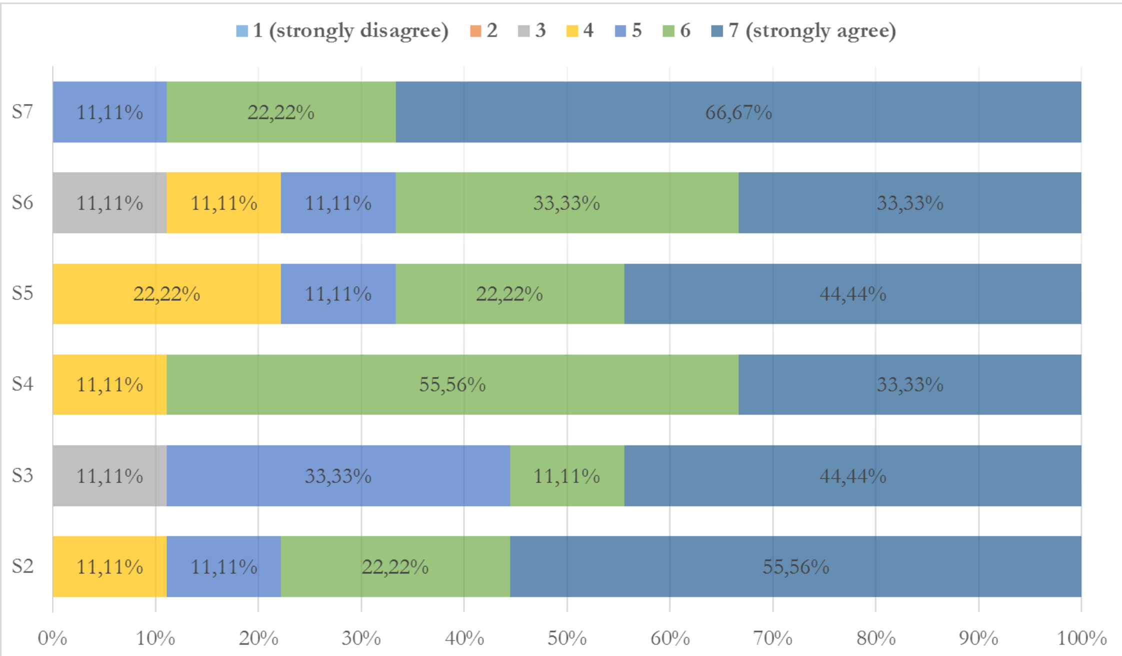


Figure 5.8 - Study EU.1: Summary of phase 2's results (FCT4U with personalisation) for S2-S7.

From that first scenario, in which the users stood still in front of the public display once they were detected, it was possible to conclude that, without interaction mechanisms, only a small portion of the available information could be seen (mainly the Map and Weather widgets). Therefore, most of the participants (7 out of 9) considered that the time it took the system to detect their presence was well above adequate ($\overline{x} = 5.8; \sigma = 1.13$). Moreover, it seems that they also considered the information was well adequate to them. Statement S2 ("The system responds quickly to my physical presence showing information directed to me in the public display.") is not very clear regarding what it was intended to evaluate since some participants had doubts, not understanding it very well. We explained we wanted to assess the "quick response" materialised with proper information, if the participant felt the information as being "appropriate". Globally, they liked what they saw in terms of information, although not being personalised in the first session of tests. As it was expected, the results improved in phase 2 with the personalised version of the system ($\bar{x}$ = 6.2; σ = 1.03). Only one participant (p7) did not agree with S2, maintaining the four (4) assigned in phase 1. Other two kept the original values of 5 and 6, which were already positive.

In general, the participants considered that the information provided by this passive approach of the public display was just barely above appropriate ($\bar{x}$ = 5.3; σ = 1.25). The results of S3 ("I am satisfied with the information that appears without the need of an active interaction on my part with the display.") could be better, but it seems participants felt that something could be done to improve what they get from the display. Perhaps the order of the widgets, the number of visualised widgets, the information selected, among other aspects, was not very satisfactory considering that they knew they could actively interact with the system. Once again, the results got more positive with the activation of personalisation in phase 2 ($\bar{x}$ = 5.8; σ = 1.31). Only one participant did not agree with S3, maintaining the value of 3 assigned in phase 1, and it was the same user that maintained the value of 4 between phases regarding the previous statement.

Before using the personalised version of FCT4U, only six participants were really satisfied with the order in which the information widgets appeared in the display (S4). Although two of them strongly agreed with S4, responses were widely distributed presenting two participants not agreeing and one with neutral opinion ($\bar{x}$ = 5.1; σ = 1.97). This statement presented the worst result, however it improved a lot in phase 2 ($\bar{x}$ = 6.1; σ = 0.87). The positive results obtained in that second session of tests were closely related to the fact that one personalisation (view order) was implemented exactly to address the widgets order. This time, participant p7 has increased her/his agreement from a value of 1 to 4. However, s/he maintained a neutral opinion.

Most of the participants (7 out of 9) gave a positive score to the system usefulness through statement S5 ($\bar{x}$ = 5.4; σ = 1.89). Globally, personalisation made them agree even more ($\bar{x}$ = 5.4; σ = 1.89), which is very positive. Finally, in general, the participants considered that the system had the potential of saving them time ($\bar{x}$ = 5.2; σ = 1.13), especially if the widget order and content would be tailored for their needs and wants ($\bar{x}$ = 6.0; σ = 0.94). If the results of phase 1 already demonstrated a global agreement with statements S6 and S7, and the need of having a system tailored to its users, then phase 2's results for S6 ($\bar{x}$ = 5.7; σ = 1.33) and S7 ($\bar{x}$ = 6.6; σ = 0.68) were clearly in line with the previous results. Participant p7 only agreed with one statement, which was S6, and in phase 2, demonstrating s/he was coherent and, mainly, skeptical about the potential of FCT4U.

The results obtained with the six statements (S2-S7) undoubtedly give us a positive global scenario regarding the personalisation of essential content and interfaces made available through FCT4U without the need of an active interaction by its end users. Personalisation improved the participants' perception towards the usefulness of what is provided by FCT4U.

The following four statements addressed personalisation issues more directly. We wanted to know if participants would like to have a more personalised experience when using FCT4U (S8). In phase 1, seven participants agreed with that intention and only two were undecided (see Figure 5.9), maybe because they would not know what to expect ($\bar{x}$ = 5.3; σ = 0.94). It is curious that in the second phase, the results are very similar with the same value of $\bar{x}$ = 5.3, but we found out participants increasing the score and others decreasing it, thus with a value of σ = 0.47, which indicates that all participants agreed with S8 (see Figure 5.10). What happened is that some understood the benefits of personalising the system, increasing the agreement value, and others felt a less need for it since they were happy with the results. Nonetheless, the results show there is still space for more personalisations since globally its potential end users still want it.

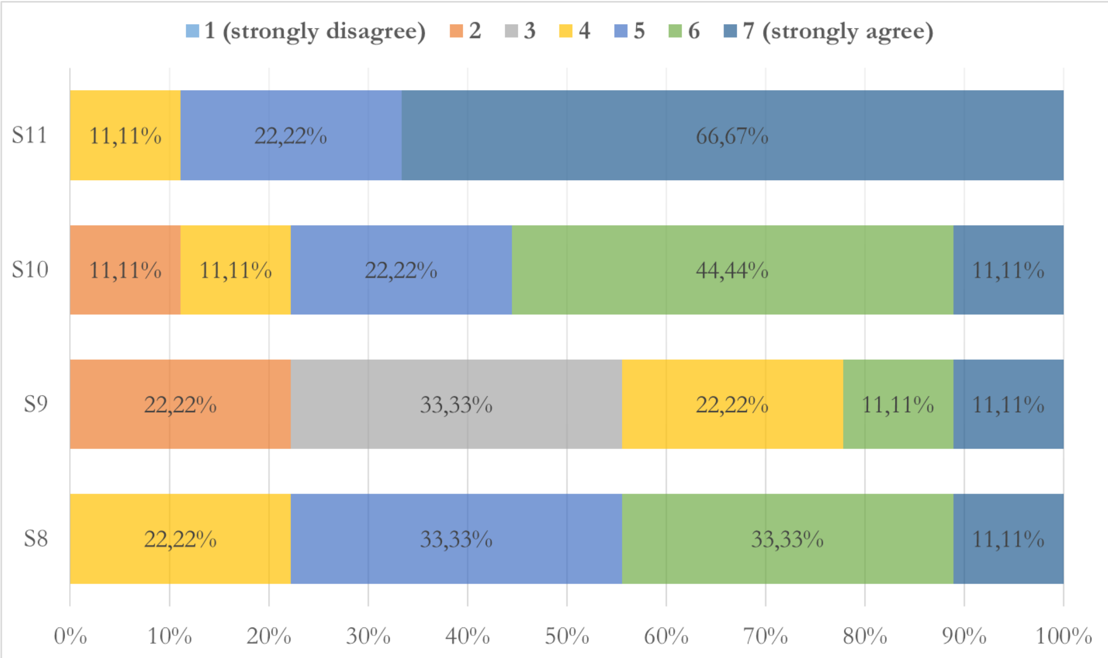


Figure 5.9 – Study EU.1: Summary of phase 1's results (FCT4U w/out personalisation) for S8-S11.

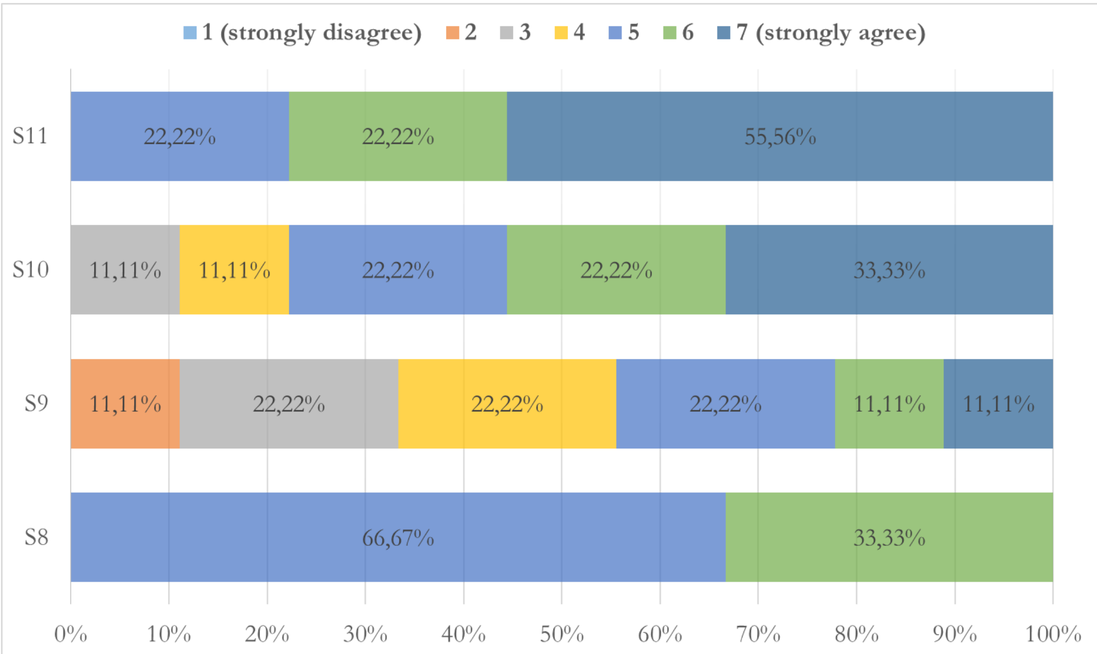


Figure 5.10 - Study EU.1: Summary of phase 2's results (FCT4U with personalisation) for S8-S11.

The results obtained with statements S9 and S10 are very interesting. Participants clearly did mind sharing their current locations to get a more personalised experience based on where they are at certain times ($\bar{x}$ = 3.8; σ = 1.62). Only two agreed with the statement. The result for S9 is an indication that potential end users need a method to control their data, particularly sensitive personal data, needing to trust in our system. However, participants substantially agreed with S10 ($\bar{x}$ = 5.2; σ = 1.40), as they felt it would be something more controlled towards presenting them with a better lunch menus, which is a feature they appreciate. This statement S10 was much more focused then the higher scores in the responses. The results got more positive with the personalised version of FCT4U in phase 2. Only three participants did not agree with S9 ($\bar{x}$ = 4.3; σ = 1.49) and there were more of them strongly agreeing with S10 ($\bar{x}$ = 5.6; σ = 1.34). Participants felt the benefits and, after a period using the system, they could trust it a little more.

However, the results for statement S11 ("I want a privacy mechanism in FCT4U to control what information related to me appears in the public display.") indicate that participants still want and need a mechanism to feel in control over the system and its personalised features (phase 1: $\bar{x}$ = 6.2; σ = 1.13 / phase 2: $\bar{x}$ = 6.3; σ = 0.82). The result of S11 was expectable; thus, we implemented the mobile interface also to provide them with control over some aspects of FCT4U, allowing them to partially manage their data.

Finally, it is important to summarise the results regarding the last ten statements presented in Table 5.5, which address the user experience and usability of the system as a whole. We selected those statements from the whole set applied in phase 1 since they could be the most representative of what we wanted to find out between the two phases of tests. Therefore, the remaining results about FCT4U as a whole can be seen in Figure 5.11 and Figure 5.12, regarding phase 1 and phase 2, respectively.

Overall, the system received positive responses from the participants regarding statements S12, S13 and S14, even in the first phase. The fact they were more accustomed to using it and the integration of personalisation may be the two factors that improved the results in the second stage. Indeed, participants were very satisfied with FCT4U in terms of being easy to use, user-friendly and requiring few steps to work with it.

Moreover, the results obtained for statements S18, S19 and S20 indicated another important dimension, which is the confidence evidenced by the participants. Once again, the results are even more positives in phase 2, but, overall, they show how participants are willing to use the system in the future (S20), being available to recommend it to others (S19), which may be an evidence of how they were confident about it (S18).

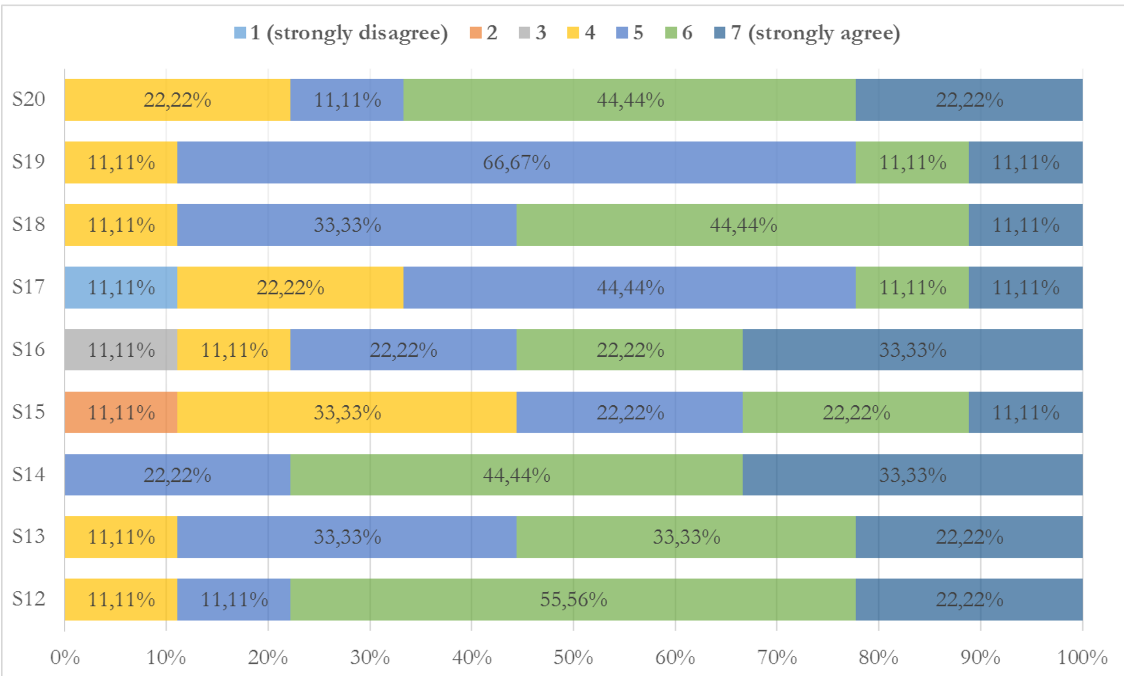


Figure 5.11 – Study EU.1: Summary of phase 1's results (FCT4U w/out personalisation) for S12-S20.

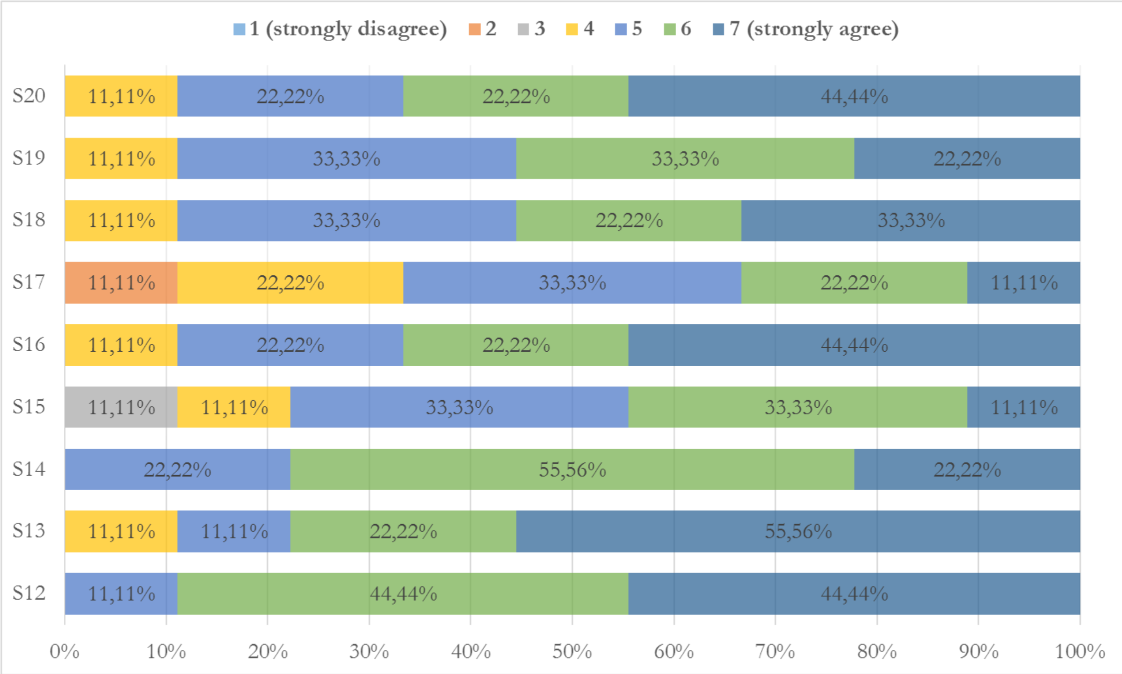


Figure 5.12 - Study EU.1: Summary of phase 2's results (FCT4U with personalisation) for S12-S20.

The worst results were obtained with statement S15 ("With the use of the whole FCT4U system I can recover from mistakes quickly and easily.") and S17 ("The FCT4U system does everything I would expect it to do."), which although presenting positive scores indicate the system has some room for improvement. Nonetheless, personalisation can help improving these two dimensions if it is well designed, taking them into account.

5.1.4. Final Remarks

It was still impossible to offer a fully personalised system after 15 days os usage since a more substantial period of usage would be needed to gather more interactions stream data. Then, it would be possible to better analyse the users' behaviours to study how to improve personalisation, working on the parameters and selecting appropriate resources. This cannot be considered a limitation of the model, but as a process that is needed to make the best possible personalisation choices. An initial data collection period is a procedure to be expected in any (semi-)automatic user profiling system.

Furthermore, the results proved that the personalisation of a few elements of FCT4U improved the global perception of the participants about FCT4U's goals and on how the system can support them on a daily basis. Moreover, this study with end users allowed us to confirm that they still want and need a mechanism to feel in control over the system and its personalised features. It is important to address users' privacy concerns adequately to increase their trust (Teltzrow and Kobsa, 2004) in FCT4U, and in similar personalised applications, retaining some user control over the experience (Schade, 2016b).

This study also worked as a first evaluation of P²MUCA. However, since XUPIL is the language that CAPE "understands" and this one is encapsulated by P²MUCA, the evaluation of the latter is always intertwined with the evaluation of the formers. Thus, this study allowed to test the solution's performance as a whole, leaving promising results since it worked in response to a realistic scenario.

Once again, we presented the definition of context-aware personalisations, showing how context can be used to segment essential data. Therefore, they allowed us to test both the model and XUPIL to prove their viability. The expressiveness of the personalisation options definition appear as a limitation. The number of possible combinations grows exponentially when dealing with many parameters (for instance, the viewOrder personalisation in appendix A.2). Thus, it would be useful to be able to define that a given parameter does not matter for a specific personalisation option, which would cut the number of personalisation options needed to cover the combinations of parameters.

5.2. Study EU.2 – The New Application already Knows the User

The main purpose of this third study was to respond to **RQ$_C$**. Therefore, it was designed to assess if the user interaction with a first application can be useful to personalise the experience of the same user when interacting with a second application. This is a formulation that appears because our pervasive generic personalisation solution can promote interoperability between personalised applications. Once again, the user study was directed to end users. It was conducted after the evaluation of P^{2}MUCA, in the previous study, since we had to be sure the platform was fully working, responding well to all requests.

Therefore, P^{2}MUCA is once again validated and since its evaluation is always intertwined with the evaluation of X-Users/XUPIL and CAPE, this study worked as an evaluation of the solution's performance as a whole.

For this study, we had the collaboration of another research team that asked us to apply our solution to their project, regarding the development of a synchronisation mechanism based on the user behaviour to deal with TV delays in second screen scenarios (Centieiro, Romão, et al., 2015a; Centieiro, Romão, et al., 2015b). We worked with them in the development and the studies around two prototypes, WeSync and WePlayTennis, having the responsibility of designing a solution to cope with potential cheaters in a competition-based scenario (Madeira et al., 2015), whilst the synchronisation mechanism study was their responsibility.

5.2.1. WeSync and WePlayTennis: Adaptation to TV Delays according to the Behaviour of the User

WeSync and WePlayTennis are two second screen apps that prompt users to guess the outcome of situations in, respectively, football and tennis matches that are being broadcasted on the TV.

Motivation

It is possible to find two kinds of people when it comes to the living room: those for whom the mobile devices do not exist when the TV is on, and those who cannot put their devices away when watching TV (Chmielewski, 2014). The proliferation of mobile devices, such as current smartphones and tablets, allows TV viewers to become accustomed to "doing their own thing" (Videonet, 2014). There is a whole category of second screen apps targeted to those who want to go one step further and complete their TV show experience. Thus, it is possible to take advantage of the millennials' inability to put down their mobile devices (Nielsen Holdings, 2014) by creating useful, interactive apps that complement rather than distracting users from the TV. These apps work as promising vehicles to accomplish personalisation since they can provide show-related information and support "social viewing" through dedicated social networks, and even interactive experiences, through polls and quizzes, synchronised with the show content.

However, there is usually a delay when receiving TV signals. It is common for some viewers to get events on second screen apps that are not synchronised with the corresponding key moments in the TV broadcasts. The users' engagement can be ruined when a situation like this one occurs, since the app content may be presented too early (spoilers), or too late (obsolete content). Synchronisation can affect the user experience of the viewers, mainly when receiving key events on an app before watching them on the TV broadcast. Thus, second screen synchronisation is an important step towards TV personalisation (Videonet, 2014).

For instance, relative delays encountered in digital TV degrade the soccer watching experience, especially when there are fans close to each other (i.e., neighbours) watching non-synchronised TV feeds (Mekuria et al., 2012a). A study in which users were prompted to bet, during a football match TV broadcast, if a goal would happen in a few seconds showed that some users were frustrated or stressed for not being able to perform the action (Centieiro et al., 2014). This happened since the match on TV was delayed relatively to the real match and consequently to the match key events on the second screen app. It is in-line with the study of (Kooij et al., 2014), which states that a variation of the playout delay can go up to 6 seconds in TV broadcasts in the same country, and more than one minute in some web-based TV broadcasts.

There are solutions for the synchronisation between a second screen app and the TV without the need for hardware modifications, such as using automatic content recognition (ACR) on the TV broadcast audio to synchronise each viewer's app, or relying on inter-destination media synchronisation (IDMS) to synchronise all the viewer's TVs (Centieiro, Madeira, et al., 2015). However, viewers are dependent on the TV service providers to implement these solutions, or else the synchronisation issue will remain. On the other hand, it is possible to use the Synchronisation Mechanism through User Feedback (SMUF), which relies on the feedback given by the users on how apps' key events are synchronised with the corresponding TV broadcasts' key moments (Centieiro, Romão, et al., 2015b).

However, a mechanism like that one presents a negative counterpart within gaming scenarios since users can easily find out how to exploit it. If users set up a delay higher than the one they actually have, then they will end up receiving the content on the second screen app after watching the corresponding content on TV. This can also happen in situations in which the synchronisation is made by ACR methods, such as audio fingerprinting, where users can synchronise the app with the TV box at a past moment. In both cases, users will be able to win more points since they can, for instance, answer more questions correctly and within shorter periods.

In our context, these users are cheaters since they exploit the system to get an unfair advantage or achieve a target that they are not supposed to (Yan and Choi, 2002; Yan, 2003). In the context of online gaming, there are different types of cheating (Yan and Randell, 2005): modifying the game client program, data, or both; exploiting a bug or a loophole; or abusing of procedures to perform disloyal actions. Johan Huizinga defines cheating as the action of pretending to obey the rules of the game, while secretly subverting them to gain an advantage over an opponent. He captured the spirit of cheating as anti-play (Huizinga, 1971). Furthermore, evolutionary psychology (EP) observes human nature as the result of a universal set of evolved psychological adaptations to recurring problems in the ancestral environment. EP considers the hypothesis that people have a cheater-detection module (Cosmides, 1989). Cheating is a violation of a particular kind of restrictive rule that goes along with a social contract. Social exchange is a collaboration system for mutual benefit and cheaters violate the social contract that governs social exchange (Cosmides and Tooby, 2005). The selection pressure for a dedicated cheat detection module is the presence of cheaters in the social world.

There are several approaches for cheating detection in games, but they are not focused on second screen gaming. Usually, those solutions are focused on online multiplayer environments, such as the one proposed by (Laurens et al., 2007), which monitors the player behaviour for indications of cheating play, and the one presented in (Yeung et al., 2006), which proposes a scalable method to detect whether a player is cheating, or not, based on the dynamic Bayesian networks. Alayed et al.

present another behavioural-based cheating detection in online first-person shooters games using machine learning techniques (Alayed et al., 2013), while (Ferretti and Roccetti, 2006) focused on an algorithm based on the monitoring of network latencies.

Sync2FairPlay overview

Sync2FairPlay is a module that can be integrated into any second screen gaming app, giving users the control of the synchronisation between the app and the TV broadcast. It provides a universal synchronisation mechanism that helps users to achieve a pleasant second screen experience, while automatically dealing with cheating to provide a fair gaming competition to all players. Sync2FairPlay is comprised of three main components (see Figure 5.13):

- SMUF - The interaction mechanism that relies on the users' feedback on how apps' key events are synchronised with the corresponding TV broadcasts' key events. SMUF provides a universal and simple interaction control to the users, who can adjust their second screen experience regarding how they perceive the TV delay. This component appears in the higher layer of the app implementation, integrated into the interface with the user.
- CDM (Cheater-Detection Module) - We are using P²MUCA, following the X-Users model, for the adaptation in order to build a profile for each user according to his gaming and syncing behaviours, based on interaction stream data, such as the current app delay, and the total of key events responded. CDM allows the system to detect users with an "abnormal pattern", which may be classified as cheaters.
- CPM (Cheating-Penalisation Module) - This module deals with the users who are detected as cheaters by the CDM, returning the users' cheater status, according to their previous behaviours. The module can also be used directly by the game mechanics of the second screen app since its output is a cheater status (counter) that will be used to set the users' penalisation. In the context of this work, this status is used to define the level of limitation on the users' interaction with SMUF.

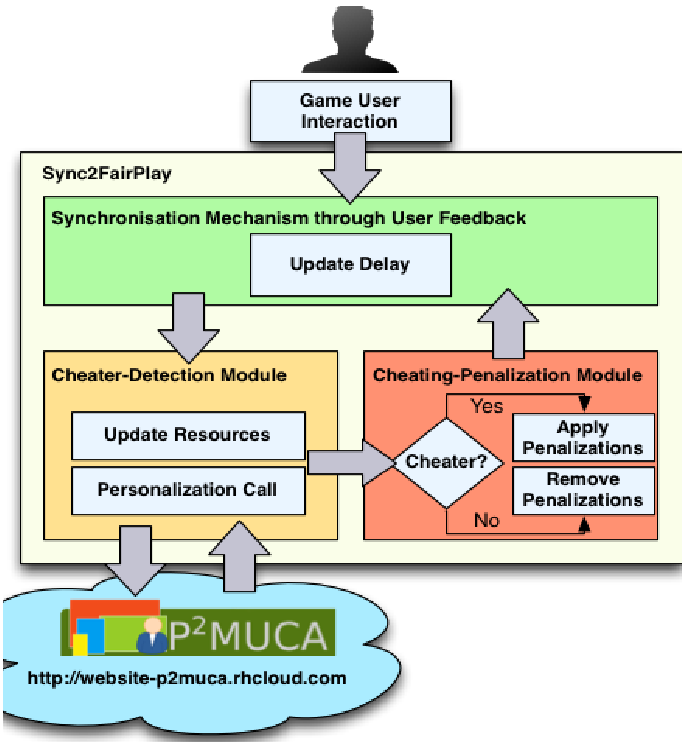


Figure 5.13 - Sync2FairPlay's architecture (including the algorithm's main actions).

Figure 5.13 summarises how Sync2FairPlay works within the context of an app that uses it. A user interacts with the app interface, either to play or to synchronise the app through SMUF. Next, the CDM's algorithm manages the interactions resources involved in the current user interaction (e.g., the new app delay, an answer to a quiz question, among others), updating them in P²MUCA. At specific moments, CDM calls P²MUCA to request the personalisation option that corresponds to the profile of the user. The personalisation option will indicate if the user is considered a cheater. Then the result is sent to CPM, which acts accordingly, removing penalisations (if the user had any) in case of a negative result or updating the users' cheater status, when the user is detected as a cheater. Afterwards, the cheater status is used to define the user's penalisations (to limit the user interaction with SMUF).

Cheating-Penalisation Module

The CPM is responsible for taking action in case a user is detected as a possible cheater. We could simply prevent a user from playing the game, but it is necessary to take into account that we need to give the benefit of the doubt to users due to factors like luck, network latency, lack of information regarding the real TV delay, and possible imprecisions (false positives cases) of the CDM. Although not being in the field of gaming, the Instagram[11] app implements an interesting mechanism to detect when a user is supposedly trying to exploit the system, commenting or liking too fast and too much. When that happens, Instagram blocks the user temporarily (see Figure 5.14).

Therefore, we envisioned a penalisation mechanism with some points in common with Instagram's mechanism (see Figure 5.15). We set up three penalisation stages that are applied after the user is identified as a potential cheater (Figure 5.16 presents a summary of the CPM's flow to help to understand the stages):

- Penalisation Warning: a message appears on the second screen app if the user is identified as a potential cheater for the first time. The message states that the delay set by the user does not correspond to the real delay of the TV broadcast s/he is watching, and therefore, the system resets it back to 0 seconds. The message also states that the user can change the delay again as s/he wishes, but if the same behaviour is found again, s/he will be penalised. However, if the user thinks that this message is an error (i.e., a false positive), then s/he can ignore it. To dismiss this message, the user can click on a button labelled: "OK, I understand", which will take her/him to the SMUF screen.
- Temporary Penalisation: if the system identifies a user as a potential cheater and verifies that s/he was already detected as a cheater on the previous evaluation, then a temporary penalisation message will be displayed. In this case, it is stated that the user's delay still does not correspond to the current TV broadcast s/he is watching. Therefore, the delay resets to 0 seconds, and the user cannot change it until the personalisation option (her/his profile) switches back to non-cheater. The user's delay will maintain the value of 0 seconds each time the user is detected as a cheater.
- Permanent Penalisation: finally, if the system verifies that a user was identified as a cheater on four consecutive user evaluations, a message appears informing that the user's delay is 0 seconds and what are the remaining consequences (see Figure 5.15). It is a permanent penalisation under the current competition (can be a result of continuous cheating, e.g., user is listening to the radio, user has a lot of good luck with the predictions or knows a lot being a false positive), but it is temporary across competitions and apps, because, at the end of the competition, s/he may already have a different profile. However, all knowledge about the

[11] https://www.instagram.com

user interactions and her/his profile will be reused in other competitions, from the same app or from another (if it uses P²MUCA too), thus, s/he may be penalised from the start.

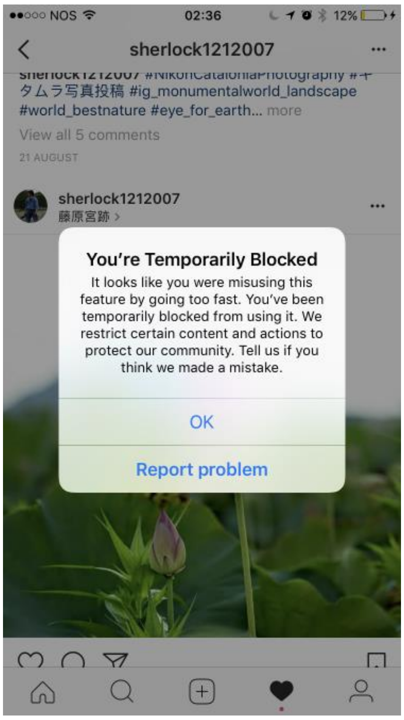


Figure 5.14 – Instagram app blocks a user temporarily.

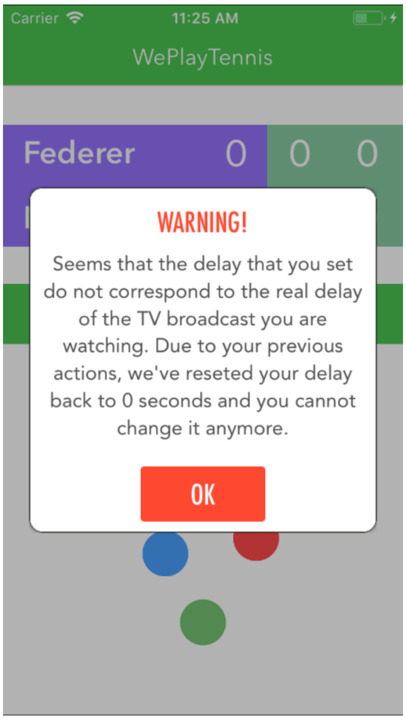


Figure 5.15 - Sync2FairPlay's CPM applies a permanent penalisation to a user on WePlay-Tennis.

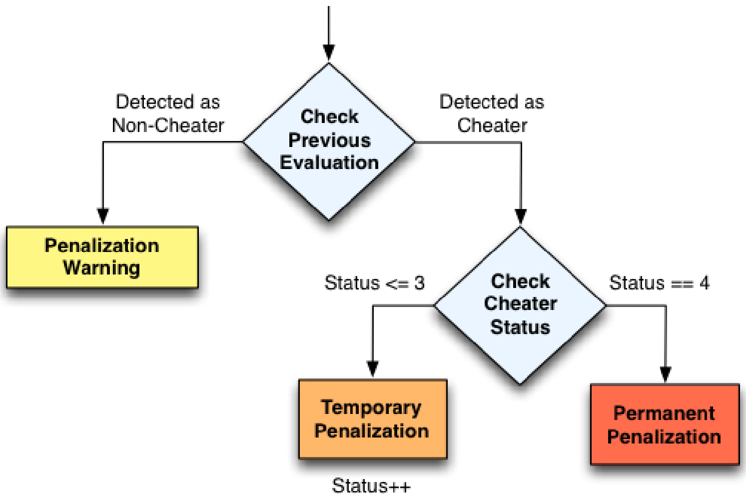


Figure 5.16 – WeSync/WePlayTennis's CPM: summary flow when a user is detected as a cheater.

The reason why we started by adding a penalisation warning was to acknowledge users that they could be doing something unexpected. We wanted to give them the possibility to set a new delay, or maintain the old one if they were being honest (it would mean a false positive was detected), and at the same time notify them to rethink their behaviour, since they could be penalised. With the temporary penalisation, we sought to punish users on the next key event by resetting their delay to 0 seconds, so they could not watch what would happen on the TV broadcast before receiving the key event on the app. The idea was to evaluate how users behave (do they still keep giving correct answers?) when a delay was not set as if they were on the venue where the event was taking place. Once again, to account for false positives, which can just be due to luck on a couple of key events, the system does not apply a harsh penalisation until a user receives four temporary penalisations. After this, a permanent penalisation is applied, setting the user's delay to 0 seconds and preventing her/him from changing it until the end of the broadcast.

It is important to note the existence of the Cheater Status counter, which determines when to apply the right penalisation. For game developers, this Status can be useful as an input for the game mechanics to: remove game points, limit the participation in the next events, and ban the user from the gaming experience, among others possibilities.

SMUF and WeSync's rationale and design

When using a mechanism like SMUF, it is the client application (and not the server) that delays the events from appearing on the second screen device, which allows the existence of a scalable solution, regardless of the number of users. Thus, we developed a mobile game called WeSync in order to evaluate how users interact with SMUF in a competitive scenario.

WeSync is a mobile app that prompts users to guess the outcome of corner kicks (Figure 5.17), penalty kicks and free kicks during a football match. Users can also check their predictions' outcomes, as well as their friends' scores. Furthermore, WeSync also notifies users when a goal is scored (allowing them to quickly share their thoughts on social networks), or when a half starts or ends. However, when these events are not synchronised with the TV broadcast, users need to synchronise them by using SMUF, which can be done by adjusting the SMUF's slider in a screen that appears right after each key event occurrence (see Figure 5.19 and Figure 5.20). Users rate their experience through SMUF, providing feedback on how the app is presenting the events. Each subsequent event is presented taking into account the previously provided feedback, in order to achieve the synchronisation between the app and the TV broadcast. Once users state that the app is synchronised, the screen with the slider stops from appearing right after each key event. A popup is shown explaining that from now on they still can adjust their experience whenever they wish by clicking on the top-right button that just appeared (see Figure 5.18).

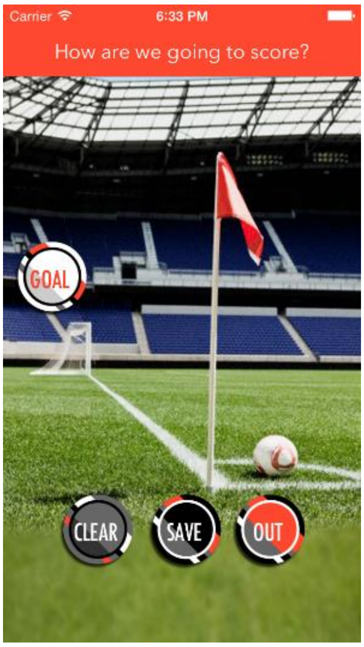


Figure 5.17 - WeSync UI: a key event that prompts users to predict a corner kick.

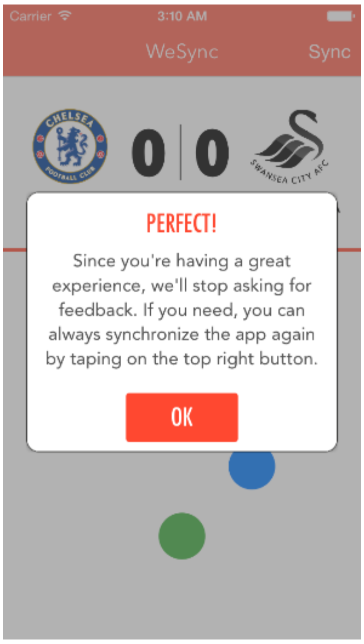


Figure 5.18 - WeSync UI: notification when the app is synchronised.

We introduced four interaction cues to facilitate the user interaction in SMUF context (see Figure 5.19 and Figure 5.20): the temporal indication (e.g. "app is 3 seconds ahead"); the illustrative animation above the slider which behaves accordingly to the delay set, the overall colour of the slider and the animation, and the experience rating (e.g. "Great!", "Good!", "Fair.", and "Poor…"). The colour intervals and the experience rating values were based on the work done by (Mekuria et al., 2012b).

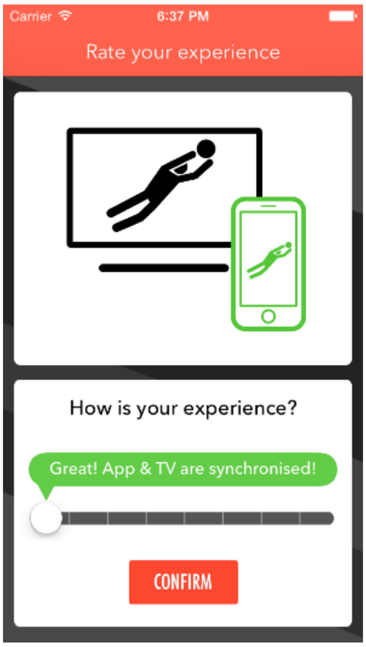


Figure 5.19 – WeSync's SMUF: initial screen with the default at position "Great!".

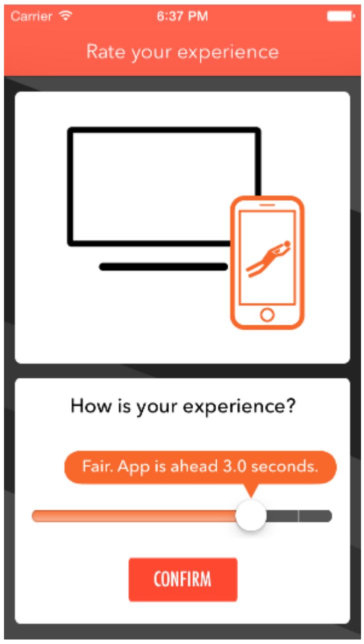


Figure 5.20 – WeSync's SMUF: user setting a delay of 3s.

Finally, we did not include any information regarding the live sports event, such as the match time, which could be used to compare the TV broadcast match time with the app match time to synchronise both feeds. We wanted to have a universal synchronisation mechanism that could be deployed on any broadcast, regardless of the information presented on the TV screen. More details about SMUF and how WeSync works can be found in (Centieiro, Madeira, et al., 2015).

The WePlayTennis Application

WePlayTennis is another second screen app of the same type of WeSync, but with the focus on tennis instead of football (see Figure 5.21). The WePlayTennis prototype prompts users to predict on the outcome of tennis rallies[12] (if it is a winner or an error from one of the tennis players) and serves (if it is an ace, double fault, winning serve, or return ace). The UI with the indication of a prediction's result can be seen in Figure 5.22, which indicates, in this case, that the user had success in her/his prediction.

WePlayTennis follows the same concept of WeSync, also using the Sync2FairPlay component and implementing the same game mechanics. However, we made a revision of the Sync2FairPlay's CDM and CPM modules in order to deal better with the potential dishonest players in the scope of WePlayTennis. Despite the good results obtained with WeSync, we made a slight adjustment to the algorithm of CDM, requesting the personalisation option to P²MUCA more often to get even less false positives. Previously, in WeSync's implementation, CDM would only request the personalisation option after a prediction event. With the change, the CDM started requesting the personalisation

[12] A tennis rally starts with the serve and the return of the serve, followed by continuous return shots until a point is scored, which ends the rally.

whenever the interaction resources are updated, whatever they are. This was an important and decisive change to detect presumed cheaters just before they could exploit the system. Given that, the algorithm behind the penalisation module was adjusted to prevent players from achieving a good score before a permanent penalisation could be applied. We made some more minor adjustments to the algorithm, which are not relevant at this moment.

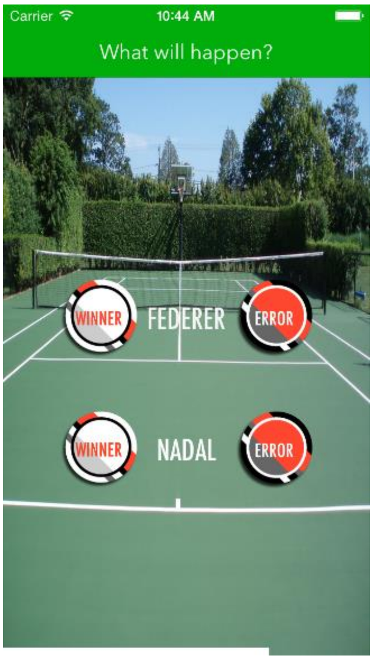


Figure 5.21 - WePlayTennis UI: A key event that prompts users to predict a point.

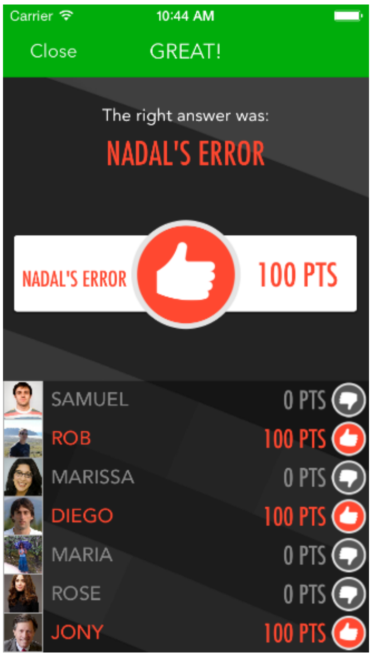


Figure 5.22 - WePlayTennis UI: The indication of the prediction's result.

5.2.2. Using Personalisation in the Cheater-Detection Module

The use of personalisation in the prototypes went through the implementation of CDM in order to cope with potential dishonest users (cheaters) as they could freely control the app delay with SMUF. The CDM component was implemented around the personalisation definition based on X-Users and its algorithm includes several calls to P²MUCA. Essentially, this module for cheaters detection corresponds to the full use of our personalisation solution in the scope of these second screen apps.

In the current context, app delays placed by a user need to be personalised (automatically adapted) to benefit the user experience and its social exchange as a whole, in a peer-based context, persuading and preventing users from cheating while exploiting the system. We wanted to adapt automatically the app delay set by users (TV viewers) according to their behaviours in the second screen gaming experience. The personalisation defined here might be used by any app of the type of WeSync and WePlayTennis, provided it is registered on P²MUCA. First, we had to select a set of resources (interaction stream variables) that would make sense in the context of WeSync/ WePlayTennis. They would have to be resources originated by the main interactions of the users with the system, which can be found in Table 5.7 (the XUPIL script defining the entire personalisation is in appendix A.3).

Therefore, we selected data coming from the interactions with SMUF and key events. Initially, we decided to define only one parameter (formerly called `TVViewerProfile`) based on the selected resources in order to represent only two personas (`cheater` and `non-cheater`). However, we found out that the separation of the resources could work better since some of them would be more related with a kind of a synchronising profile while the others would be more related with a gaming profile. The new combinations of the resources produced two parameters, giving us a greater granularity to work on. It means that, at each key event, the user is represented by two values, corresponding to the two parameters: `GamerProfile`, which is defined by expression (21); and `SyncerProfile`, which is expressed by (22).

R	Resource	Description
Interaction stream resources		
1	keyEvents	[increment] Total of key events (questions) answered/predicted.
2	totalResponseTime	[sum] Sum of the response time used for all answers/predictions.
3	totalCorrectResults	[sum] Total of correct answers/predictions.
4	currentUserDelay	[replace] Current value of the app delay after last key event, placed by user.
5	totalDeltaUserDelay	[replace] Sum differences between successive app delays at each key event.
6	totalUserDelay	[sum] Sum of the app delays at each key event.

Table 5.7 – WeSync/WePlayTennis' resources (interaction stream data used for personalisation).

$$0.3 * \left(\frac{1}{\left(\frac{totalResponseTime}{keyEvents}\right)}\right) + 0.7 * \left(\frac{totalCorrectResults}{keyEvents}\right) \tag{21}$$

$$0.5 * currentUserDelay + 0.25 * \left(\frac{totalDeltaUserDelay}{keyEvents}\right) + 0.25 * \left(\frac{totalUserDelay}{keyEvents}\right) \tag{22}$$

These expressions were determined after several iterations of simulations based on empirical data and domain knowledge. Furthermore, the weights were slightly refined with the user experiments study (in section 5.2.3) and the results confirmed the expressions are appropriate for our goal. We were mainly interested in discovering if at each key event (moment) a user should be marked as a cheater, or not.

This way, we defined the same two options (clusters) for each parameter (profile): `High` and `Low` (they revealed themselves as being adequate as demonstrated by the results in section 5.2.3). In both prototypes, we considered that a user should only be marked as a potential cheater (final profile) when both parameters (sub-profiles) result in high values. The combinations of the parameters' options result in the four personalisation options (Table 5.8), corresponding to the personas considered in our scenario: (1) `Non-Cheater Tier-1`; (2) `Non-Cheater Tier-2`; (3) `Non-Cheater Tier-3`; and (4) `Cheater`. The different sub-profiles (levels) of non-cheater users gave us the necessary granularity to observe better the behaviour of the users. However, these three tiers of non-cheater can be treated in the same way by an app if it does not need to distinguish them to respond differently to each one.

		GamerProfile	
		Low	High
SyncerProfile	Low	Non-Cheater Tier-1	Non-Cheater Tier-2
	High	Non-Cheater Tier-3	Cheater

Table 5.8 – WeSync/WePlayTennis: Combinations of the two parameters options result in four personalisation options.

5.2.3. User Study with End Users

We carried out three phases of user tests with the prototypes in order to evaluate not only the users' feedback regarding the different parts of the proposed system, but also to obtain user interactions data to design the best CDM solution and to evaluate the gained knowledge about the users across the two different applications.

Design and participants

The WeSync prototype was used in the first two phases of tests and WePlayTennis was used in the third phase. For the tests, WeSync was based on a simulation of a football match broadcast displayed on a TV, where a video with several highlights was presented (it is possible to have several videos in order to simulate different matches). In order to further simulate a live environment, a random TV delay (that ranged between 0-6s) was set for each participant and video, according to the most common delay values registered by (Kooij et al., 2014). Finally, WeSync presented information about the match (result and teams) and the user's score, while waiting for key events to occur. On the other hand, for WePlayTennis, we simulated a tennis match broadcast displayed on a TV which was a video with several highlights of a Wimbledon final between two top players (Rafael Nadal vs Roger Federer) easily recognised by people who do not follow the tennis circuit on a regular basis. Its mechanics and usage procedure were similar to the WeSync's; thus, participants that were accustomed to using WeSync app already knew how to handle this new second screen app within a competitive scenario in a sports context.

After each phase's tests session, we asked participants to rate statements (with a five-point Likert-type scale) in a questionnaire regarding their experience with SMUF and their cheating perception. Users were also free to write down any further comments and, in the end, we conducted a brief interview. Additionally, during each session, we also registered all the users' interactions data for the resources described in Table 5.7. Regarding the three phases of tests, they were particularly designed as follows.

Phase 1:

We aimed to assess SMUF's usability and usefulness, and to collect data to create and adjust the different personalisation profiles. The user tests were conducted with 15 voluntary participants (11 male and four female) aged 23 to 45 ($\bar{x}$ = 31.5; σ = 7.4). The tests took place in a lab at our University campus and included two test sessions, with the participants being briefed before each one. In each session, participants watched an 8-minute highlight video from a football match, containing six key events to predict what the outcome would be. Each participant had two initial moments to become familiar with SMUF, before the key events started appearing, allowing her/him to understand better how to set a delay.

The questionnaire used in this phase had six main statements, besides initial questions to characterise the user, being presented in Table 5.9.

Phase 2:

In this phase, we wanted to evaluate the system as a whole, studying how participants interact with it and collecting data to analyse how well CDM worked to ensure a cheating-free gaming experience when SMUF is used.

This session of user tests was conducted with 30 voluntary participants (21 male and nine female) aged between 18 and 36 ($\bar{x}$ = 20.67; σ = 3.6). The tests took place in the lobby area of the library in our University campus. We chose this area in order to have a broader number of participants from different areas and with diverse profiles, which although may or may not be familiar with games or the second screen concept, are the most suitable target for mobile apps (Nielsen Holdings, 2014). Before each test session, participants were briefed about the test, noting that this was a game competition with a digital prize to be awarded by the end of the tests, in order to encourage participants to compete using all possible means. The tests were conducted individually for each participant.

Statements
Synchronisation mechanism
S1. I had a good synchronisation experience.
S2. It is easy to use the interaction mechanism.
S3. This mechanism is useful to synchronise the app with the TV broadcast.
Competition and cheating perception
S4. This mechanism allows for unfair competition.
S5. It is easy to cheat while using the synchronisation mechanism.
S6. The synchronisation mechanism motivates me to be honest while using it.

Table 5.9 – Study EU.2: Statements of phase 1's questionnaire.

Each participant watched a 6-minute highlight video from a soccer match that took place in 2012. While some participants may have remembered the final score of the match, none of them could recall the plays of the highlight video. During the video, the participants were prompted to predict the outcome of 12 match's key events, between corner kicks and free kicks, and they received one notification at the start of the match. This notification and the first two predictions were used as learning checkpoints so participants could understand how to interact with SMUF and how the predictions unfolded.

Users had seven seconds to answer to each prediction request (a decreasing bar indicated the remaining time users had to answer), which was enough time to visualise what was happening on the field and make a prediction. Each correct prediction was awarded 300 points (0 points for wrong predictions), which means that the highest possible score was 3000 points (the first two predictions did not count towards the score). A scoreboard was added next to the TV, so users could quickly check how they were standing comparing to the previous players (Figure 5.23). This procedure was important to remember them about the competition in which they were inserted, to put some pressure on them so they could come up with new strategies, which would be cheating, if possible.

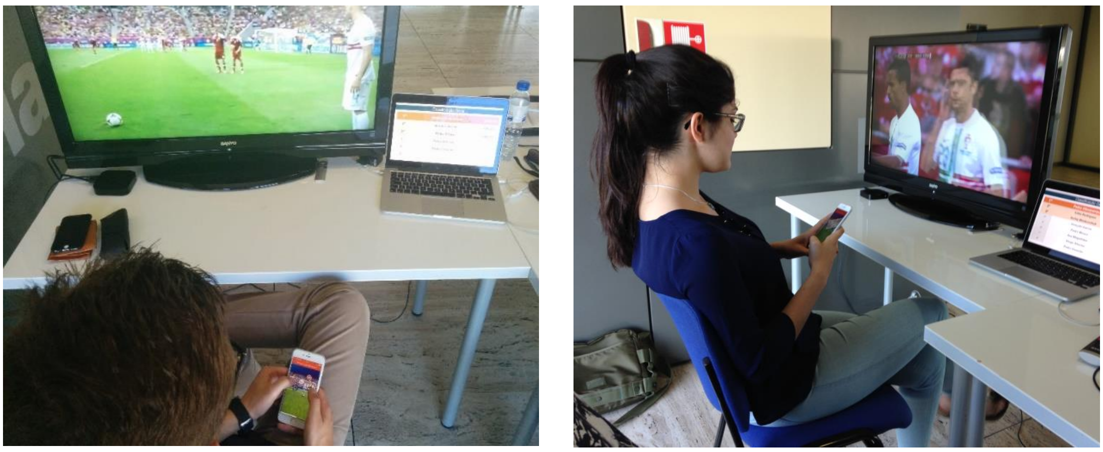


Figure 5.23 – Study EU.2: Two participants interacting with WeSync (scoreboard near the TV).

Regarding the questionnaire, the first part of the questionnaire gathered users' personal data, such as, age, gender, and their familiarity with second screen apps, which was not included in phase 1's questionnaire. Then, the first six statements were exactly the ones used in phase 1's questionnaire. The remaining four statements (again, using a five-point Likert-type scale) addressed the detection and penalisation modules (Table 5.10).

Statements
S7. I think I acted as a cheater by setting a high delay to have an advantage in the competition.
If you were detected as a cheater…
S8.1. I think the detection mechanism acted properly.
S8.2. I think the penalisation mechanism was adequate.
S8.3. I think I would not cheat if I knew that there was a cheating detection mechanism beforehand.

Table 5.10 – Study EU.2: Additional statements in phase 2's questionnaire.

Phase 3:

After analysing the results of the previous phase, we concluded that the parameters used in the personalisation were appropriate to reach the intended goal. Therefore, we maintained the definition of the personalisation instance with the four personalisation options based on the same two parameters. As we did not need to change anything regarding the definition of the WeSync's personalisation, the CDM in WePlayTennis used a simplified XUPIL script (see appendix A.4) that only needed to refer WeSync's personalisation. As previously stated, WePlayTennis was implemented with a slightly different algorithm for CDM and CPM in order to improve the treatment of users marked as cheaters.

In the context of the current research work, the most important aspect to evaluate in the third phase was the knowledge about the users' behaviours that WePlayTennis could have from the beginning due to the use of P²MUCA, taking advantage of the users' interactions with WeSync. We intended to assess if that knowledge gathered with WeSync could be successfully used to cope with users when these adapt the delays in WePlayTennis. It was also important to help surpass the cold start issue originated when users start using the app for the first time. Therefore, we needed to include participants from the previous phase in the new tests with WePlayTennis. This third phase of user tests was conducted with 29 participants (see Table 5.11 for details). Nineteen of them already participated in phase 2's tests, and the remaining 10 participants deal with the whole concept for the first time.

Participants	Total	Male	Female	Age min	Age max	$\bar{x}$	σ
All	29	20	9	18	41	24.5	7.12
Group 1 [G1] (already in phase 2)	19	13	6	18	36	21.4	4.18
Group 2 [G2] (only in phase 3)	10	7	3	19	41	30.3	7.85

Table 5.11 – Study EU.2: Characterisation of the phase 3's participants.

We tracked the participants across the two phases of tests through the creation of several P²MUCA accounts. In the second phase, we gave account credentials to each participant who had to keep them in case s/he would participate in the next phase of tests. The sign-in system of both applications was based on the P²MUCA sign-in mechanism, so participants were signing in with both the application in use and P²MUCA. The new participants in phase 3 have received the new credentials. Another solution would be based on the following flow of actions for each user:

1) WeSync would register the participant within P²MUCA, with her/his authorisation;

2) WeSync would store the P²MUCA credentials and show them in the user account's area;
3) The user would use them with another application that connects to P²MUCA, which would be WePlayTennis in our study.

The tests with part of the participants took place in the same lab used for phase 1's tests, but they also occurred in other places according to the availability of participants since we wanted to reach a more differentiated range of people. Before each test session, participants were briefed about the test, noting that this was a game competition with the digital prize to be awarded by the end of the tests. Once again, we wanted to encourage participants to compete using all possible means. The tests were conducted individually with each participant.

Each participant watched a highlight video of approximately 8 minutes from a tennis match that took place in the 2008 Wimbledon Championships. It was the men's singles final between Rafael Nadal and Roger Federer. None of the participants remembered the result of the match; not even recalled the tennis points included in the highlight video. During the test, the participants were prompted to predict the outcome of 12 situations of play (between running points and serves) presented as the video highlights. They received one notification at the beginning of the match. This notification and the first two predictions were used as learning checkpoints so participants could understand how to interact with SMUF and how the predictions unfolded. The remaining configuration of the tests followed the same procedure applied in phase 2.

Regarding the questionnaire, once again, the first part of the questionnaire gathered user's personal data, such as, age, gender, familiarity with second screen apps, but also familiarity with tennis and its rules, which was not included in phase 2's questionnaire. Then, we maintained the same ten statements that were used in phase 2's questionnaire. We included the last statement to collect the perception of the participant regarding the previous knowledge that the application might have about her/him, if s/he felt the application already knew her/him. A final question was included to know if s/he was already marked as a cheater when s/he started using the app (Table 5.12).

Statement
S9. I think the system recognised me correctly after the previous use of the WeSync app.
Final Question
Q1. Were you identified as a cheater when you started using the WePlayTennis app?

Table 5.12 - Study EU.2: Additional statement and question in phase 3's questionnaire.

Results and discussion of phase 1 (user experiments and data collection)

We carried out user experiments with the WeSync app in order to evaluate the SMUF's usefulness and viability in a gaming scenario and, more important, to collect essential interactions data to build the gaming and synchronising profiles. To this research, the first three statements regarding SMUF were important in order to ascertain that SMUF is well understood and accepted by the participants. This is an important condition for us since this way it supports the study regarding the most important aspect of this research, which is the use of personalisation through the CDM component.

Therefore, regarding the first part of the questionnaire, its results were very positive concerning the overall participants' experience with SMUF. Figure 5.24 shows that, in general, participants had a good synchronisation experience ($\bar{x}$ = 4.1; σ = 1.13) and found SMUF easy to use ($\bar{x}$ = 4.0; σ = 1.07). Moreover, they agreed that SMUF is useful as a mechanism to synchronise the app with the TV broadcast ($\bar{x}$ = 4.6; σ = 0.83).

The remainder of the questionnaire also showed very interesting results. Although the majority of participants have agreed with S1 (6.67% agreed, and 46.67% strongly agreed) that SMUF allows users to have an unfair competition ($\bar{x}$ = 3.8; σ = 1.32), there are a considerable number of neutral opinions and two of them even stated "the competition would not be unfair since everyone has the same conditions to cheat". More important and, maybe, definitive is that according to S2's results participants really found it easy to cheat using SMUF ($\bar{x}$ = 4.4; σ = 0.83). Generally, they felt it could be done after a few key events, which indicates the necessity to include a module to cope with cheating, automatically adapting the app to the users' behaviours. Nonetheless, S3's results showed participants felt motivated by SMUF to be honest ($\bar{x}$ = 3.6; σ = 1.18). We believe this happens because of social and peer-pressure. Moreover, it is interesting to refer that one participant told us s/he would be dishonest if s/he would perceive that others were cheating.

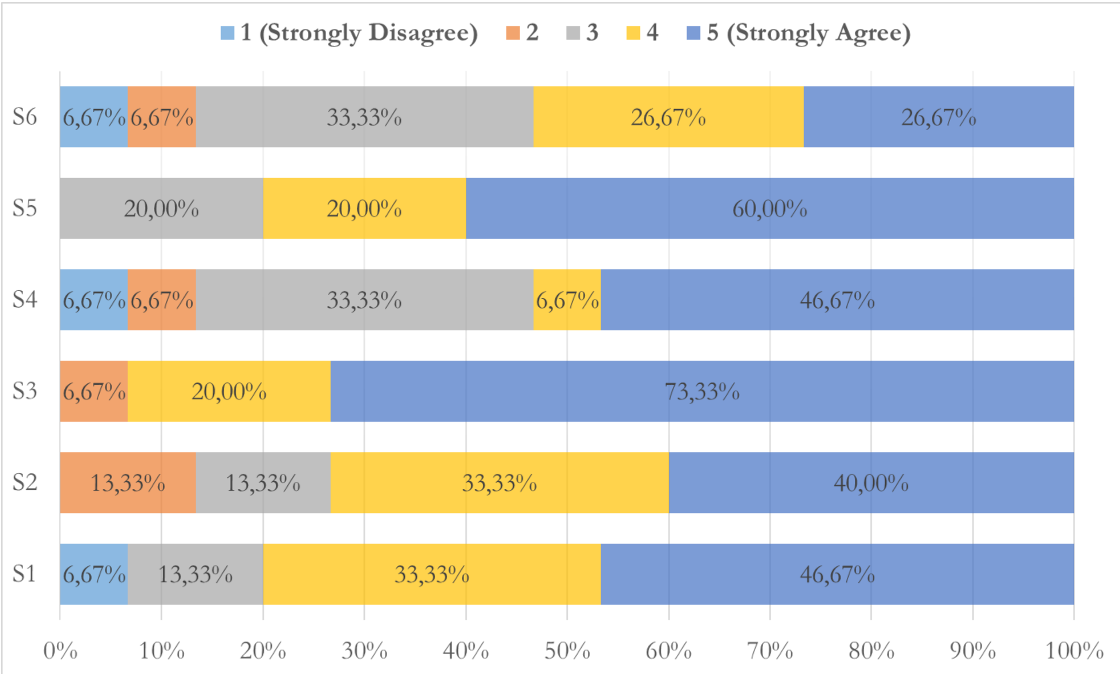


Figure 5.24 - Study EU.2: Summary of the phase 1's questionnaire results regarding: the SMUF experience (S1-S3) and the cheating perception (S4-S6).

Furthermore, after the first session, we asked participants to indicate (using a five-point Likert-type scale) how they behaved in terms of honesty when setting up the app delay for each key event. This was important to link gathered interactions data to users' perception on cheating. For example, no participants stated that they behaved like cheaters in this session and our results actually show no cheaters. Moreover, the results show that participants 1, 2 and 15 had a behaviour progression almost into cheater profile during session 1 and they confirmed that to us. Table 5.13 shows data of only six participants since these present the most interesting values to understand the global results. Data are relative to the moment right after the conclusion of all six key events.

In the second session, we asked each participant to set up an app delay for each key event according to a "behaviour guide" with indications (using a 5-point Likert-type scale: 1-"100% dishonest" and 5-"100% honest") on how we would like them to behave. This way, we would gather interactions data that would correspond to participants behaving according to the four profile options. As we have indicated, only four of them behaved like clear cheaters. It was easy for participant 15 as s/he almost did it in session 1. With the two sessions, we got data corresponding to 30 users instead of only 15, which was better for testing purposes. The final profiles confirmed the parameters' expressions previously presented. The final expressions were obtained after several simulations with the collected data in which we were refining them in terms of the weight given to each resource.

P#-S#	TV Delay	Final Profile	GamerP. (exp. 18)	SyncerP. (exp. 19)	R2/R1	R3/R1	R5/R1	R6/R1	R4
1-S1	2,4	non-cheater tier-2	0,59	4,31	4,56	0,67	0,75	2,00	4,5
1-S2	0,2	non-cheater tier-1	0,21	3,17	3,84	0,17	0,17	1,75	2,0
2-S1	1,6	non-cheater tier-3	0,34	18,17	4,24	0,33	4,00	7,33	24,0
2-S2	4,5	non-cheater tier-1	0,52	5,92	5,95	0,33	0,67	2,67	6,0
11-S1	5,9	non-cheater tier-1	0,30	7,69	2,22	0,33	0,08	4,25	4,5
11-S2	3,3	**cheater**	0,53	16,88	2,54	0,67	1,67	8,92	12,0
12-S1	4,0	non-cheater tier-1	0,19	4,10	2,75	0,17	0,08	2,25	2,5
12-S2	4,4	**cheater**	0,67	14,50	3,42	0,83	2,00	7,33	12,0
14-S1	3,9	non-cheater tier-1	0,22	7,69	4,07	0,17	0,08	4,25	4,5
14-S2	5,7	**cheater**	0,71	26,13	4,60	0,83	1,58	12,83	21,5
15-S1	5,1	non-cheater tier-3	0,24	12,56	4,50	0,17	0,92	6,50	9,5
15-S2	0,8	**cheater**	0,56	12,73	3,79	0,67	0,92	6,83	8,5

Table 5.13 – Study EU.2: Data results of 6 key-participants: final profiles, parameters and resources values.

Results and discussion of phase 2 (using an adaptive WeSync)

Participants made 360 predictions, and accessed SMUF 140 times ($\bar{x}$ = 4.66 per test session) during the whole evaluation process. As stated before, after each prediction's outcome, a personalisation option (profile) was requested from P^{2}MUCA and, during the tests, the system classified the participants according to Table 5.14.

Profile (personalisation option)	Result
Non-cheater tier-1	22%
Non-cheater tier-2	44%
Non-cheater tier-3	17%
Cheater	17%

Table 5.14 – Study EU.2: classification of participants during the phase 2's tests.

From the participants classified as cheaters, 72% were really cheating at the time (participants' delays were higher than the TV delay and the answers were always correct), but 28% were not (false positives). However, not even all false positives had consequences beyond the penalisation warning (only 23% had a temporary penalisation with cheater status = 1), and all of them rolled back to a non-cheater profile after it. Finally, it is also important to note that all participants who tried to cheat (9 participants, 33%) were detected, and in some cases, if the video had a longer length, they would not be able to change the delay anymore.

Table 5.15 presents data that allowed us to correlate final scores with users profiles among the top 5 participants. It is clear that the penalisation system we implemented was bland. It allowed that 3 participants with "cheating behaviours" could make it to the top. One of them even got the best score, together with a non-cheater participant. However, it was what we wanted since this way participants had the necessary leeway to decide what to do after being marked as a cheater. We needed to collect data, so we did not prevent participants from achieving a good score before a permanent penalisation could be applied.

We also wanted to observe how some participants would manage to get a break into the system, defining strategies in order to exploit it. Thus, for instance, we noted that participant 0 was alternating between non-cheater and cheater (status 1) for some time, winning points with that strategy and before being caught by the system. However, we can see that the cheater status of participants 0 and 25 was already equal to 4, which means they ended up with a permanent penalisation. Participant 20 was near it, having a status equal to 3 and if the test had continued s/he would end up with the same punishment. In order to address these cases, game developers can penalise users by resorting to the game mechanics (e.g., by deducting points).

There are two other cases, which are not present in Table 5.15, that are worth mentioning. A participant with 600 points was penalised early on for being a cheater. Then s/he tried to increase her/his delay to see how far s/he could go until s/he was penalised again. Another one with 900 points changed his behaviour, by decreasing the delay initially set, after being detected as a cheater.

Participant #	**Score (pts)**	**Final Profile**	**Gamer Profile [H = 0.58, L = 0.13]**	**Syncer Profile [H = 3.33, L = 0.0]**	**Real Cheater**
27	1800	Non-cheater tier-2	High (0,56)	Low (1,49)	No
20	1800	Cheater (Status = 3)	High (0.43)	High (3.2)	Yes
0	1500	Cheater (Status = 4)	High (0.46)	High (2.16)	Yes
28	1500	Non-cheater tier-2	High (0,58)	Low (1,12)	No
25	1500	Cheater (Status = 4)	High (0.34)	High (2.25)	Yes

Table 5.15 – Study EU.2: phase 2's top 5 participants - scoring and profiles (highest and lowest values indicated in the top row) at the end of the competition.

Regarding the questionnaire, results revealed that participants did not use second screen apps to interact with TV shows on a regular basis. The majority of participants (46.67%) stated they never use second screen apps, while 26.67% rarely use them. 20% interact with second screen apps in a regular basis, and only 6.67% use it frequently. Nevertheless, the briefing included an explanation of what are second screen apps, and the problem of having apps not synchronised with the TV broadcast. Participants seemed to have understood the concept.

As for the synchronisation mechanism, 53.3% of the participants agreed with statement S1 and 23.3% strongly agreed with it. When we asked if SMUF was easy to use (S2), the response was very positive since the majority of participants strongly agreed with the statement (46.67%). Participants understood the issue that SMUF tried to solve, as they rated S3 very positively: 53.3% agreed and 40% strongly agreed with it. These results (Figure 5.25) corroborate the ones from phase 1, where other participants also rated the first three sentences very positively.

About the cheating perception statements, most of the participants stated that SMUF allows for unfair competition (40% agreed, and 20% strongly agreed). However, 20% of participants were undecided regarding S4. Some of them stated at the final interview that they only realised that SMUF allowed for an unfair competition after playing WeSync. Similarly, participants rated S5 positively (56.7% agreed, and 16.7% strongly agreed), meaning they thought it is easy to cheat while using SMUF. However, the results from both statements S4 and S5 (($\bar{x}$ = 3.6, σ = 1.10) and ($\bar{x}$ = 3.8, σ = 0.89), respectively) revealed an important decrease in relation to the same statements in phase 1 ($\bar{x}$ =

3.8, σ = 0.57) and ($\bar{x}$ = 4.4, σ = 0.82), respectively), showing that in the presence of the detection and penalisation modules, participants do not find so easy to exploit the synchronisation mechanism. When we asked participants if the synchronisation mechanism motivated them to be honest while using it (S6), results ($\bar{x}$ = 3.6, σ = 1.19) were very similar to the ones found in phase 1 ($\bar{x}$ = 3.6, σ = 1.18). Once again, we believe these values may result of social peer-pressure from playing with friends, being observed, or being seen as a dishonest person playing WeSync.

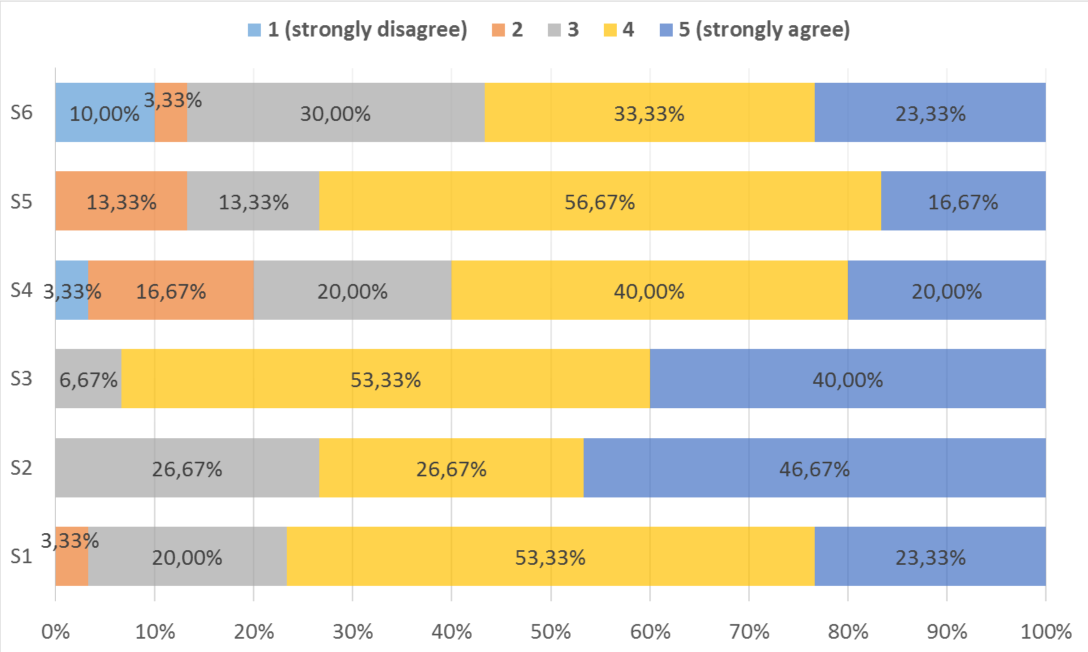


Figure 5.25 – Study EU.2: Summary of the phase 2's questionnaire results regarding: the SMUF experience (S1-S3) and participants' cheating perception (S4-S6).

Results from S7 were not surprising, as we knew beforehand from the analysis of the data logs that only 33% of participants tried to cheat. Therefore, the results match those data from logs, showing that 60% of participants disagreed with S7 (i.e., the majority of them did not cheat), while 16.7% were undecided about it (see Figure 5.26).

Lastly, participants who were detected as cheaters (false positives included), were asked to rate three additional statements: S8.1, S8.2, and S8.3. Regarding S8.1, results were clearly positive as all participants agreed or strongly agreed with it (50% each). This means that the participants understood that the CDM was working properly, even for those corresponding to false positives. Next, we evaluated if the penalisation mechanism was adequate and the results were equally positive (41.7% strongly agreed and 33.3% agreed). Finally, Figure 5.26 also illustrates how participants present great differences regarding statement S8.3, and thus, we cannot draw conclusions from those dispersed, but interesting, results. Maybe participants would need more time to interact with the app in order to really feel the CPM working and understand it to take any conclusions.

Furthermore, as for the suggestions and comments made by participants, some of them stated they noticed they could exploit the system only after playing WeSync, or after synchronising it with the football match. It opens a question; how are they going to behave when they start using a similar application for the first time? These participants are natural contenders for the third phase of tests. Some participants said they liked the experience and they thought that second screen apps like WeSync would improve the way people watch TV broadcasts. Lastly, one participant alerted to the situation of being possible to use other devices (e.g., radio) to know what will happen next on some TV shows like soccer matches transmissions. We acknowledge this particular issue, but we suppose that the CDM module might detect a player as a cheater even in these cases if we slightly adjust the created parameters.

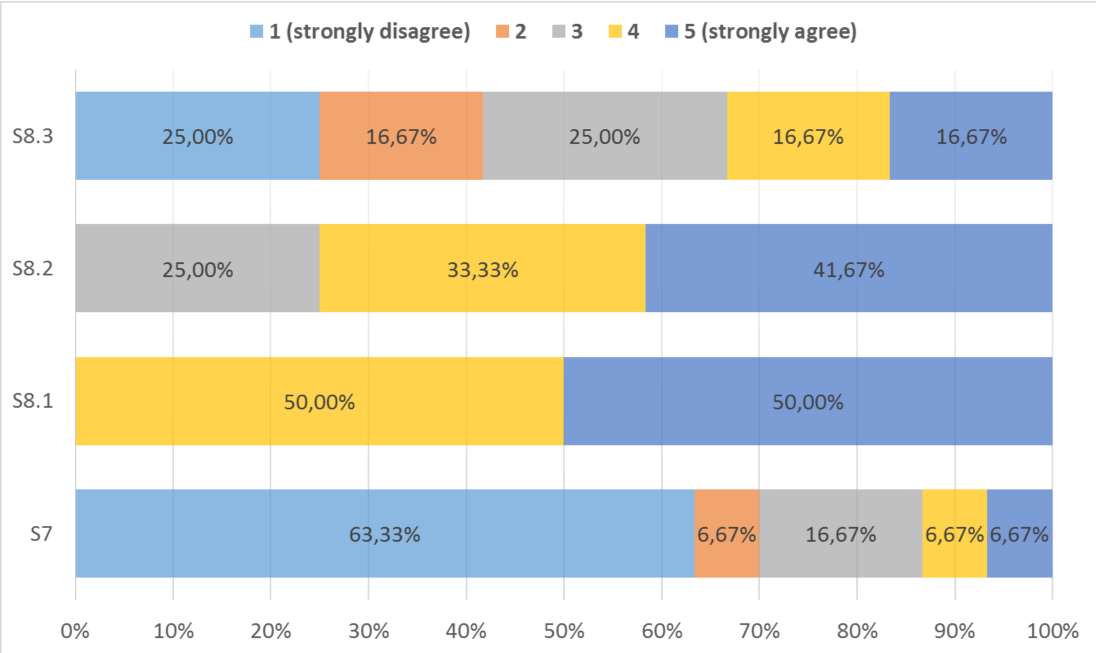


Figure 5.26 - Study EU.2: Summary of phase 2's questionnaire results regarding the CDM and CPM components (S7-S8.3).

Results and discussion of phase 3 (using an adaptive WePlayTennis)

Participants made 348 (29 participants x 12) predictions and accessed SMUF 121 times ($\bar{x}$ = 4.17 per test session) during the whole evaluation process of this third phase of tests. As stated before, after each prediction's outcome and SMUF's interaction, a personalisation option (profile) was requested from P²MUCA. Throughout the tests, WePlayTennis got different responses according to Table 5.16, in which we can see the distribution of the results by the four possible profiles.

Profile (personalisation option)	Response rate (Phase 3)	Response rate (Phase 2)
Non-cheater tier-1	35%	22%
Non-cheater tier-2	21%	44%
Non-cheater tier-3	33%	17%
Cheater	11%	17%

Table 5.16 - Study EU.2: classification of participants during the tests of phase 3.

Regarding the responses to personalisation calls classified as cheaters, 87% of the participants were really cheating at a given time (participants' delays were higher than the TV delay and the answers were always correct), but the remaining 13% were not (false positives). All false positives less one had only a penalisation warning. The other one had a temporary penalisation since the cheater status was equal to one. However, all of them rolled back to a non-cheater profile after it, which reveals the system was working better than previously with WeSync. These results are overall more positive than the ones from phase 2. The cheater-detection module was working better due to the changes made to its algorithm and the gathered knowledge with WeSync. It is important to note that P²MUCA already had enough data from the interactions of users with WeSync. Finally, it is also important to note that all participants who tried to cheat at a given time (10 participants (34%), 7 from G1 and 3 from G2) were detected, and the penalisation system implemented in CPM was less bland. It not allowed participants with "cheating behaviours" to make it to the top. They were not able to exploit the system in order to reach good results since they were caught faster than in phase 2, getting a permanent penalisation.

As for the results obtained with the questionnaire, similarly to phase 2, the overall results revealed that participants did not use second screen apps to interact with TV shows on a regular basis. The majority of participants (62.0%) stated they never use second screen apps (half of them), or rarely use them (the other half). 20.7% of them occasionally interacts with second screen apps, 13.8% uses them frequently, and only 3.5% uses second screen apps very frequently. In spite of these results, in comparison with the previous phase, we had higher scores with ($\bar{x}$ = 2.3, σ = 1.16) against ($\bar{x}$ = 1.9, σ = 0.97) in phase 1. In particular, participants of G1 (already in phase 2) are primarily responsible since they were the ones who responded with the highest scores, with ($\bar{x}$ = 2.5, σ = 1.17) against ($\bar{x}$ = 1.9, σ = 1.10) of G2. We assume it is because some of them felt that the WeSync experience increased a bit their overall experience with second screen apps.

Nevertheless, once again, the briefing included an explanation about what are second screen apps and the problem of having apps not synchronised with the TV broadcast. Participants seemed to have understood the concept. The briefing also included an explanation of how tennis is played since the participants' familiarity with the sport and its rules was globally very low ($\bar{x}$ = 2.3, σ = 1.26). Nonetheless, some participants (7.0%) had a deep knowledge about tennis since they practiced it in a regular basis. Others (31.0%) knew something about the sport because they practiced it from time to time, or liked to watch matches broadcasted on TV. The remaining (62.0%) had little or no knowledge about tennis and, above all, its rules, particularly, how points are played and obtained. We may suppose this lack of knowledge is an important reason for the personalisation results previously shown in Table 5.16, in which we have higher rates for `Non-cheater tier-1` and `Non-cheater tier-3`. These two personalisation options are the ones that result from a lower `GamerProfile`; thus, meaning we had participants with lower scores in the prediction of the match's key events.

Regarding the evaluation of the synchronisation mechanism through statements S1-S3, 51.7% of the participants agreed with statement S1 and 34.5% strongly agreed with it, which means they had a good synchronisation experience ($\bar{x}$ = 4.2, σ = 0.66). When we asked if SMUF was easy to use (S2), the response was very positive ($\bar{x}$ = 4.2, σ = 0.86), as the majority of participants strongly agreed with the statement (48.28%). Once again, these participants also understood the issue that SMUF tried to solve, as they rated S3 very positively ($\bar{x}$ = 4.4, σ = 0.62), with 51.7% agreeing and 41.4% strongly agreeing with it. These results are very much in line with the previous phase, being even slightly higher than those results. It is interesting to compare the responses from participants that already have been in the previous phase (G1), which gathered experience with the WeSync use, with the new participants (G2). Figure 5.27 shows that the results obtained with G1 are actually higher than those of group 2. Participants in G1 are responsible for the improvement of the overall results, which may be due to the experience accumulated with the previous tests. It should be noted we observed that some of the participants in G2 had a little more difficulty dealing with SMUF.

As for the three following statements (S4-S6) regarding the participants' cheating perception, most of the participants in phase 2 stated that SMUF allows for unfair competition (S4), but the results collected in this third phase were very different ($\bar{x}$ = 2.0, σ = 0.85). According to phase 3's participants, SMUF does not allows for an unfair competition with 27.6% of them strongly disagreeing and 48.3% disagreeing with S4. Only 17.2% of them were undecided about statement S4, especially participants of G1. It is interesting to note that first-time participants (G2) disagreed more with S4 than G1's participants (see Figure 5.28). Moreover, S5 ("It is easy to cheat while using the synchronisation mechanism.") had similar results. Once again, overall, participants of phase 3 disagreed more strongly with this statement ($\bar{x}$ = 3.1, σ = 1.29) than participants from the previous phase ($\bar{x}$ = 3.8, σ = 0.90), which is a positive point supporting the changes made to CDM and CPM . As in S4, according to our perspective, first-time participants responded more positively ($\bar{x}$ = 2.9, σ = 1.37) to S5 rating it with lower scores, in average, than G1's participants did ($\bar{x}$ = 3.2, σ = 1.24).

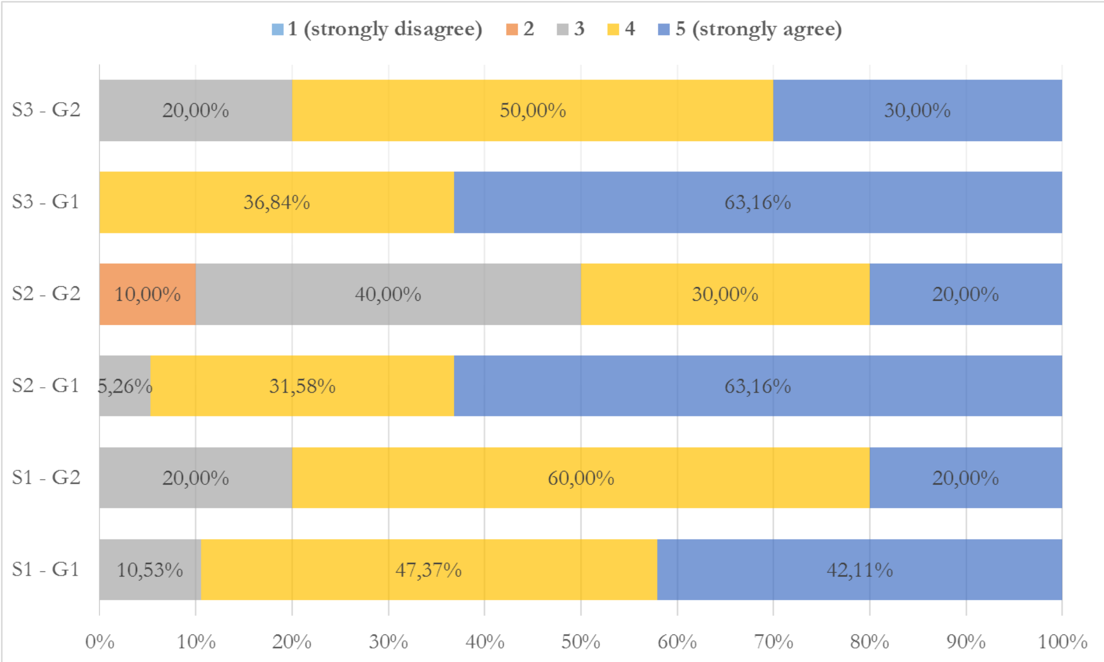


Figure 5.27 - Study EU.2: Summary of phase 3's questionnaire results regarding the SMUF experience (S1-S3) per group.

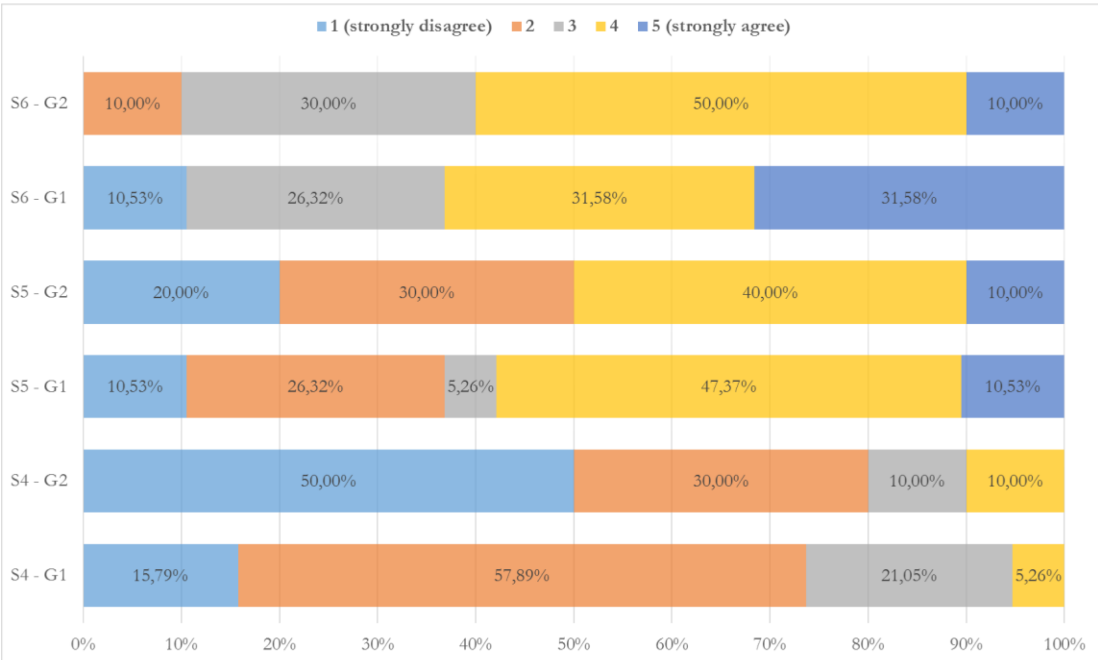


Figure 5.28 - Study EU.2: Summary of phase 3's questionnaire results regarding participants' cheating perception (S4-S6) per group.

According to what we gathered from the final interview and the observations, the participants' perception is that it is not easy to cheat since globally they noted the presence of a mechanism limiting that unfair action for the game. The ones that tried to cheat, or were marked as cheaters, did realise it would not be easy to cheat without consequences. Others did not even realised it and so, for them, it was natural that SMUF did not promote an unfair competition. Participants of G1, mainly the ones marked as cheaters at some given time, understood that they were facing an improved mechanism for cheaters detection and that penalisation application was being faster and harsher. Moreover, five of them realised it from the beginning since the system behind WePlayTennis recognised them as cheaters, from the first moment, due to their experience with WeSync. However, some participants still agreed to the two statements, as they thought they could still find ways to cheat the system, based on their experience as cheaters in the tests with WeSync. The result of S4 is better than the result of S5. Participants were agreeing with S5 and not with S4 just because they thought that it is up to each one finding ways to exploit the system; thus, they did not agree that the mechanism allowed for unfair competition.

As for S6, approximately 60% of the participants (strongly) agreed with it ($\bar{x}$ = 3.7, σ = 1.08), which is transversal to the two groups. The S6 result showed that the SMUF mechanism can motivate these participants to be honest while using it. On the one hand, we suppose these values may result from social peer-pressure, from being observed in the tests or seen as a dishonest person. On the other hand, three of them stated it was because the app did not let them exploit the system. Therefore, it may be due to social reasons or technological limitations imposed by the app, as the statement was a bit open to interpretations. Figure 5.28 details the results of S4-S6 per group.

The results of S7 are very similar to the ones from phase 2's S7, and they were not surprising as the analysis of the data logs showed us that only 34% of participants tried to cheat. Therefore, the results corresponded to those data from the logs, showing that almost 70% of participants disagreed with S7 since they did not cheat ($\bar{x}$ = 2.1, σ = 1.59) (see Figure 5.29 for details). Participants who were detected as cheaters (false positives included) also rated the statements S8.1, S8.2, and S8.3. Regarding S8.1, results were clearly positive as participants agreed or strongly agreed with it ($\bar{x}$ = 4.3, σ = 0.85). However, one participant rated S8.1 with two points, as s/he felt frustrated for being marked incorrectly as cheater. The overall result of S8.1 means that participants understood that the CDM was working properly, even for those corresponding to false positives. Regarding S8.2, the results were were not so positive ($\bar{x}$ = 3.5, σ = 0.96) since approximately 40% of the participants did not agree that the penalisation mechanism was adequate to its end. From our observations and final interview, we suppose that some participants, when classified as cheaters, felt frustrated for being under a more severe penalisation algorithm compared with the one implemented in WeSync.

Lastly, there was an evident disparity ($\bar{x}$ = 2.5, σ = 1.35) in the assessment of S8.3 ("I think I would not cheat, if I knew that there was a cheating detection mechanism beforehand."), as can be seen in Figure 5.29. Participants of G1 already knew about the cheating detection mechanism, as well as the penalisation mechanism, and the majority of them was really trying to cheat. Therefore, they naturally disagreed with the statement, as the mechanism did not influence them so as not to cheat. They were trying to test and exploit the system from the beginning. On the other hand, the three G2's participants, classified as cheaters at some point, agreed and strongly agreed with S8.3, which may indicate they were trying to use the app as best as possible and will use it in the future as fair as possible.

In phase 2, we detected participants with a clear cheater behaviour, so they became natural contenders for this third phase of tests, as we wanted to evaluate how they would behave when starting to use a similar application for the first time. We invited as many of them as possible to participate in this new experiment, now based on WePlayTennis. From the 19 volunteers that have integrated G1, six of them had a clear proactive cheater behaviour while using WeSync and another one had been classified as a cheater, at some point, but s/he never got more than a warning. Five of them ended their participation in the experiment of phase 2 with a profile of cheater. As a last question of the questionnaire, we asked participants to indicate if they were identified as cheaters when they started using the WePlayTennis app. From the 19 participants in condition to respond, we obtained a result of five, which were effectively the five participants coming from the WeSync's experiment with a profile of cheater. Therefore, the system worked according to our expectations, using the knowledge acquired with WeSync to improve the overall experience of the users with the new application, just from the beginning. Moreover, the results of S9 ("I think the system recognised me correctly after the previous use of the WeSync app") also support this important finding since almost all participants gave their agreement (agreed (31.58%) and strongly agreed (57.89%)) to the statement ($\bar{x}$ = 4.5, σ = 0.7). Two of them had doubts, but it could have been because the statement would not be very clear and they would be expecting something specific. However, we only had a message to the ones starting WePlayTennis as cheaters.

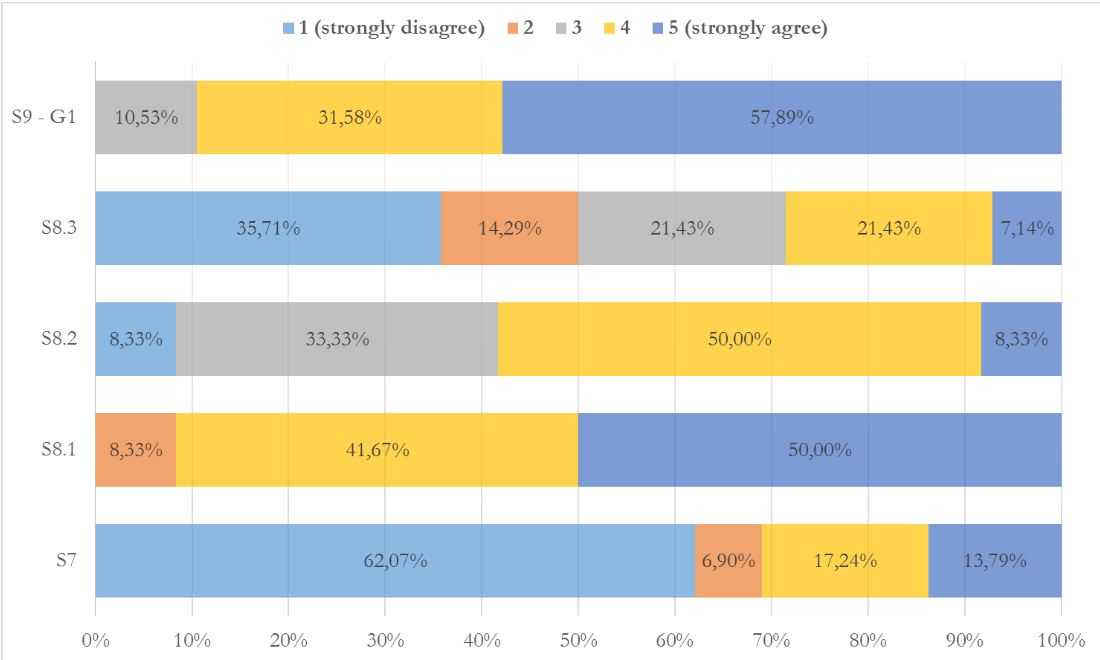


Figure 5.29 - Study EU.2: Summary of phase 3's questionnaire results regarding the CDM and CPM components (S7-S8.3).

Finally, it should be noted that three participants, who were accustomed to interacting with the Instagram application, told us they found similarities between the ways WePlayTennis and Instagram deal with people trying to exploit the corresponding system, which is an important finding for us.

5.2.4. Final Remarks

In order to design the best solution possible, we designed a research plan comprised of three main phases, as can be seen in Figure 5.30. In the first phase, we carried out user experiments with the WeSync app in order to evaluate the SMUF's usefulness and viability in a competition scenario and, more important, we collected important interactions data to define the best possible personalisation. Afterwards, we designed the personalisation for Sync2FairPlay, based on the gaming and synchronising profiles, and implemented the algorithms behind CDM and CPM. In the second phase, we evaluate the system as a whole, studying how participants interacted with it and collecting data to analyse how well CDM worked, verifying if the personalisation definition would need some tuning. Before the third phase, we did some fine tune of the modules, regarding their algorithms, and analysed users' interaction data to select and invite participants for the last phase. In the last phase, we evaluated the knowledge about the users' behaviours that WePlayTennis could have from the beginning due to the use of P²MUCA, taking advantage of the previous users' interactions with WeSync. The system worked according to our expectations, using the knowledge acquired with WeSync to improve the overall experience of the users with a new application, just from the beginning.

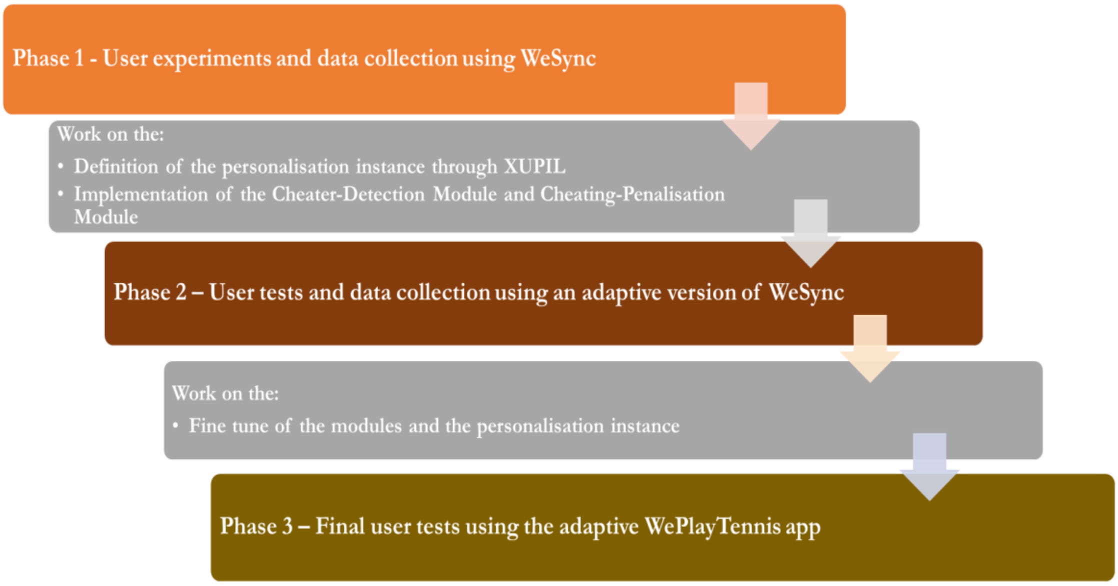


Figure 5.30 - Study EU.2: the research plan comprised of three main phases.

The results obtained with the successive phases of tests were very positive, validating the overall concept of Sync2FairPlay and its different components. They showed us that participants were able to synchronise the second screen applications with simulated TV broadcasts, while having, as much as possible, a cheating-free experience in a competition scenario. Overall, our personalisation-based approach was able to detect users trying to cheat, based on users' interaction data, like the current users' delays and the number of key events answered. The ones identified as cheaters were penalised, which prevented them from continuing to use SMUF to exploit the system, taking advantages in the competition. With further use of Sync2FairPlay-based applications, the definition of the two parameters used in the CDM's personalisation instance can be improved to obtain even less false positives and to address the issue related to the radio scenario, which was mentioned in the results of phase 2.

The personalisation applied in this study was even more important as we needed to maintain possible cheaters identified across different applications, which worked smoothly. The results were very interesting and promising since they showed that, with the use of P²MUCA, we could have various applications simultaneously using, a more extensive and consolidated, knowledge about the end users in order to be automatically personalised from the beginning towards them. P²MUCA provided the ability for both applications to exchange data safely, which is called interoperability. This study has successfully responded to **RQ**$_C$ as our model-driven platform provided the interoperability and data sharing feature that allowed both applications to offer a better-personalised experience to their users.

Lastly, this study also worked as another evaluation of P²MUCA, which has responded with success to the requests of third-party applications. Moreover, the evaluation of P²MUCA meant the evaluation of X-Users as its underlying model, XUPIL as the language used to instantiate the personalisation implemented in the cheater-detection module, and CAPE as the core component of the platform. Therefore, this study allowed testing the solution's performance and viability as a whole, for the second time, in response to another realistic scenario.

5.3. Study EU.3 – Involving End Users in the Personalisation Design

As in the two previous studies, we wanted to find out if personalisation would add real value to human-computer interaction to engage better the end users (patients, in this case) in the use of the applications. All the previous studies were designed to involve our research team, prototypes' developers, and the participants (developers or end users) in the tests. But at this stage, we wanted to go one step forward and take another approach, involving end users in the design process. We wanted to confirm the lessons learned from the previous studies and draw guidelines about designing and implementing personalisation. We needed to have their participation at different moments along the implementation of a personalised application, since we wanted to understand better how they think about the goals, design, and user experience of that application in order to draw a generic personalisation process. To accomplish this goal, we planned a Participatory Design[13] (Muller, 2002) approach to involve the end users, besides the stakeholders, designers, and researchers, in the design process of a personalised prototype to help ensure that it would meet the needs of its intended user base.

The prototype used in the process emerged naturally from our involvement in a funded research project[14] that integrated the development of PhysioMate, a system based on the use of natural user interfaces (NUI) and serious gaming for physical rehabilitation. Moreover, a physiotherapy clinic, after participating in a testing session, asked us to develop a particular case of PhysioMate, which was "just Physio Kidding" (jPk), towards children with special needs. In this way, the stakeholders involved were the managers/coordinators of the funded project and the physiotherapy clinic. Moreover, our team involved the researchers and designers along the way. Lastly, we wanted to involve expert users, besides common users that cannot be forgotten (Nielsen, 2010). However, the involvement of the former has become increasingly important as the balance between novice and expert users tilts in the direction of the experts (Nielsen, 2010).

At this point, it is important to clarify that, to the current research study, the expert users (experts) involved in the process had to comply with two requirements:

1) several years of experience using applications, technologies and tools related to the prototype we were designing;
2) be professionals in the prototype's domain (physical medicine and rehabilitation field), whose vision would be important to design the best possible application for them and their patients (the novice or common users who are not responsible for knowing the scientific and clinical principles that should support the application development).

Thus, after having a first version of the PhysioMate prototype, we conducted a user study with experts, the therapists (usually, physiotherapists) and rehabilitation-related healthcare professionals, who could use the PhysioMate system with their patients. The goal was to evaluate the system's concept, its usability, and the integration of personalisation in the prototype. After that, with the involvement of the physiotherapy clinic, we could get the involvement of the experts' team from the clinic in the design and evaluation of the second prototype ("just Physio kidding"). One of the experts was also the manager of the clinic, being the coordinator of their team and the point of contact with our team of researchers.

One more time, the use of P^2MUCA was implicitly important here, so this study also served as a field assessment of our personalisation solution's performance.

[13] Participatory Design was also called co-operative design (from its origins).
[14] TailorPhy - Smart Sensors and Tailored Environments for Physiotherapy (grant PTDC/DTP-DES/6776/2014).

5.3.1. PhysioMate/just Physio Kidding: Pervasive Physical Rehabilitation based on NUI and Serious Gaming

The main idea behind PhysioMate is the implementation of a richly interactive and smart rehabilitation exergaming system towards each one of its users (Madeira, Luís Costa and Postolache, 2014). Initially, the project was designed having in mind the rehabilitation of wheelchair-dependent patients (Madeira et al., 2010), whether they are, for example, victims of stroke, older adults, victims of neuromuscular diseases, or other people with reduced mobility.

Motivation and background

Stroke survivors need extensive rehabilitation work in order to achieve independence in daily living activities, having to attend long sessions of physiotherapy and, many times, in a wheelchair. It is required a high motivation from the patient in order to comply with crucial treatments and therapies (Ferreira et al., 2014). The gamification (Deterding et al., 2011) of these therapies' exercises and their deployment in ubiquitous and pervasive devices, such as smartphones or tablets, may provide a more engaging and compelling rehabilitation, even at patients' homes (Ferreira et al., 2014). The integration of gamification elements provides selective attention and promoting emotional involvement or social comparison of effort and progress (Helf and Hlavacs, 2016). Furthermore, serious games can provide rehabilitation environments that can increase the motivation of patients to achieve successful completion of rehabilitation programs that can be dreary or very demanding (Wiemeyer and Kliem, 2012). Serious games are games with a goal that moves beyond entertainment, delivering engaging interactive media to support learning in its broadest sense (Stone, 2008). The growth of serious games as an industry and research field evidences that games and their related technologies increasingly transcend the traditional boundaries of their medium (Deterding et al., 2011).

Healthcare rehabilitation has emerged as a leading target area for serious games providing new patient, health caregiver, and public expectations. Several approaches to serious games-based therapy (theragames) appeared showing a broad general interest in improving and sustaining this concept. A user can to train and enhance essential functions or specific skills, receive important task-oriented information and perform task-oriented exercises, involved in a gameplay scenario to accomplish the goals of the game's activities and levels. For instance, DroidGlove is an interesting Android-based serious game for wrist rehabilitation that runs on small portable devices (Deponti et al., 2009). It turns the device into a wrist rehabilitation tool, with motion and position sensors letting to drive the user through the right sequence of movements to recover hand mobility and articulation. In the neuro-rehabilitation field, it is recognised that serious games function as a robust methodology to increase motivation and active participation in exercises. (Ma and Bechkoum, 2008) developed a serious-game based therapy to stimulate physical exercises practice by stroke patients with upper limb motor disorders. The system allows patients to interact in real-time with virtual objects to practice specific motor skills through diverse modalities.

The integration of virtual reality (VR) technology with exergames provides more motivation and engagement to patients while in rehabilitation (Chang et al., 2012). Therefore, different motion capture systems have been developed and applied to rehabilitation. The most precise is a system of reflective markers attached to the bodies and limbs of patients, which are tracked by optical sensors to determine their 3D position. However, these markers are usually cumbersome and uncomfortable, being expensive and hard to deploy in a clinical setting (Chang et al., 2012). On the other hand, it is possible to use commercial game interface devices to infer mechanical motion, such as Nintendo Wii U Remote, Sony PlayStation Move, Sony PlayStation EyeToy and the Microsoft Kinect, which are less complex and of relatively low-cost. The fusion between augmented or virtual reality and NUI is rather a successful one and it can be seen when looking at conceptual product ideas and, especially, the Kinect's success stories. Kinect is the most well-known NUI device. NUI can be seen as the ability to interact with a machine using nothing but the human body, being powered by touch, gestures, sound, or senses. Depth detection cameras, like the ones in Kinect devices, determine the distance to the picture elements, which enables gestures to be identified without any marker.

This way, nowadays, it is possible to find many different projects using Kinect for rehabilitation. For example, (Chang et al., 2011) present a system to assist therapists in the students' rehabilitation in public school settings. This solution automatically determines whether the students' movements have reached a rehabilitation standard. The main goal is to give students the opportunity of having a less monotonous recovering process and to give the therapist a useful tool to monitor the students' information and progress. REMOVIEM (Lozano-Quilis et al., 2013) is an interesting project that is being used in the rehabilitation of patients from the Association of Multiple Sclerosis of Castellon. It promotes body movements, tracked by Kinect, through three motor rehabilitation games that try to improve the gait, the balance and weight transfer of patients. A project that has points in common with PhysioMate is the one presented by (Saini et al., 2012), which consists of a home-based platform that includes an online biofeedback component to gather the motion tracking and exercises. It allows the system to store patients' movements and knowledge results, displaying them to both patients and their therapists.

(Luna-Oliva et al., 2013) made a study providing evidence about the usefulness of Kinect Xbox 360 as a therapeutic modality for children with cerebral palsy (CP) in a school environment. The outcome measures showed improvements in different tests. A solution often followed consists of a game composed of some minigames, which is the case with Liberi (Hernandez et al., 2013). It was developed to help children with CP exercise more, in this case, pedalling on a recumbent bicycle to control characters in the minigames. The authors found the majority of the target group preferred to play action games, even though they had difficulties playing them. They proposed a set of usability guidelines (Hernandez et al., 2013), such as avoiding a fast pace, not requiring precision timing and not requiring multiple simultaneous inputs in order to mitigate some of their weak performance. Along with the previous work, (Chang et al., 2013) showed that two adolescents with CP demonstrated high motivations for exercising with Kinect and improved their performance during the intervention.

Several researchers emphasised that the design of a good game should be considered as equally important to the use of custom hardware and therapy requirements. (Notelaers et al., 2010) suggested that design aspects, such as clear movement feedback, are essential elements to design a good game for Multiple Sclerosis patients. A series of general game design principles and design patterns were formulated and applied to the development of custom-made games for rehabilitation (Burke et al., 2009; Notelaers et al., 2010). (Annema et al., 2010) implemented several design principles, including a smooth start-up and configuration, to aid the therapist and ensure a motivating therapy session. There is the work from (Abeele et al., 2010), which encouraged researchers to integrate the principles of 'slow fun' (i.e., no time-dynamics) in digital games for rehabilitation for spasticity. Moreover, (Alankus et al., 2010) highlighted the relevance of audio and visuals. Lastly, there is an interesting proposal using a procedural content generation system and propagating this as a viable way to personalise rehabilitation sessions (Dimovska et al., 2010).

PhysioMate overview

PhysioMate includes components of serious games, which were only programmed to include elements of gamification at the beginning of the project. We are also exploiting the use of NUI devices and, in a first phase, we started by using Microsoft Kinect to infer mechanical motion of patients, since it is a relatively low-cost consumer game interface device and is easy to set up. PhysioMate relies on the serious gaming and NUI concepts in order to encourage physical exercise to combat both physical and cognitive deterioration, and it should be a complement to the work of physical therapists, with and without their real-time supervision. However, PhysioMate is aimed to be used initially in the scope of a physiotherapy program under the supervision of a therapist. Therefore, the therapist plays an important role in this system, since s/he is the user who is responsible for:

- configuring the PhysioMate components according to each patient's individual needs;
- supervising patients' performance and progress.

Concerning scenarios of use, in the professional rehabilitation space (e.g., a clinic), we can have multiple "workstations" with one display and a Kinect device for each user, all of them connected to the same computer (middle server). Each display will show its application running. In a second scenario (e.g., in the patient's home), there is only one Kinect and one display connected to the home computer, for instance. In both scenarios, data will be sent to the PhysioMate server in order to be processed and integrated into a global view. This allows having appropriate supervision, even when we have a patient that is in her/his home or doing rehabilitation exercises more independently in the clinic.

Since initially the project was thought to be used in the rehabilitation of wheelchair-dependent victims of stroke, the research team selected a set of essential primary movements of the upper-body, more specifically of the upper-limb, for balance training and motor coordination (Table 5.17). The game mechanics of the PhysioMate components must follow this set of movements in order to address the typical rehabilitation routines towards the target patients. PhysioMate even provides an interface where it is possible to combine these primary movements to generate more complex routines that can be used in the games.

Movement	Description
Weight transfer	transferring the weight of the body from one hip to another
Side reach	inclination of the trunk - for both sides
Anterior reach	take torso forward
Rotation of the trunk	for both sides
Later reach	bring torso back

Table 5.17 – PhysioMate/"just Physio kidding": The set of primary movements.

PhysioMate is then composed of two different genres of NUI-based serious games: routines; and challenges. Regarding the routines set, the patient plays games that are routines of movements created by the physiotherapist. The main aspects of this game are the following: directed only for physiotherapy, there can be scheduled routines that integrate a global plan of rehabilitation exercises created by the therapist, who can, for instance, create and add routines to the system, supervise what her/his patients are doing, and analyse patients' progression in a particular routine. About the challenge games (Figure 5.31 shows the "Recycling Therapy" game), they address cognitive stimulation and their mechanics must integrate the primary movements or routines previously configured.

Platform and design of "just Physio Kidding"

"just Physio Kidding" is a PhysioMate sub-project designed to address the therapy work in a clinic that deals with children with special needs, such as the cases of children with spinal muscular atrophy or developmental delay, besides the ones with CP. For instance, Children with CP have muscle weakness, reduced range of motion, and poor control over their movements, which pose additional difficulties and challenges with the finely controlled movement required by Kinect (Hernandez et al., 2012). This way, it is required to provide a high motivation to these children to comply with crucial treatments and therapies. The gamification of these therapies' exercises and their deployment in ubiquitous and NUI based devices may provide a more engaging and compelling rehabilitation (Hamari et al., 2014), even at patients' homes with the supervision of the parents. However, the implementation of "just Physio kidding" needs to take into account both the specific context being gamified and the qualities of the end-users (Hamari et al., 2014), children with special needs, in order to obtain the desired outcomes.

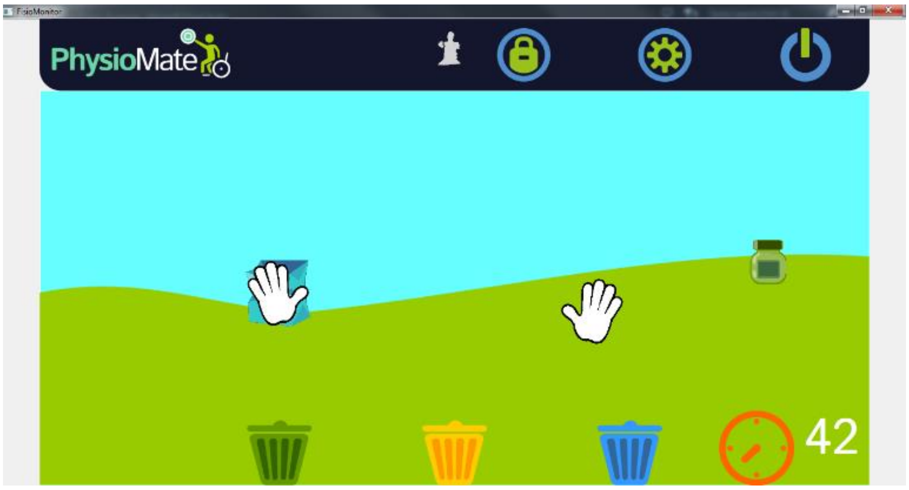


Figure 5.31 - PhysioMate: The "Recycling Therapy" challenge game.

"just Physio kidding" is implemented upon PhysioMate but, meanwhile, the Leap Motion[15] sensor was integrated into the system. It was a new requirement since the clinic's therapist needed to follow different therapy exercises with patients, to improve gross motor skills, at upper and lower limbs and torso, but also fine motor skills, at hands and fingers. There are four main differences about PhysioMate: a new user interface design, more games (themes) that apply to children, incorporation of game movements addressing the fine motor skills of hands and fingers, and the integration of personalisation (presented in section 5.3.2 and discussed in section 5.3.3). Moreover, "just Physio kidding" is now provided by an integrated platform giving tools to the therapists to customise the game and monitor the patients' performance and their detailed data. It is comprised of six main components (Table 5.18).

JPK	**a desktop NUI-driven patient application**
JPKT	a desktop therapist application (allows NUI-based control)
JPKW	a therapist and administrator website
JPKM	shared models and web services client
JPKDATA	shared data models
JPKREST	RESTFul web services

Table 5.18 – "just Physio kidding": The components of the platform.

The JPK and JPKT applications are both NUI-controlled (Kinect and Leap Motion). The patient has access to the gaming component (JPK), and the therapist has access to JPKT (Figure 1), which allows her/him to configure and schedule games, analyse and monitor her/his patients' sessions in loco. The Web interface (JPKW) provides remote access to the supervision tools (Figure 5.32), but adding more detailed statistics. The therapist plays an important role in this system since s/he is the user responsible for configuring games according to the patient profile and supervising patients' performance and progress. The patient's performance can be analysed with the help of data charts and 3D

[15] Leap Motion controller - https://www.leapmotion.com

visualisation of sampled patient's skeleton movements (see Figure 5.33). All components and their applications share a common data access point, the web services client JPKM, which allows having a complete separation between logic and data. The applications share a common data model that is implemented as a shared class library. Data is stored locally for each platform installation, since the system can be distributed across various therapy spaces, and synchronisation with the platform's central database is performed at each local platform initialisation.

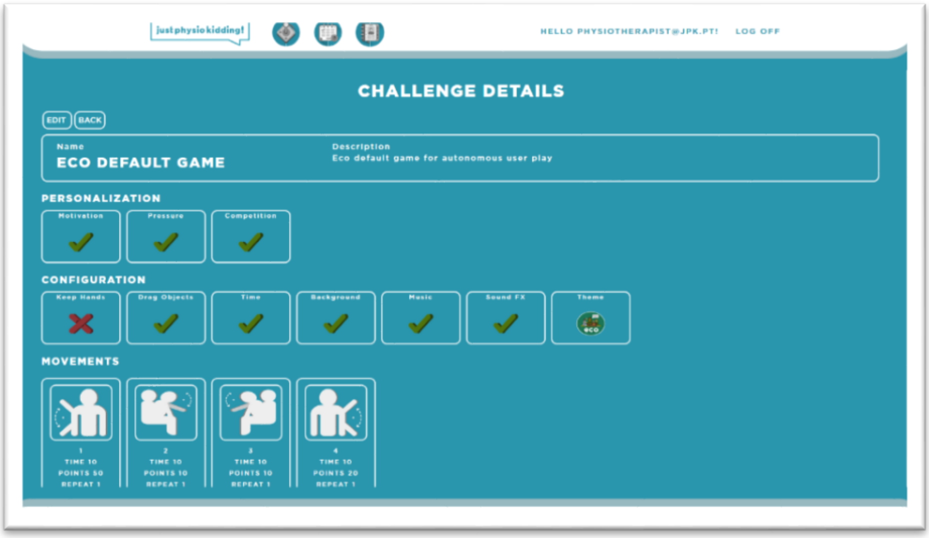


Figure 5.32 – "just Physio kidding": JPKW's interface for game customisation (configuration) where each personalisation instance can be controlled by the therapist.

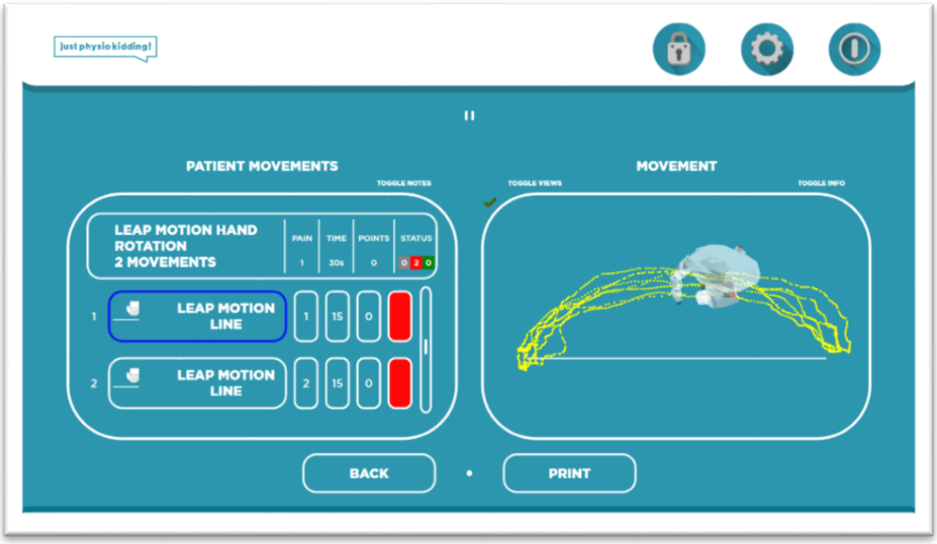


Figure 5.33 - "just Physio kidding": Analysis of a patient's game movement in a JPKT's interface.

The platform includes four games, which were previously called Challenges in PhysioMate. They are available for autonomous execution by the patients, or their caregivers and therapists, through the Games button, but they can also be scheduled to the patient by the therapist. The first three games are Kinect-based only. Actually, the three share the same basis regarding interaction design and gameplay, but each one of them presents its theme: Eco, Noir and Bill. We started by re-implementing "Recycling Therapy", which originated "Eco" (Figure 5.34) since it addressed a theme easily recognisable by children and already evaluated in the first tests with good results. The Noir theme is about a detective and objects that should be identified as clues to an important case under investigation (Figure 5.35). The third game has something to do with the Wild West and precious metals that have to be harvested, identified and put into the right railway wagon in order to be kept safe from the great evil enterprise. For each game, there are three groups of items and the corresponding containers. Objects from a group share common identifying characteristics, like the colour or the material type, suggesting to the user where they should be placed.

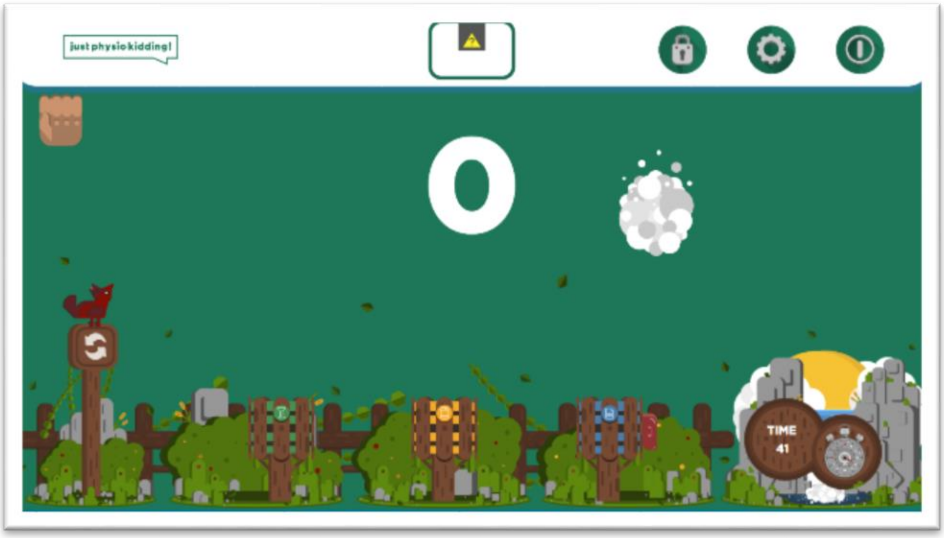


Figure 5.34 - "just Physio kidding": The challenge game "Eco, The Last Boy Scout".

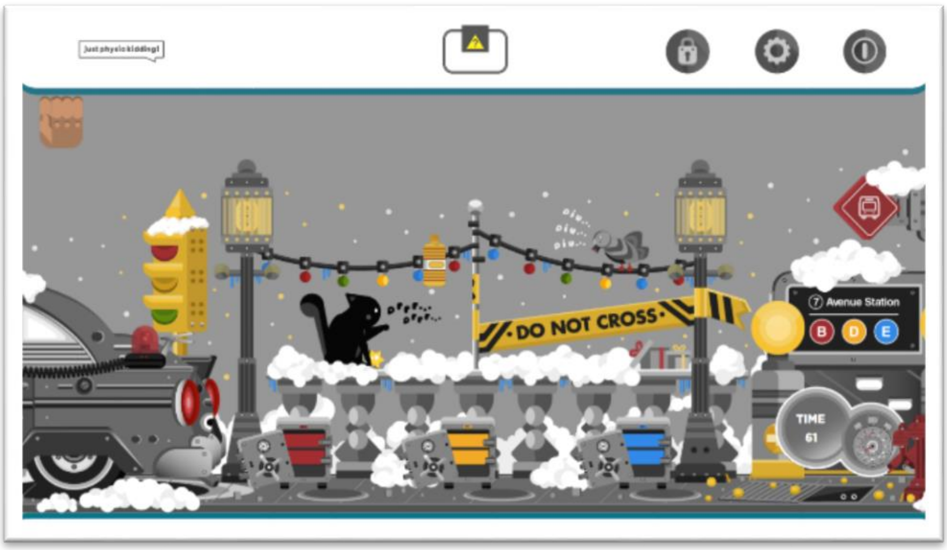


Figure 5.35 - "just Physio kidding": The challenge game "Noir, Cats Have 7 Lives".

The user interaction with the games must be done executing the required movements, which are encapsulated by the gameplay. The objects are placed on the screen at predetermined positions to require the execution of one or more primary movements to accomplish the action of reaching the object and dragging it to the appropriate container. The game containers are positioned at the bottom of the screen, at left, centre and right locations. The required hand for a movement must be used for it to be recognised as valid. Game variations to the default game movements for each theme can be created using the therapist's applications JPKT/JPKW. The intended result of a game movement can be changed to be a simple reach of the object (touching it), instead of having to drag it. Other options are keep touching the object for a few seconds, grabbing (closed hand) the object while dragging it, among others. The fourth game is a multi-sensor game with a different design. This game uses both the Kinect and the Leap Motion sensors to track and validate the performance of the movements. The user has to follow and reach the objects that appear sequentially on the screen and then execute hand/fingers movements with/on them.

The games' data of authenticated users are sampled and recorded by the platform. It includes game-specific data like the movement score, duration, and status, but also Kinect data of the skeleton's joints and bones and Leap Motion data of hands and fingers. Regarding the game mechanics, the jPk game uses points, levels, leaderboards, badges, and challenges/quests, as tools to create a meaningful response (Zichermann and Cunningham, 2011). The quests are put to the patient as simpler games that are routines of movements (from PhysioMate) created by the physiotherapist. They are directed only for physiotherapy training/exercising that should be executed according to the results of the main games. There can be scheduled routines that integrate a global plan of rehabilitation exercises created by the therapist, who can, for instance, create and add routines to the system, supervise what her/his patients are doing, and analyse patients' progression in a particular routine.

"just Physio kidding" implements a ranking system to inform the patient of her/his level of familiarity and to reward continued expertise using the system. The levelling progression is adapted to the patient to keep her/him engaged and motivated with the system, and rehabilitation program, as much as possible. Therapists can customise the games towards each one of her/his patients, defining profiles taking into account that each child presents particular characteristics. The way engagement with the games works with one child may differ greatly from another child. It is important the user experience be reflected in a progressively nonlinear scoring and a ranking system with a leaderboard to document progress benchmarks. The games can be customised according to each patient, based on different features like: time limit on/off; number of objects to caught; objects appearing on predefined zones; only touch the objects, caught them and/or drag them to containers; background complexity.

The system awards users with badges and trophies as a means to reward the user for overcoming a specific action or task, providing a more friendly progression usage meter and promoting user engagement. The badges and trophies are grouped into the following categories: (1) game usage badges; (2) game trophies; and (3) global trophies.

5.3.2. Personalisation Design for "just Physio Kidding"

We applied personalisation to "just Physio kidding" with the involvement of the experts' team (see section 5.3.3), which helped us to find the aspects that could be important to benefit from it. We decided to implement three personalisations in order to start the process, but with the requirement of not being too much overwhelming for both the experts, participating in the design process, and the children, who would participate testing the application. Besides that, important features, such as, the levelling and progression and the movements included in a game would not be automatically adapted to the patient, as the experts would prefer to customise them according to the rehabilitation intervention that they were applying. The correct personalisation of these aspects would be something to analyse carefully over several weeks using "Just Physio Kidding" with patients. In this way, it would be possible to collect data and observe the influence of the application on the therapeutic progression of children. Thus, it was decided to start with personalisations to address psychological aspects, such as, engagement and motivation to use the system and influence of pressure and competition on the realisation of the game's movements. Table 5.19 contains the set of resources that were considered as being those that could provide interesting data for the personalisation of "just Physio kidding" towards the patient's adherence to the therapeutic practice.

R	Resource	Description
Interaction stream resources		
1	**ranking**	[replace] Current patient's position in the ranking for the specific challenge.
2	**lastChallengeScore**	[replace] Score of the last challenge game played by the patient.
3	**meanLast3challenges**	[replace] Average score for the last three challenge games played by the patient.
4	**resultsScreenTimeSpent**	[sum] The time spent by the user/patient in the results screen.
5	**resultsScreenConsults**	[increment] The number of times the user has been in the results screen.
6	**badgesTrophiesScreenConsults**	[increment] The number of times the user has been in the trophies screen.
Preference resources		
7	**clinicalConditionSeverity**	[question_MC] The general patient's clinical condition severity for the rehabilitation process, which is a value in [1-5] always updated by the physiotherapist whenever there is a change in the condition.
8	**emotionalState**	[question_MC] The patient's current emotional condition that is a value in [1-5] always updated by the physiotherapist at the beginning of a session.
9	**motivationLevel**	[question_MC] The patient's current motivation level to participate in a jPk-based rehabilitation session, which is a value in [1-5] always updated by the physiotherapist at the beginning of a session.

Table 5.19 – Resources used in "just Physio kidding" (interaction stream data and preferences used in the personalisations).

P1) Motivation

This personalisation is based on the "predisposition profile" of the patient. If it is low, then the personalisation option will indicate that the game elements should be modified in order to motivate the patient, trying to be more persuasive towards making her/him use the application more frequently and with more interest. At the opposite extreme, if the predisposition is high then the game will keep the standard elements. Table 5.20 shows three possible personalisation options for this particular personalisation, according to the three options (clusters) for the parameter `predisposition-Profile`.

predispositionProfile	*Personalisation option*
low	highMotivationElements
medium	motivationElements
high	standardElements

Table 5.20 - "just Physio kidding": three personalisation options for the "motivation".

This predisposition profile parameter is calculated by expression (23), which is based on resources that are defined as user preferences. The `clinicalConditionSeverity`, `emotionalState` and `motivationLevel` values are assigned to the patient by her/his physiotherapist (or other professional in her/his place), who chose a value between 1 and 5 for each state. On the one hand, the higher the values of both the `clinicalConditionSeverity` and the `emotionalState`, the lower the patient's clinical and emotional conditions for the rehabilitation session. These "in-verted scales" were designed by the experts' team based on the literature of the domain followed by them. On the other hand, the `motivationLevel` is straightforward with a value of five meaning the patient is as much motivated as s/he can be, according to the knowledge her/his therapist has about her/him. Naturally, a value of one means the opposite.

$$\frac{1}{clinicalConditionSeverity} * 0.2 + \frac{1}{emotionalState} * 0.4 + motivationaLevel * 0.4 \quad (23)$$

P2) Competition

Usually, children like to be part in competitions against friends or, at least, against their own past performance. Although "just Physio kidding" has been designed for children with special needs, but according to the experts involved in the design, regardless those special needs, it is possible to find patients who really like to compete and others that stress in that context. Therefore, in this personalisation instance, the visualisation of the score and ranking screen at the final of a challenge game will be based on three new parameters: `competitiveProfile`, given by expression (24), `psychoProfile`, given by expression (25), and `performanceProfile`, which is expressed by (26). There are four possible personalisation options, according to the combination of the three profiles (see Table 5.21).

$$\frac{resultsScreenTimeSpent}{resultsScreenConsults} * 0.5 + resultsScreenConsults * 0.2 + badgesTrophiesScreenConsults * 0.3 \quad (24)$$

$$(6 - emotionalState) * 0.5 + motivationLevel * 0.5 \quad (25)$$

$$\frac{1}{ranking} * 0.5 + meanLast3challenges * 0.3 + lastChallengeScore * 0.2 \quad (26)$$

<table>
<tr><th>Personalisation option</th><th>competitiveProfile</th><th>psychoProfile</th><th>performanceProfile</th></tr>
<tr><td rowspan="2">globalRanking</td><td rowspan="2">competitive</td><td>OK</td><td rowspan="2">high</td></tr>
<tr><td>KO</td></tr>
<tr><td rowspan="2">selfRanking</td><td>competitive</td><td rowspan="2">OK</td><td>low</td></tr>
<tr><td>nonCompetitive</td><td>high</td></tr>
<tr><td rowspan="2">gameResult</td><td>competitive</td><td rowspan="2">KO</td><td>low</td></tr>
<tr><td>nonCompetitive</td><td>high</td></tr>
<tr><td rowspan="2">nothing</td><td rowspan="2">nonCompetitive</td><td>OK</td><td rowspan="2">low</td></tr>
<tr><td>KO</td></tr>
</table>

Table 5.21 - "just Physio kidding": four personalisation options for the "competition".

The personalisation is much related to the way the user/patient interacts with the screens where appear her/his table of results and the trophies/badges s/he has conquered. Usually, if the user consults this kind of screens a lot, then s/he might be very competitive. The patient's most recent performance is also important to define how the score and the ranking screen should appear to the patient. Certainly with a good performance comes the patient's interest in seeing how s/he behaved relative to others. For instance, a patient with a competitive profile and high performance will see the global ranking, regardless her/his psychological profile for the day. It happens the same in case the patient has a non-competitive profile and shows a low performance, but this time s/he will see nothing. However, the psychological state is important and determinant when the patient is competitive and has low performance, or the opposite, as can be seen in Table 5.21. All these profiles and personalisation options were designed after tests sessions with the patients, supervised by the experts' team, usually, the physiotherapist leading the team.

P3) Pressure

This personalisation instance appears in line with the previous one, being closely related. According to the therapists, this personalisation is important since in some cases there is margin to put more pressure on the patient, even if s/he is a child. The personalisation "competition" was important to engage patients towards the game and, naturally, its underlying therapeutic intervention, adding a subtle pressure factor. But for the current personalisation, sometimes, the therapists want to see how the patient handles the exercises incorporated into the game when s/he is under more pressure, with a subtle competitive factor. Thus, a pressure element might work well, being useful to test how patients efficiently progress.

The visualisation of the clock/time and score during the challenge game will be dependent on two parameters, which are the `predispositionProfile`, already focused on P1, and the `performanceProfile`, which was presented in P2. There are four personalisation options as can be seen in Table 5.22. According to the experts' team, at this point, the most important condition to (de)activate time and/or score is the predisposition profile. If the patient appears in the rehabilitation session with a low predisposition, pressure elements should be off. On the other hand, a high predisposition means the patient may be able to face a game scenario under more pressure.

		performanceProfile	
		low	high
predispositionProfile	low	timeOFFScoreOFF	
	medium	timeOFFScoreON	timeONScoreOFF
	high	timeONScoreON	

Table 5.22 - "just Physio kidding": four personalisation options for the "pressure".

5.3.3. User Study with End Users

The user study was organised in three phases, or we can think about it as being a study composed of three smaller user studies. Initially, we wanted to test the PhysioMate prototype among healthcare professionals, especially physiotherapists, who we considered as the experts that should use the application with their patients. In the second phase, after implementing "just Physio kidding", we tested it with the end users, which were the patients (children with special needs) and the experts' team from the clinic that proposed the prototype's development. Finally, we carried out one last user study to evaluate the personalised version of "just Physio kidding". This last user study was conducted to gather preliminary results in order to evaluate if the project was going in the right direction, applying personalisation and using the P^{2}MUCA solution. Furthermore, we have already scheduled a thorough testing phase to move forward briefly. It will be conducted with a full-functional personalised version of "just Physio kidding" working "in the wild", that is, in the clinic for several consecutive weeks.

Design and participants

The design of the study was based on a participatory design approach. Thus, the study plan integrated not only the moments for user testing, but also the different moments in which we involved end users, mainly the experts, in the design process. Figure 5.36 provides an overview of how the three user testing moments were intercalated with the design and implementation phases.

After finishing the first prototype, we carried out user tests with a panel of experts in order to assess the PhysioMate's usefulness and usability, and to gather insights on how its games could be automatically adapted towards its end users. Therefore, we selected the experts for the panel from participants of two workshops that we conducted to present PhysioMate and other related prototypes under the scope of the funded project TailorPhy. At each workshop, we presented the system, made a demonstration, and asked participants to use PhysioMate. Regarding tasks, first, they had to use the application for the therapist to create routines, then they had to perform and test them in the Routines game, and finally they played the "Recycling Therapy" game. After testing PhysioMate, the participants filled in a questionnaire, presented in Table 5.23, Table 5.24 and Table 5.25, and we ended the session conducting a brief informal interview with each one, whenever it made sense.

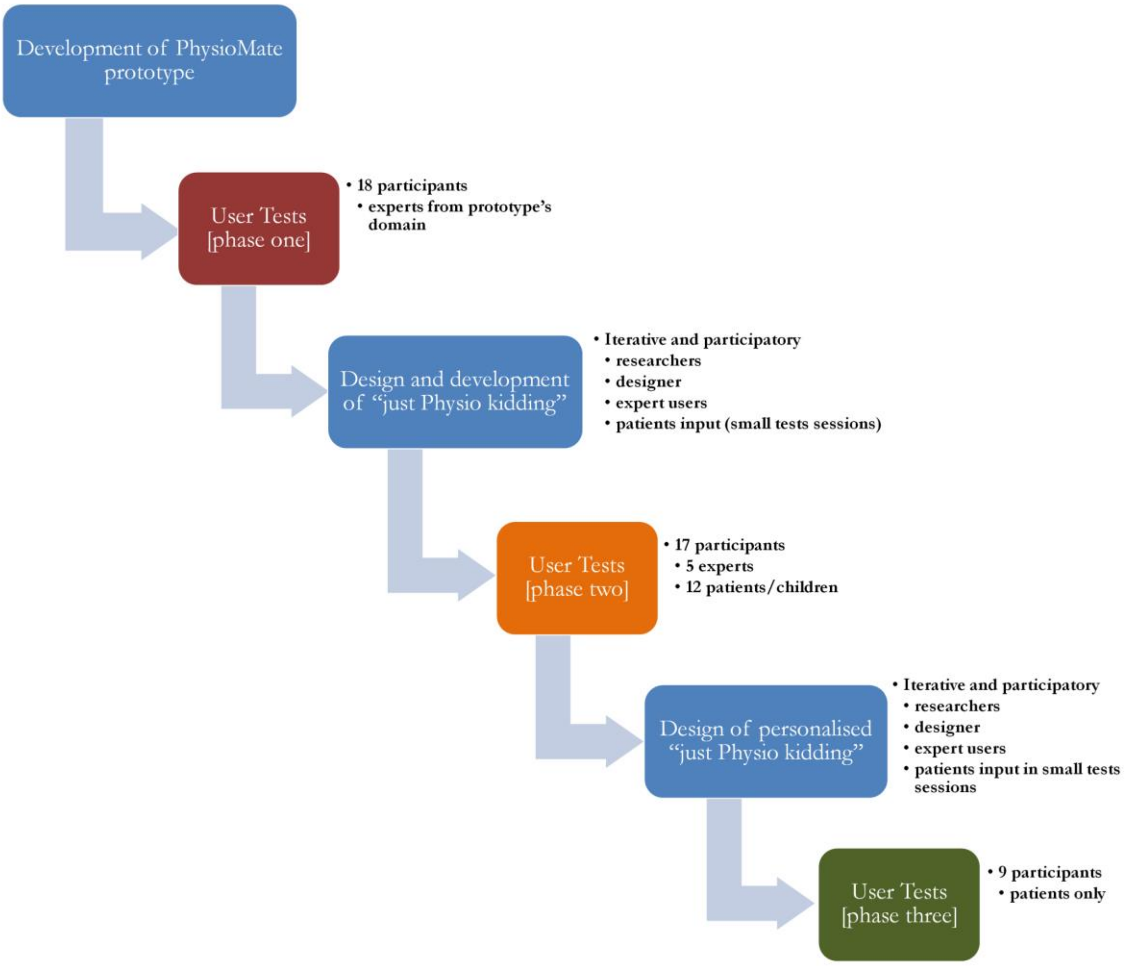


Figure 5.36 - Study EU.3: Overview of the study plan showing the main phases.

The user tests were conducted with 18 volunteers (7 male and 11 female) aged 21 to 56 ($\bar{x}$ = 34.9; σ = 10.1). Not all participants were physiotherapists. We had two physiotherapy students, one programmer of systems with similarities to PhysioMate, one biomedical engineer that was involved with adaptive technologies for amputees, and the remaining were the physiotherapists. In terms of years of experience in the field of rehabilitation, participants presented values ranging from 0, the students, to 35, the oldest ($\bar{x}$ = 11.9; σ = 9.9). It was important to consider students since they are naturally motivated to bring new approaches and are used to deal with these new devices and solutions, while the perspective of older professionals allowed us to observe how they are still open to work with solutions for rehabilitation based on recent technologies. Only nine of them already knew the Kinect device before being presented to it at the workshop. None of them had worked with it in rehabilitation scenarios; however, eight participants have used other game consoles, such as Nintendo Wii, with commercial exergaming applications not exactly developed for specific therapies. Furthermore, they all stated that patients showed high motivation while using them, which is exactly what we expect to happen with PhysioMate.

Afterwards, we got the involvement of the physiotherapy clinic, which rehabilitation work was focused on children with special needs, for the design and development of the second prototype, "just Physio kidding", based on the principles and architecture of PhysioMate. This involvement brought to the project the possibility of applying a participatory design with the regular participation of the experts' team (see Figure 5.37), and children, as the ultimate end users of the games (see Figure 5.38). We drew on the knowledge of the experts' team to design, and evaluate, "just Physio kidding" throughout its whole development process, in terms of: instructional objectives and strategies best suited for the target-rehabilitation; implementing adequately the set of required movements of the upper-body; designing adequate engaging/motivating game activities for the target-audience; and evaluating the overall effect of gameplay. It was very important to involve these stakeholders in the design process of the system, mainly to have their input towards what could be personalised.

Figure 5.37 - Study EU.3: Coordinator of the experts' team testing PhysioMate.

Figure 5.38 - Study EU.3: Participant M testing the first version of "just Physio kidding".

We conducted a second phase of user tests after having the first version of "just Physio kidding" ready to be used by the patients, who were the children. The prototype had to be in the conditions indicated by the experts' coordinator. That version was much based on PhysioMate and was not personalised. The second user study was important to collect some interaction data and observe how children interacted with the application, mainly its games, but also with menus and screens through natural user interfaces. Moreover, we could also get the evaluation of members of the experts' team.

The user tests were conducted with two different groups of participants:

- G1) 10 children (seven male and three female) aged 5 to 27 ($\bar{x}$ = 10.5; σ = 7.28).
- G2) five "experts" (one male and four female) aged 24 to 33 ($\bar{x}$ = 28.8; σ = 3.12).

In terms of years of experience in the field of rehabilitation, G2's participants presented values ranging from 2 to 11 ($\bar{x}$ = 6.2; σ = 2.64). Most of them was used to work with Nintendo Wii in rehabilitation intervention, mainly using commercial exergaming applications that are not exactly developed for specific therapies. Those applications do not provide any interface for the therapist, in order to supervise what the patients do with the application and analyse exercises data in detail and according to the therapy's metrics. They are not developed with the explicit goal to improve rehabilitation, not complying with the specific needs of physical therapy (Taylor et al., 2011). Furthermore, like in the previous session of user tests, they all stated that patients showed high motivation while using them, which is very important as previously referred. None of them knew about applications similar to PhysioMate/"just Physio kidding".

Regarding the group of children, the tests were conducted on different days, according to their sessions of therapy in the clinic or depending on the availability of their parents, or tutors. They were children with special needs, presenting different levels of diseases and clinical conditions, such as: spinal muscular atrophy (SMA), developmental delay, both global and motor, cerebral palsy (CP) of different levels, leukoencephalopathy, and hemiparesis. We are only making a global characterisation of the participants, as we will not detail here each one of the children and her/his case, nor present the study results per disease/condition, because it is not the focus of this research work. Those detailed results will appear in future articles. However, as an example, Figure 5.38 shows M, who suffers from SMA, participating in the tests while a researcher is observing and her physiotherapist is supporting her due to her condition. The third phase of user tests would be carried out with nine of the children involved in the previous phase (six male and three female), aged 5 to 22 ($\bar{x}$ = 8.7; σ = 5.03). The tests were conducted several weeks after the second phase.

The questionnaire used with the experts, in phase one, was divided into three parts, each one with a set of statements that participants had to rate (using a seven-point Likert-type scale) at the end of the testing phase. The first part (Table 5.23) was used to assess PhysioMate/jPk's usefulness and appropriateness, based on the set of selected primary movements and the creation of the routines, apart from assessing each one of the main applications.

Statements
PhysioMate/jPk's usefulness and appropriateness:
S1. The chosen set of movements is suitable for the intended purpose of rehabilitation.
S2. The creation of routines based on movements is suited to the intended purpose of rehabilitation.
S3. The existence of a challenge game with game mechanics based on the primary movements improves the rehabilitation process.
The NUI application (games) for the patient:
S4. The NUI application (which is composed of the routines and the challenge games) for the patient is suitable for the potential end users.
S5. The design of the patient's NUI application is appealing to the potential end users.
S6. The features of the patient's NUI application are suitable for the potential end users.
The application for the therapist:
S7. The application for the therapist is suitable for the potential end users.
S8. The design of the therapist's application is appealing to the potential end users.
S9. The features of the therapist's application are suitable for the potential end users.

Table 5.23 - Study EU.3: Statements (S1-S9) of the questionnaire applied to expert users.

The second part of the questionnaire had the goal of measuring the global usability of the PhysioMate/jPk system, so it was based on the System Usability Scale (SUS), which is the most used questionnaire for measuring perceptions of usability and it is technology independent. Therefore, we applied the 10-item questionnaire (Table 5.24) with seven response options (thus using a seven-point

Likert-type scale, 1 – strongly disagree; 7 – strongly agree). Participants completed the questionnaire right after having used PhysioMate/jPk and before any debriefing or other discussion, following the advice from (Brooke, 1996).

Statements
P2.S1. I think that I would like to use PhysioMate/jPk frequently.
P2.S2. I found PhysioMate/jPk unnecessarily complex.
P2.S3. I thought PhysioMate/jPk was easy to use.
P2.S4. I think that I would need the support of a technical person to be able to use PhysioMate/jPk.
P2.S5. I found the various functions in PhysioMate/jPk were well integrated.
P2.S6. I thought there was too much inconsistency in PhysioMate/jPk.
P2.S7. I would imagine that most people would learn to use PhysioMate/jPk very quickly.
P2.S8. I found PhysioMate/jPk very cumbersome to use.
P2.S9. I felt very confident using PhysioMate/jPk.
P2.S10. I needed to learn a lot of things before I could get going with PhysioMate/jPk.

Table 5.24 – Study EU.3: SUS statements used in part 2 of the questionnaire applied to expert users.

The third part of the questionnaire was directed to understand how the experts look into the personalisation of PhysioMate/jPk towards its end-users, particularly, the patients (Table 5.25). Once again, we applied the questionnaire using a seven-point Likert-type scale (1 – "strongly disagree"; 7 – "strongly agree").

Statements
P3.S1. The PhysioMate games automatically adapted to the patient profile will be more interesting for the rehabilitation process.
P3.S2. The PhysioMate games can be easily adapted in the following ways:
P3.S2.a. Design of graphics.
P3.S2.b. Messages for the patient.
P3.S2.c. Level of difficulty of the routines.
P3.S2.d. Selection of movements and routines.
P3.S2.e. Challenge game (Recycling Therapy) configuration.

Table 5.25 – Study EU.3: Part 3 of the questionnaire applied to expert users.

Regarding the assessment with the patients in phase 2, since they were children with special needs, our team in agreement with the experts' team thought it would be preferable to be the therapist, who supervised the child's tests session, to apply and fill in a small questionnaire based on part of SUS (see Table 5.26). Therefore, after having everything tested to their satisfaction, the therapist had to fill in the questionnaire based on her/his observations regarding the interaction of the child with the application, a short interview with the child, and the gathered knowledge s/he already had about the child. Before filling it, the therapist would test the application and fill in the questionnaire with nine statements already used in the first part of phase one.

Statements
S1. I think that I would like to use jPk frequently *with the patient.*
S2. I found jPk unnecessarily complex *to the patient.*
S3. I thought jPk was easy to use *with the patient.*
S4. I think that I would need the support of a technical person to be able to use jPk *with the patient.*
S5. I felt very confident using jPk *with the patient.*

Table 5.26 – Study EU.3: SUS-based statements used in the questionnaire applied to patient users by an expert user.

Lastly, the third phase had only the children testing the application, but once again under the supervision and support of the therapist (the expert user). The previous questionnaire was also applied in the third phase. It worked as a reference that would allow assessing if the personalised version added any improvement to the evaluation made by the experts based on the indications from the children. However, as a complement, an additional questionnaire had to be applied in order to evaluate each one of the personalisations designed for "just Physio kidding" (see Table 5.27). Although having participated in the design of the personalisations, we provided the therapist with a tutorial explaining and exemplifying each one of the instances, regarding its goal, resources involved, their combinations to reach the proposed profiles, and the required result. Once again, we applied the questionnaire using a seven-point Likert-type scale (1 – "strongly disagree"; 7 – "strongly agree")

Statements
S.P1 I think the personalisation "motivation" responded well to the patient.
S.P2 I think the personalisation "competition" responded well to the patient.
S.P3 I think the personalisation "pressure" responded well to the patient.

Table 5.27 – Study EU.3: SUS-based statements used in the questionnaire applied to patient users by an expert user.

Results and discussion of phase one

The questionnaire revealed very positive results regarding the users' experience with PhysioMate. Participants really felt that the chosen set of movements was suitable for the intended purpose of rehabilitation ($\bar{x}$ = 5.2; σ = 1.26). Almost 80% of the participants rated S1 with a positive value. Only two participants rated it negatively. The results were even better ($\bar{x}$ = 6.3; σ = 0.96) when they had to rate S2 ("the creation of routines based on movements is suited to the intended purpose of rehabilitation"), with 55.56% of them giving the maximum score of seven. It seemed that only one of them had doubts about the usefulness of creating the routines, at least the way the research and development team designed them.

It is clear that participants saw the existence of a challenge game with its mechanics based on the primary movements has being important to improve the rehabilitation process. The results of S3 are conclusive and leave no room for doubt ($\bar{x}$ = 6.8; σ = 0.43). The panel of experts appreciated and whole-heartedly embraced the idea of having a game based on an everyday life scenario, since one of the major issues they found in rehabilitation is the lack of motivation shown by patients for realising repetitive exercises when they do not see anything related to everyday life tasks. The results of the first three statements proved that the experts really approved the ideas behind PhysioMate, considering it very useful for the desired goal with its appropriate set of primary movements and the creation of routines.

In relation to the NUI-based application for the patients, about a quarter of the participants showed some doubts (27.78% have rated with 4), although none of them has rated negatively the statement "The NUI application (which is composed of the routines and the challenge games) for the patient is suitable for the potential end-users" ($\overline{x}$ = 5.1; σ = 0.96). Participants rated S4 positively (44.44% agreed, 16.67% much agreed and 11.11% strongly agreed). However, the results are not as confident as in the first three statements related to the global project concept since the user will have to interact with the application interfaces using gestures. This issue raises some doubts in the experts since it can be frustrating and demotivating for patients. However, participants rated very positively S6 considering that the main features are suitable for the potential end users ($\overline{x}$ = 5.5; σ = 0.86). One participant rated it with only three, but the others gave positive ratings to S6 with 50% giving 6 out of 7 points. On the other hand, participants evaluated the design of the patient's NUI application as being really appealing to the potential end-users ($\overline{x}$ = 6.2; σ = 0.79). They believe patients will feel very motivated to use PhysioMate since the design is well done, being very interesting.

Participants also evaluated positively the application for the therapist. According to them, this is a very important application since it allows therapists to feel in control of what happens with their patients in the PhysioMate context. Participants considered the application for the therapist suitable for its potential end-users ($\overline{x}$ = 5.2; σ = 1.48), which can be exactly themselves. Only two of them rated negatively S7 with two points and 50% of them rated this statement with six points, which is very good. Moreover, the results of S8 are even better with only one participant rating it with a negative value (3, the highest negative) and two of them having some doubts (rated it with 4) regarding the design of the therapist's application ($\overline{x}$ = 5.7; σ = 1.19). At last, the results obtained with S9 ($\overline{x}$ = 5.2; σ = 1.34) were very close to the results from S7, although presenting a lower standard deviation. All participants except three gave positive ratings to S9, which means that, overall, they agreed with the features that the application offers. The combined result of S7, S8 and S9 proved that participants were very willing to use PhysioMate, particularly the application to supervise their patients.

The SUS scores were calculated for each user (Figure 5.39), according to (Brooke, 1996), with a mean ($\overline{x}$) value of 78.1 and a standard deviation (σ) of 10.5. Despite the wide usage of SUS, there has been little guidance on interpreting SUS scores, but it is known that the average SUS score is a 68 (Sauro, 2011). Therefore, the score obtained with our study is considered above average. Moreover, converting the raw score of 78.1 to a percentile rank through a normalising process (Sauro, 2011) results in a rank around 80%, which means it has higher perceived usability than 80% of all products tested with SUS. It can be interpreted as a grade of a B. A score above 80.3 gets an A and this is also the point where users are more likely to be recommending the product to a friend or colleague (Sauro, 2011). Thus, PhysioMate's score is very near it, which means it can get recommendations easily. Indeed, Figure 5.39 shows that 9 of the participants got scores above 80, which means they are strong candidates to be recommending PhysioMate to others.

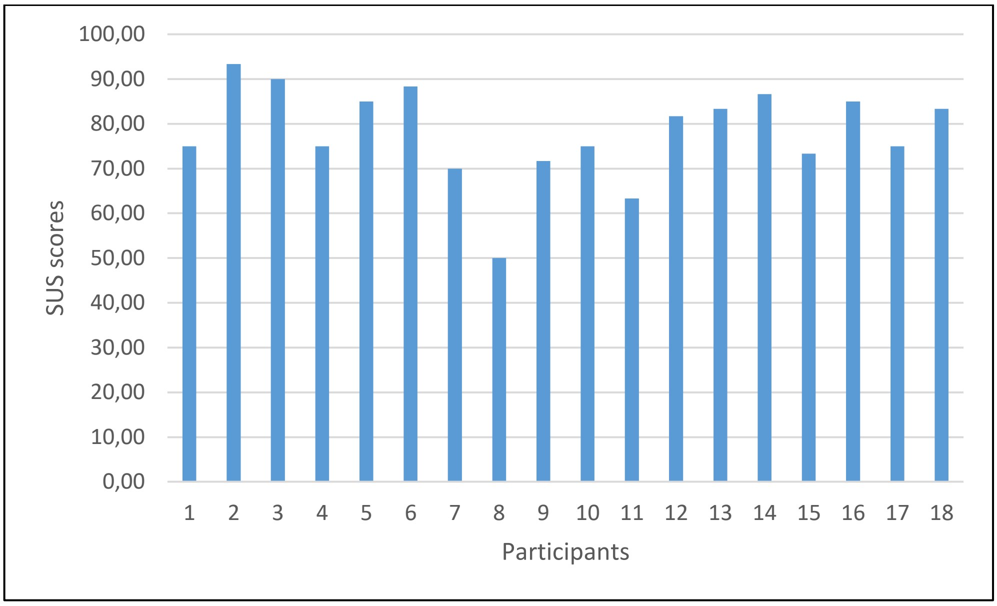


Figure 5.39 – Study EU.3: phase 1's part 2 of questionnaire results (SUS scores) for each participant.

Finally, in the third part of the tests, participants considered that applying the concept of personalisation to the games of PhysioMate will be more interesting for rehabilitation purposes. Thus, one participant rated statement P3.S1 with a 5, two of them rated it with a 6 and the remaining gave a score of seven points, resulting in a $\bar{x}$ of 6.8 and a σ of 0.55. Furthermore, Table 5.28 shows that all the personalisation aspects purposed by us have been very well received by the participants. Particularly, the level of difficulty of the routines ($\bar{x}$ = 6.4; σ = 0.98) and the challenge game configuration ($\bar{x}$ = 6.7; σ = 0.59) were clear winners for future adaptation towards the patients profiles. These overall results demonstrate that the panel of experts are willing, even keen, to use systems such as PhysioMate with personalisation elements applied to them, which can enable dynamically adaptation of therapeutics according to the patient performance, a better in-home rehabilitation procedures and remotely analysis, more efficient progress monitoring and performance feedback.

Statement	1 – strongly disagree	2	3	4	5	6	7 – strongly agree
P3.S2.a	0.0%	0.0%	0.0%	44.4%	16.7%	27.8%	11.1%
P3.S2.b	0.0%	0.0%	0.0%	11.1%	44.4%	27.8%	16.7%
P3.S2.c	0.0%	0.0%	5.6%	0.0%	0.0%	33.3%	61.1%
P3.S2.d	0.0%	0.0%	0.0%	0.0%	33.3%	44.4%	22.2%
P3.S2.e	0.0%	0.0%	0.0%	0.0%	5.6%	22.2%	72.2%

Table 5.28 – Study EU.3: results for the questionnaire's part 3 of phase 1.

Results and discussion of phase two and three

The questionnaire revealed very positive results regarding the evaluation of "just Physio kidding" by the expert's team. Regarding the first three statements (S1: $\bar{x}$ = 5.6; σ = 1.02/ S2: $\bar{x}$ = 6.4; σ = 0.80/ S3: $\bar{x}$ = 6.6; σ = 0.49), they are in line with the results obtained in the first phase, which would be expectable since they thought PhysioMate was interesting to be used in their clinic. It is noteworthy that the experts team' coordinator believed that PhysioMate, from the first interactions with it and after knowing its principles, would work well in the clinic's context. Figure 5.40 illustrates the overall results for S1-S3.

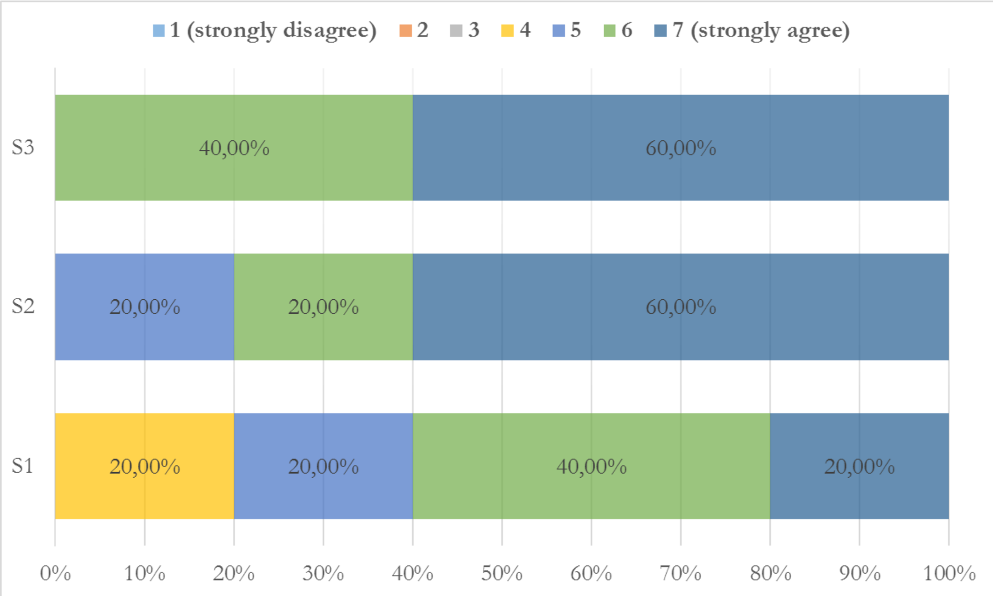


Figure 5.40 - Study EU.3: Summary of phase 2's questionnaire results regarding PhysioMate/jPk's usefulness and appropriateness (S1-S3).

Overall, the next three statements did not receive so good ratings (Figure 5.41). S5 did receive good scores ($\bar{x}$ = 6.4; σ = 0.49) since the design of the patient's NUI application was obtained after meetings with the experts, so they were satisfied with it. However, S4 ($\bar{x}$ = 5.0; σ = 0.89) and S6 ($\bar{x}$ = 5.6; σ = 0.49) generated some doubts on some of the experts since they were expecting to have more interfaces that would not demand much interaction from the patients. By the time of phase 2's testing, the project was still very PhysioMate-based, still lacking voice interaction and simpler and cleaner interfaces, which were then implemented to improve the overall accessibility towards the children with special needs.

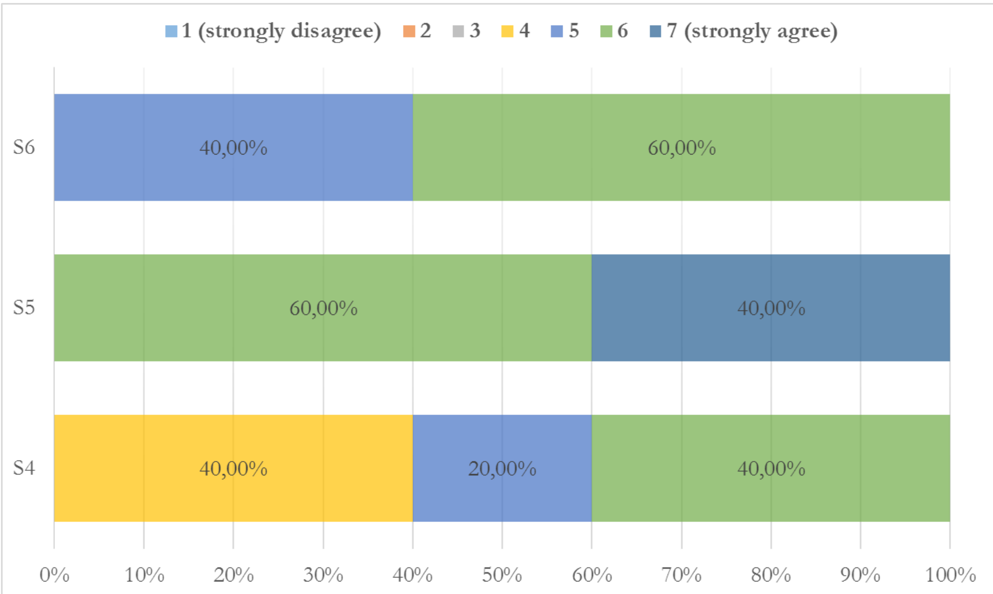


Figure 5.41 - Study EU.3: Summary of phase 2's questionnaire results regarding the NUI application (games) for the patient (S4-S6).

As Figure 5.42 illustrates, the results of the last three statements (S7, S8 and S9) were very positive since we redesigned the PhysioMate's initial application for the therapist according to the requirements of the experts. They needed to feel in control, so it was important to provide them with well-designed features and interfaces for customisation of games (see Figure 5.43), control of personalisation and analysis of data obtained with the games.

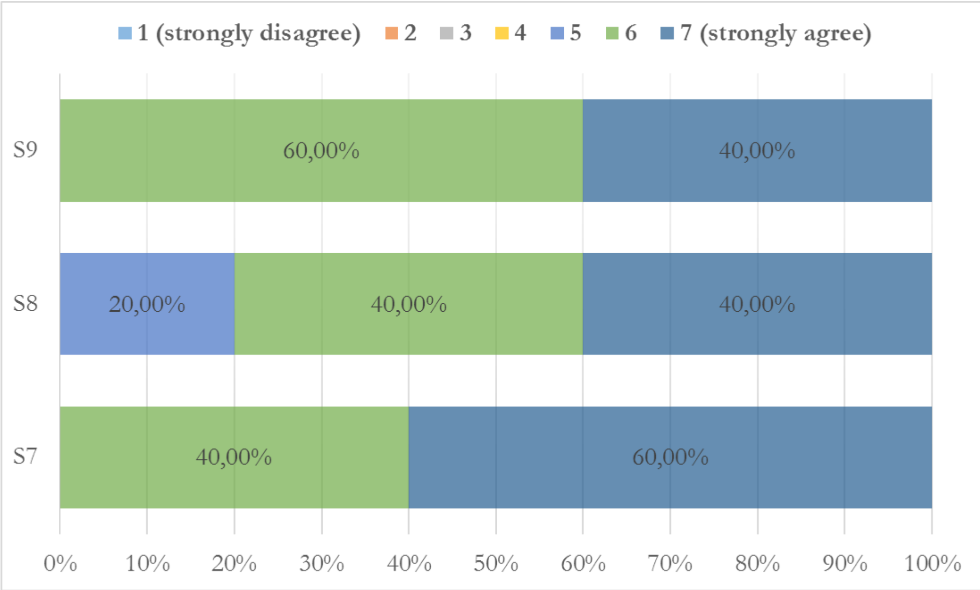


Figure 5.42 – Study EU.3: Summary of phase 2's questionnaire results regarding The application for the therapist (S7-S9).

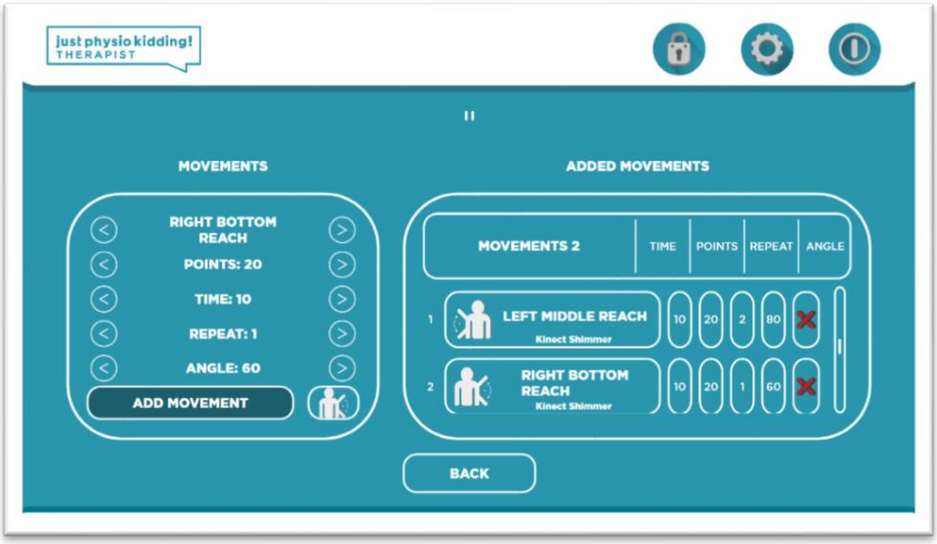


Figure 5.43 – Study EU.3: "just Physio kidding" UI for the configuration of movements in a game.

After the second phase of user tests, we started working on a first personalised version of "just Physio kidding", involving the same team that had been working on the project. We applied the personalisation instances already presented and detailed in section 5.3.2 and designed the third phase of user tests. One important conclusion obtained from the second phase of tests was that the expert's team wanted and needed a mechanism to feel in control over the system and its future personalised features. Therefore, the most recent version of "just Physio kidding" was designed according to the expectations of the experts, giving to them the interfaces and features needed to control personalisation. Figure 5.44 shows a button (Adaptação") that allows the expert to go to another interface where s/he can control personalisation (Figure 5.45).

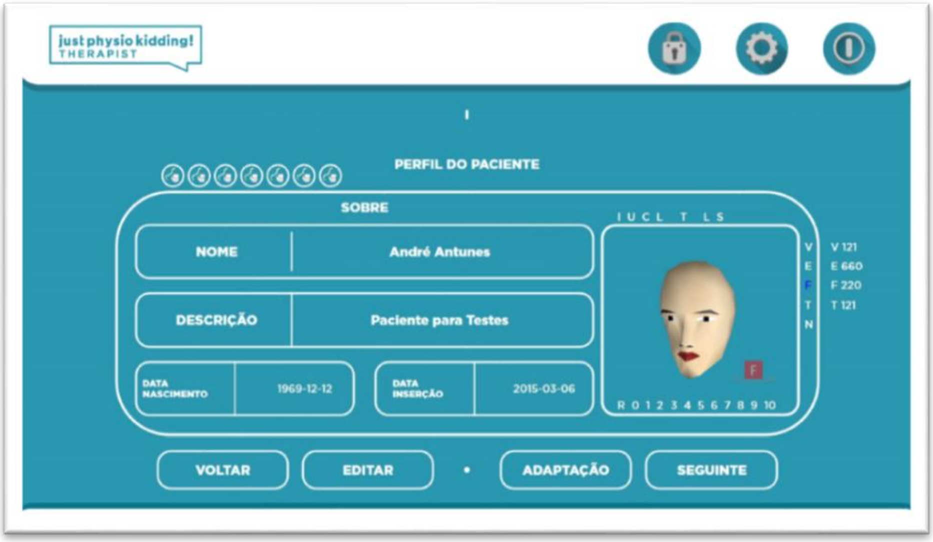


Figure 5.44 - Study EU.3: "just Physio kidding" UI regarding the basic profile of a patient.

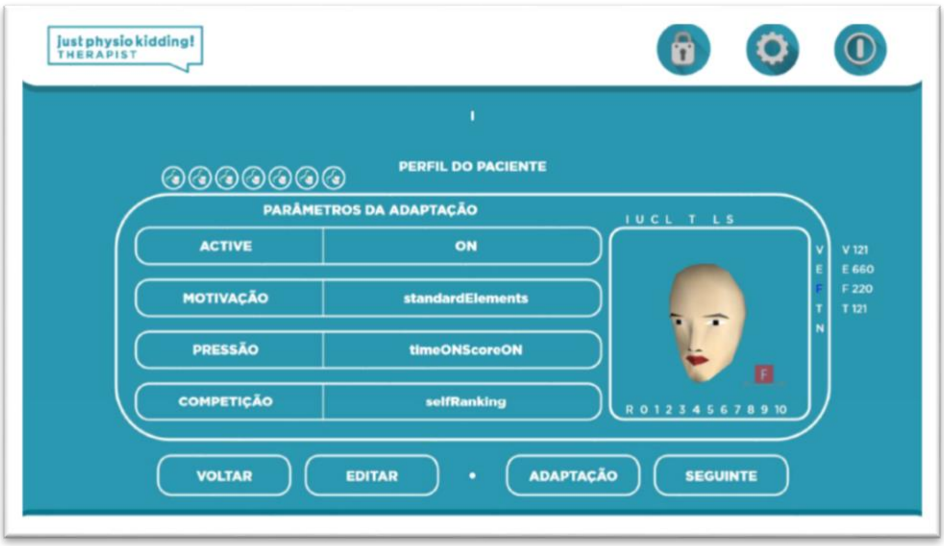


Figure 5.45 - Study EU.3: "just Physio kidding" UI that allows the therapist to control personalisation.

Regarding the testing sessions with children, in both phases, the experts' coordinator tested "just Physio kidding" with the children. Afterwards, s/he filled in the questionnaire of Table 5.26, in both phases, and the questionnaire of Table 5.27, in phase three. The results were positive, but it has to be noted that they depended more on the child condition and pathology than on the general usability of the system, as we knew from the start that the expert had evaluated positively "just Physio kidding". Thus, according to her/him, it was already a valid instrument to be used in therapeutic context with the target-patients of the clinic. By the time of phase two, s/he was evaluating if it could be used to address each particular child. We opted for using only half of the SUS's statements since by the time it was hard to understand how to apply some of the other statements. For the third phase, with the first part of the questionnaire, we wanted to evaluate if the personalised version of "just Physio kidding" did not make "just Physio kidding" more hard to understand and use, in comparison with the previous phase. Therefore, at least, we would like to obtain the same, or similar, results. We ran the risk that the results may be biased since it was the same expert conducting the tests and applying the questionnaire. On the other hand, this way, we had a reference, and from someone knowledgeable of the system, who was very interested in evaluating the system carefully towards each child.

It seems that the results were very similar, perhaps having been a bit better in the third phase. We cannot say they were statistically significant, but being similar was enough in order to conclude that personalisation was not working the other way around, turning "just Physio kidding" more complex and hard to use. It was what we all wanted to evaluate by that time. The global results by child per phase are illustrated in Table 5.29 and Table 5.30.

child	S1	S2	S3	S4	S5
c1	6	2	6	2	6
c2	3	6	3	3	6
c3	6	2	6	1	6
c4	7	1	7	1	7
c5	7	2	7	1	6
c6	7	3	5	2	7
c7	5	5	3	2	3
c8	5	7	2	6	2
c9	4	5	4	2	5
c10	7	1	7	2	7
mean	**5,7**	**3,4**	**5**	**2,2**	**5,5**
std.dev	**1,35**	**2,06**	**1,79**	**1,40**	**1,63**

Table 5.29 - Study EU.3: Phase 2's questionnaire results regarding the tests with the patients ("just Physio kidding" without personalisation).

child	S1	S2	S3	S4	S5
c1	7	2	6	2	7
c2	4	6	3	3	6
c3	7	2	7	1	7
c4	7	1	7	1	6
c5	6	2	7	2	7
c6	7	3	5	2	7
c7	4	5	3	3	4
c8	6	7	2	6	4
c9	3	6	4	1	5
mean	**5,7**	**3,8**	**4,9**	**2,3**	**5,9**
std.dev	**1,49**	**2,10**	**1,85**	**1,49**	**1,20**

Table 5.30 - Study EU.3: Phase 3's questionnaire results regarding the tests with the patients ("just Physio kidding" with personalisation).

In phase two, the expert gave positive scores to using the system with participant c10, who did not participate in phase three. The expert had some difficulties in using the system with participant c2, regardless the phase. Moreover, participants c7, c8 and c9, are clear cases that should be carefully studied and evaluated, shortly, to find out if applying personalisation will make "just Physio kidding" more suitable to them. We suppose that working on different personalisations will help to accomplish that purpose. On the other hand, Figure 5.46 shows how participant c4 (D, with developmental delay) was very enthusiastic about interacting with the new personalised version of "just Physio kidding" and Figure 5.47 shows participant c8 (Ti, with a CP of level IV) interacting with an adapted interface (background and some elements) towards his impaired vision.

In all cases, the physiotherapist (an expert user) was supervising and supporting the child, whenever he needed it and according to his limitations. It must be noted that a critical requirement in the implementation of "just Physio kidding" was to allow the therapist to be behind the patient without interfering with the tracking system and performance of the application. The results were very positive since in all cases the therapist could be in that position without interfering, which allowed her to control the child's movements while supporting her/him whenever necessary.

Figure 5.46 - Study EU.3: Participant T testing a personalised version of "just Physio kidding".

Figure 5.47 - Study EU.3: Participant Ti testing a personalised version of "just Physio kidding".

Finally, regarding the last three statements applied to children in phase three, the results were not conclusive, although being, apparently, confident and interesting (Table 5.31). We need to run more tests and create different groups of end users regarding patients' state, their functional condition and pathology.

child	S.P1	S.P2	S.P3
c1	6	6	6
c2	3	4	3
c3	6	7	7
c4	6	7	7
c5	6	7	7
c6	7	3	5
c7	3	3	3
c8	3	4	4
c9	3	4	4
mean	**4,8**	**5,0**	**5,1**
std.dev	**1,62**	**1,63**	**1,59**

Table 5.31 - Study EU.3: Phase 3's questionnaire results regarding the last three statements ("just Physio kidding" with personalisation).

5.3.4. Final Remarks

The tests' results were positive, and the process of studying and applying personalisation to a system like PhysioMate/jPk demonstrated how interesting and relevant is the involvement of the experts in the domain of the application. Once again, similarly to the two previous studies, this third study proved how important is to have a first version of the application without personalisation in order to collect data about the users to better design the best approach for personalising the application. A first step to take is to know, understand, the end users, mainly the most representative ones, thus, the need to incorporate personas in the design process.

Personalisation is essential to pervasive healthcare environments, such as the one implemented by PhysioMate/"just Physio kidding", which focuses on a human-centred paradigm aiming to provide adaptive and personalised services to the users, according to the context. Furthermore, the results proved that the personalisation of a few elements of "just Physio kidding" improved the global satisfaction of the patients and their therapists, engaging them even more in the game, so in rehabilitation, letting the system support them in a task that no longer needs to be tedious.

Moreover, this study with the expert users allowed us to confirm that they still want and need a mechanism to feel in control over the system and its personalised features. It confirmed the findings of study EU.1 regarding the importance of addressing users' privacy concerns adequately to increase their trust, retaining some user control over the experience. Therefore, the most recent version of "just Physio kidding" was designed according to the expectations of the experts, giving to them the interfaces and features needed to control personalisation. If personalisation gives control to the application, customisation gives control to the user. Therefore, we implemented both features in the prototype trying to design their integration carefully (Schade, 2016a). "just Physio kidding" was the first of the prototypes, presented in this dissertation, clearly implementing the two features, which was also an explicit requirement to fill a need of the experts. We needed to achieve a prototype with both customisation and personalisation features since both can enhance users' experience.

A shortcoming of the current study is the lack of a thorough user experience evaluation with the personalised solution in the wild. We wanted to leave the application running in the clinic for several weeks, but it is something that must be very well prepared and planned, in order to respond effectively to the goals of the users, mainly the therapists. This thorough evaluation would let us find out if the application and, mainly, the personalisation feature contributed successfully to the patients' progression in the rehabilitation, reducing their clinical condition severity. The process is now poised to move forward, but it has not come in time to be included in this dissertation. Experience of others confirms that experimentations in the wild are complex to setup and anticipation of the technical problems is compulsory (Jambon and Meillon, 2009), which takes a lot of time and is not advisable when we have tight plans.

Regarding improvements made to the solution, in Study EU.1, we had the personalisation options issue, in which the number of possible combinations would grow exponentially when dealing with many parameters (e.g, FCT4U's `viewOrder` personalisation). We solved that issue for this Study EU.3 since we did not need anymore to specify all the parameters for a specific personalisation option that only needs to be based on an option of some of them, regardless the options of the remaining. For instance, the personalisation option `timeONScoreON` in personalisation "`badges are bonus`" (see appendix A.5). This cut the number of personalisation options needed to cover the combinations of parameters that a developer need to use in a personalisation instance.

Lastly, this study allowed testing the whole solution's performance and viability, for one last time with end users and in response to another realistic scenario.

5.4. Summary

This chapter presented the conducted studies with end users in order to respond to the research questions raised previously. We have been involved in the development of different applications that worked as proof of concept prototypes in the studies throughout the different phases of the evaluation process with end users, which were the following:

- Study EU.1 (section 5.1) – This second study started right after the conclusion of the deployment of P²MUCA. This time, we wanted to assess if the system could actually work, building appropriated user profiles based on real end users interacting with an Ubicomp installation like FCT4U.

- Study EU.2 (section 5.2) – The main purpose of this study was to respond to **RQ_C**. Therefore, it was designed to assess if the user interaction with a first application can be useful to personalise the experience of the same user when interacting with a second application. It was conducted after the evaluation of P^2MUCA, in the previous study, since we had to be sure the platform was fully working, responding well to all requests.

- Study EU.3 (section 5.3) – As in the two previous studies, we wanted to find out if personalisation would add real value to human-computer interaction to engage better the end users (patients, in this case) in the use of the applications. All the previous studies were designed to involve our research team, prototypes' developers, and the participants (developers or end users) in the tests. But at this stage, we wanted to go one step forward and take another approach, involving end users in the design process.

The research questions **RQ_A**, **RQ_B** and **RQ_D** are responded once again, but this time reinforced by the results obtained with the end users tests. Those questions were already responded directly through the studies with developers, but in this chapter we obtained the end users perspective. Having an improvement in the end user experience, it means the platform and the model worked positively, facilitating the developers' task in the implementation of personalised applications. The research question **RQ_C** was fully responded with the conduction of Study EU.2.

6. Conclusions and Future Work

"There is nothing more precious than self-trust."

(Yogi Teas, 2017)

This dissertation aimed to expand the boundaries of the research on generic modelling of personalisation to enhance Human-Computer Interaction across applications in an increasingly Ubiquitous Computing world. This final chapter presents a summary of the developed work (in section 6.1) and its main achievements (in section 6.2) as well as questions and directions for further developments (in section 6.3).

6.1. Synthesis

Marc Weiser had the vision of a World much based on the deployment of rich computational infrastructures that, automatically and unobtrusively, support and facilitate the everyday life of people. This concept of a Ubicomp-based World is nowadays especially supported by mobile and context-aware computing. All the information becomes accessible from just about anywhere, at any time, and it should be delivered in a manner appropriate to the people's context. The use of mobile devices makes this human-centred vision even more personal. The Ubicomp paradigm aims to provide the right information and services in context to the end users of its environments and applications, according to their needs and wants. Consequently, it demands personalisation and ubiquitous user modelling approaches.

However, the process of choosing the appropriate HCI, composed of content, interfaces, or services, for each user is still a critical challenge in many applications of Ubicomp scenarios. The challenge is critical since the process must consider what to offer, which is the composition of the final HCI, and the aspects, features, and conditions, which are important to know who the user is in an application's context of use. Therefore, the offer of the right personalised user experience is a challenge due to issues, such as user interests, heterogeneous environments and devices, dynamic user behaviour and user privacy. Some important specific challenges are related to inferring relevant information about a user, aggregating and integrating such information effectively into long-term user models while achieving user model interoperability, and providing pervasive information access in a personalised manner.

In order to help to cope with the aforementioned limitations, this research work proposed a solution to support developers in the personalisation process of the human-computer interaction across Ubicomp applications. **The main goal was the deployment of a solution driven by a generic and interoperable model, based on a cloud-based platform providing the tools, services, and partners to help developers in the implementation of personalised solutions.** Moreover, the solution allows end users to be less bothered when they start to use a new application that needs "to know them", as their previous interactions with other applications might be useful to personalise their experience, mainly, when they start interacting with a new one. This scenario is possible provided that both applications "talk" the same language due to the use of the cloud-based pervasive model that promotes interoperability, guiding developers in the implementation of both solutions.

Therefore, our solution is driven by the proposal of the generic model (X-Users) and an XML-based description language (XUPIL) was designed to be the standard format for the definition of the rules specifying the instantiation of Ubicomp applications' personalisations. Another important component in the solution is the modelling framework Context-Aware based Personalisation Environment (CAPE), which encapsulates the X-Users model and integrates three main modules: Web services interface, personalisation API, and user profiling module. CAPE, and consequently the model, is provided by a cloud-based platform (P^2MUCA) to support developers in the implementation of personalised solutions, considering interoperability and different contexts of use, addressing dynamic user behaviour and user privacy, and aiming a high usability level. Figure 6.1 illustrates how the concept was built from the lower to the upper layer (or built from the inside out), with the model driving the solution, being the initial and central component, encapsulated by CAPE, which was later on integrated by P^2MUCA.

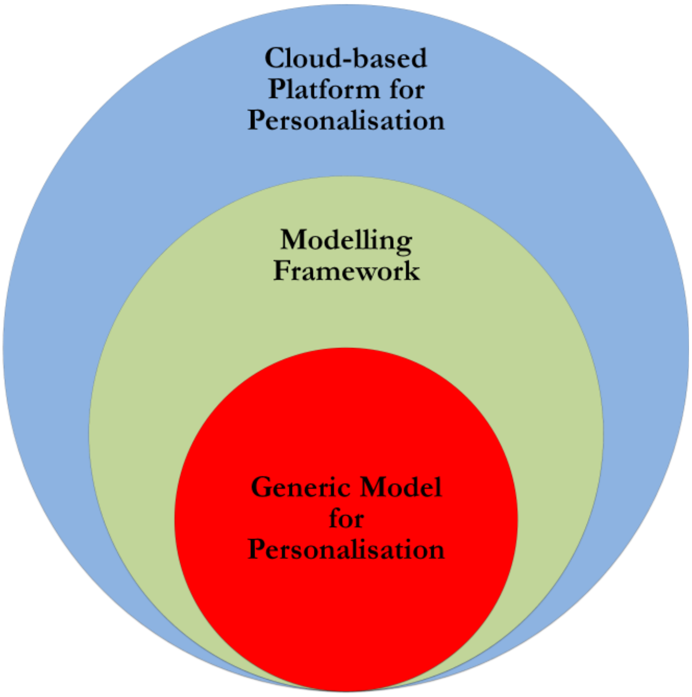


Figure 6.1 - A model-driven personalisation solution built as a stack of sub-solutions.

We conducted different studies that allowed us to evaluate the various components of the solution, separately or as a whole. We also involved end users, developers and domain's experts in the studies, gathering important knowledge both for further improvement of the solution and design of personalised applications. In order to support the studies, we were involved in the development of several Ubicomp applications, which proved their usefulness in demonstrating the design of different personalisation approaches.

On the one hand, the implementation of the solution, detailing all the components, already responded to the research questions. However, on the other hand, we needed validation by others supporting what we supposed to be clear and evident. It was not enough, so the conducted studies were very important to validate the gathered knowledge with the solution implementation, responding to the research questions raised in the first chapter. The first question $\mathbf{RQ_A}$ ("*Is it possible to have a generic model to be followed in the personalisation of different applications, or systems, across different domains in a Ubicomp based world?*") was responded by the implementation of different prototypes, having different goals and being from different domains. For instance, FCT4U combined private mobile interfaces and public situated displays that needed to be personalised, while WeSync was a second screen app that needed an adaptation to TV delays based on the user behaviour. In both cases, we conducted studies with end users collecting positive results, as the users noticed that the applications automatically adapted to their preferences and behaviours, which demonstrated the model was successfully followed in the design of different personalisation approaches.

Regarding the question **RQ_B** ("*How can user modelling and context modelling be integrated into a generic pervasive personalisation model to facilitate the implementation of personalised Ubicomp applications?*"), the definition of the model showed how the context modelling could be integrated in the general model. Moreover, the study conducted with FCT4U also demonstrated that the personalisation approach designed for this particular case had success in the integration of context, which was used both as an external service post user profiling and to segment data used in the user profiling algorithm. The way it is integrated in the model facilitates its use by the developers of applications. If, on the one hand, the positive results of FCT4U with end users showed the context integration worked well, on the other hand, the developer tests with the Walker prototype showed that developers could understand that integration. The developers involved in the study responded successfully to the proposed tasks, which included a personalisation based on context.

As for **RQ_C** ("*Is the user interaction with an application useful to personalise the experience of the same user when interacting with another application?*"), the personalisation applied in the study conducted with WeSync and WePlayTennis worked smoothly. The results were very interesting and promising since they showed that, with the use of P^2MUCA, we could have different applications simultaneously using the same knowledge about the end users in order to be automatically personalised from the beginning towards them. P^2MUCA provided the ability for both applications to exchange data safely, or the interoperability and data sharing feature.

Lastly, the research question **RQ_D** ("*Can a cloud-based solution be easy to use and be developer-friendly in the support of the rapid implementation of generic personalisation across different Ubicomp applications?*") was responded through the last conducted study (D.2), mainly with the second part involving what we considered the expert developers regarding the use of our solution. The first study with developers already gave us positive results, but only regarding the ease of understanding the concept/model and the ease of using XUPIL to define personalisations. The results promised an easy to use and developer-friendly solution. Study D.2 confirmed those results from study D.1 regarding the model and its language, but they also showed us that P^2MUCA, as a full solution for personalisation, is interesting, appropriate and easy to use by developers, being friendly, as far as its complexity allows it to be, in supporting the personalisation of applications. However, we cannot guarantee that it allows for a rapid implementation of personalisation, since we do not have benchmarks metrics, nor concrete data, that allows us to state that interesting characteristic. Part of the results of study D.2 indicated it might be possible as developers used XUPIL to design personalisations without considerable effort and in a way that we can consider fast. However, we do not have terms of comparison, so it is mainly by empirical evidence. The part of the study with the experts supports this evidence as we observed the whole process of applying personalisation to the prototypes and our experience, as developers and programmers, indicates us we can consider our solution as being able to support a rapid implementation of personalised solutions. With the participation of those prototypes' developers, which we called experts, we obtained important insight from those who have used the solution in the development of full-functional personalised prototypes for several weeks, or months, in two particular cases.

Summing up the different progresses that we have presented, we can state that we have accomplished our research goal ("*In order to provide developers with a set of easy to use and user-friendly guidelines and tools for* *applying personalisation to a wide spectrum of applications, we propose a model-driven and cloud-based platform targeted to the rapid development of personalised Ubicomp applications*"). The studies demonstrated the viability of our solution for the personalisation of the human-computer interaction across Ubicomp applications, even if from different domains.

It is important to note that we must make good decisions as to when to reach into the personalisation toolbox (Kramer et al., 2000), which is P²MUCA in the particular case of this research. It will help ensure the success of the personalised applications when it is done with the unwavering focus on delivering value to the end users (Kramer et al., 2000), but carefully testing, measuring, and iterating on the designs, as it has been done with the "just Physio kidding" prototype in study EU.3. Intelligibility is a particular usability aspect of interest, as it should help users to form an accurate mental model about how to use an application (Anind K. Dey in (Ferscha, 2012)). End users must understand how an application works, being able to predict what it will present and do in the future since all of this will influence the adoption and use of a personalised solution.

Finally, observing the path taken since the early days of initial ideas and prospects to the obtained results, it is rewarding to verify that we have fulfilled out our main objectives. With the novel aspects brought to the personalisation modelling made available through a cloud-based platform and the different contributions to the topic, we finish this dissertation with a sense of gratification.

6.2. Achievements

At the end, our main achievements can be summed up as follows:

- **Definition of X-Users** (in section 3.2), which is the generic personalisation model, integrating user and context sub-models, to guide and facilitate the implementation of personalised Ubicomp applications;
- **Definition of XUPIL** (in section 3.2.2), which is the XML-based Domain-Specific Language (DSL) providing the X-Users-based format for the definition of the rules that specify the instantiation of the application's personalisations;
- **Creation of CAPE** (in section 3.3), which is the X-Users-driven personalisation modelling framework in the form of personalisation APIs and configuration modules;
- **Implementation of P²MUCA** (in section 3.4), which is a fully functional cloud-based platform that supports and helps developers in the implementation of personalised Ubicomp applications, allowing X-Users to be effectively pervasive and interoperable; P²MUCA[16] can already be accessed by any potential user, mainly developers, to test it and use it freely (however, they will need our guidance since we did not provide appropriate manuals).
- **Implementation of personalised prototypes**, following X-Users and using P²MUCA, which worked as a proof of the feasibility of the whole solution, but at different levels of involvement and with different objectives, namely:
    - from the beginning to the end of the entire development cycle of:
        - **LEY** (section 4.1.1) and **FCT4U** (section 5.1.1) - mainly to conduct lab experiments with developers and end users, but with strong possibilities for continuity;

[16] P²MUCA: http://website-p2muca.rhcloud.com/

    - **PhysioMate/"just Physio kidding"** (section 5.3.1) – to involve different stakeholders, mainly expert users, in the personalisation design and will be used soon in the wild by their end users;
  - in the design and implementation of the personalised version of:
    - **Walker** (section 4.2.1) – to conduct lab experiments with developers using a third party application;
    - **WeSync/WePlayTennis** (section 5.2.1) – to demonstrate the interoperability of the solution and a different way of applying personalisation, in third party applications and deploying a concept that can be used by other applications.
- **Conduction of successful developers studies** (chapter 4) with experiments in the lab that demonstrate the feasibility of applying approaches that usually are identified as being used to test the user experience in the HCI field.
- **Conduction of human factors studies** (chapter 5) in the design of personalised solutions for different domains.

6.3. Further Developments

The achievements presented in the previous section represent a stable state in our research. Still, the accomplished results open new doors for additional investigation, which can advance in following directions:

- **Implementing alternatives for the algorithm of the user profiling module.** Currently, user profiles are generated using the k-means clustering algorithm, which works well under certain conditions, but not so well in other situations. If we do not have enough initial data to build each one of the profiles defined for a personalisation instance then the results might not be the best ones. In the case of the study with WeSync, we had success because we collected enough users' interaction data with the first version of the application to feed the profiles, creating the initial needed clusters. On the other hand, with "just Physio kidding", we created dummy users to have the clusters well defined from the beginning. XUPIL provides the element `type` under a personalisation definition, which should be used by the developer to indicate which user profiling technique should be used in a particular personalisation instance, so we will implement alternatives to k-means. In the future, it may be possible to provide a platform feature that allows developers to submit their own algorithms.
- **Adding a method to insert initial data corresponding to the specified personas.** This point is related to the previous one, as it would be an asset to cope with the cold-start problem, automatically generating those dummy users. This would avoid the lack of historical information that occurs when the application still does not have information about the users, consequently leading to a potential situation in which the personalisation will not work properly in the beginning.
- **Improving the expressiveness of the personalisation options definition in XUPIL.** In Study EU.1, we had the personalisation options issue, in which the number of possible combinations would grow exponentially when dealing with many parameters (e.g, FCT4U's viewOrder personalisation). We solved that issue for Study EU.3, for instance, the personalisation option `timeONScoreON` in personalisation "`badges are bonus`" illustrates the

improvement (see appendix A.5). However, we still have some limitations in the expressiveness of XUPIL, since, in some cases, we need to include two or more tags of the element `personalization_option` for the same option. Thus, we need to revise this question to offer a solution in which the number of `personalization_option` tags used is equal to the number of options for the personalisation instance. For instance, in "just Physio kidding" personalisation, we have both the options `selfRanking` and `gameResult` in the personalisation "`competition`" with two entrances, but in terms of expressiveness would be better to have only one.

- **Performing platform scalability tests.** The scalability issue was not under the scope of this research work. It did not concern us, because we did not intend to perform studies with many users. However, if we want to make the platform available to be used by as many applications as possible, then it might be an important issue. It still needs to be stress tested with a (very) large number of simultaneous users to see how well it scales. Some synthetics tests that stressed CAPE directly were not a problem, even though it took a few seconds to process 1000 requests. Nevertheless, the ideal scenario would be to put some of the prototypes into a near-production environment, along with a few other applications at the faculty, to test how P²MUCA would respond under load. So, we should perform load and stress tests on our solution using a scalable cloud-based tool.

- **Changing the deployment of the application.** Following the previous point, it would be important to change the deployment of the application on OpenShift from a non-scaled application to a scaled one, thus allowing to have multiple servlet instances serving the requests and removing the need of a SSH tunnel that connects the gears running P²MUCAWebsite and P²MUCA Service. If it is the database that proves to be the bottleneck, other database management systems (e.g., PostgreSQL) or database clustering and replication solutions can be used.

- **Providing documentation to third-party developers.** Several documents regarding XUPIL, the HTTP-based API, and the platform, should be added to the website so that third-party developers are able to understand and use the solution to its fullest.

- **Redesigning the P²MUCA's website.** The platform needs a better-looking website, with a more modern, organised and appealing design.

- **Becoming a commercial solution.** Eventually, the P²MUCA may even become a commercial product if there is a demand for it. However, first, we must ensure trustworthiness and privacy, as there is an obvious challenge regarding the privacy of the collected information. Users have to trust in the platform, since they may ask who is to store all this data, for how long, where, and what is it used for. (Mark Langheinrich in (Ferscha, 2012))

- **Conducting thorough user experience evaluation.** It would be important to conduct thorough users testing with personalised applications in the wild. User evaluation in the wild poses several difficult challenges, but they are surmountable and this kind of field research can be a crucial component of evaluation (Kellar et al., 2005). Therefore, a thorough evaluation would let us find out if the application and, mainly, the personalisation feature contributed successfully to the everyday life of the end users, depending on the application domain and context of use. For instance, we wanted to leave the "just Physio kidding" system running in the clinic for several weeks, but it was something to prepare and plan very carefully, in order to respond effectively to the goals of the therapists regarding the rehabilitation of their patients.

- **Studying personalised mobile applications to improve therapy.** There is an emerging consumer-driven demand for a more personalised health system and there is no question that the rapid evolution of the mobile apps market became an important driver for personalization in the health field. We have been involved in the project MAiThE (Madeira, Macedo, et al., 2017), which focuses on the deployment and study of personalised mobile Health (mHealth) apps to provide patients and caregivers with self-management capabilities that make them feel empowered in their ability to find strategies, in a more informed and collaborative way, and to optimise therapy outside the clinical context, with remote support from health practitioners. Three mHealth applications, which are BroiStu (Demarin et al., 2015; Madeira et al., 2013), Super-Fon (Madeira, Patrícia Macedo, Reis, et al., 2014; Madeira, Mestre, et al., 2017), and OnParkinson (Pereira et al., 2015; Pereira et al., 2016) will be used to study how personalisation should be designed and integrated to improve the user experience in mobile solutions. We will conduct thorough user experience evaluation in the wild, using the personalised versions of the applications. We will capitalise on the findings of study EU.3 and P^{2}MUCA is already being used to apply personalisation to these three applications. The insight gathered with the development and assessment of the apps tailored to the end users' needs will result in a conceptual model to guide in the development of future mHealth apps. The project will produce an impact study based on thorough apps evaluations conducted on the field with participants from different regions.

References

Abeele, Veronika Vanden, Geurts, Luc, Husson, Jelle, Windey, Frederik, Annema, Jan Henk, Verstraete, Mathijs, and Desmet, Stef. 2010. Designing Slow Fun! Physical Therapy Games to Remedy the Negative Consequences of Spasticity. In *Proceedings of the 3rd International Conference on Fun and Games*. 1–2.

Abowd, Gregory D. 2012. What next, Ubicomp? Celebrating an Intellectual Disappearing Act. In *Proc. of the 2012 ACM Conference on Ubiquitous Computing - UbiComp '12*. New York, New York, USA: ACM Press, 31–40. DOI:http://dx.doi.org/10.1145/2370216.2370222

Abowd, Gregory D., Dey, Anind K., Brown, Peter J., Davies, Nigel, Smith, Mark, and Steggles, Pete. 1999. Towards a Better Understanding of Context and Context-Awareness. In *Proc. of 1st international symposium on Handheld and Ubiquitous Computing (HUC '99)*. Springer-Verlag, 304–307.

Adlin, Tamara, and Pruitt, John. 2010. Chapter 1 – What are personas? In *The Essential Persona Lifecycle: Your Guide to Building and Using Personas*. Morgan Kaufmann, 1–5. DOI:http://dx.doi.org/10.1016/B978-0-12-381418-0.00001-2

Adomavicius, Gediminas, and Tuzhilin, Alexander. 2010. Context-aware Recommender Systems. In Francesco Ricci, Lior Rokach, Bracha Shapira, & Paul B. Kantor, eds. *Recommender Systems Handbook*. New York, New York, USA: Springer US, 217–253. DOI:http://dx.doi.org/10.1145/1454008.1454068

Al-bar, A., and Wakeman, I. 2001. A survey of adaptive applications in mobile computing. In *Proc. of the 21st International Conference on Distributed Computing Systems Workshops*. IEEE Comput. Soc, 246–251. DOI:http://dx.doi.org/10.1109/CDCS.2001.918713

Alankus, Gazihan, Lazar, Amanda, May, Matt, and Kelleher, Caitlin. 2010. Towards customizable games for stroke rehabilitation. In *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*. New York, New York, USA: ACM Press, 2113. DOI:http://dx.doi.org/10.1145/1753326.1753649

Alayed, Hashem, Frangoudes, Fotos, and Neuman, Clifford. 2013. Behavioral-based cheating detection in online first person shooters using machine learning techniques. In *Proc. of IEEE Conference on Computational Inteligence in Games (CIG'13)*. IEEE, 1–8. DOI:http://dx.doi.org/10.1109/CIG.2013.6633617

Allen, James Frederick. 1979. A plan-based approach to speech act recognition. In *Technical Report 131/79*. Dept. of Computer Science, University of Toronto, Canada.

Alt, Florian, Memarovic, Nemanja, Elhart, Ivan, Bial, Dominik, Schmidt, Albrecht, Langheinrich, Marc, Harboe, Gunnar, Huang, Elaine, and Scipioni, Marcello P. 2011. Designing Shared Public Display Networks – Implications from Today's Paper-Based Notice Areas. In *Proc. of 9th International Conference Pervasive 2011*. San Francisco, USA: Springer Berlin Heidelberg, 258–275. DOI:http://dx.doi.org/10.1007/978-3-642-21726-5_17

Annema, Jan-Henk, Verstraete, Mathijs, Vanden Abeele, Vero, Desmet, Stef, and Geerts, David. 2010. Videogames in therapy. In *Proceedings of the 3rd International Conference on Fun and Games - Fun and Games '10*. New York, New York, USA: ACM Press, 94–98. DOI:http://dx.doi.org/10.1145/1823818.1823828

Assad, Mark, Carmichael, David J., Kay, Judy, and Kummerfeld, Bob. 2007. PersonisAD: distributed, active, scrutable model framework for context-aware services. In *Proc. of the 5th international conference on Pervasive computing (PERVASIVE'07)*. Springer-Verlag, 55–72.

Ayodele, Taiwo Oladipupo. 2010. Types of Machine Learning Algorithms. In Yagang Zhang, ed. *New Advances in Machine Learning*. InTech, 19–48.

Ballagas, Rafael, Borchers, Jan, Rohs, Michael, and Sheridan, Jennifer G. 2006. The Smart Phone: A Ubiquitous Input Device. *IEEE Pervasive Computing* 5, 1 (January 2006), 70–77.

DOI:http://dx.doi.org/10.1109/MPRV.2006.18

Bang, Magnus, Gustafsson, Anton, and Katzeff, Cecilia. 2007. Promoting New Patterns in Household Energy Consumption with Pervasive Learning Games. In *Proc. of Persuasive'07*. Springer Berlin Heidelberg, 55–63. DOI:http://dx.doi.org/10.1007/978-3-540-77006-0_7

Bang, Magnus, Svahn, Mattias, and Gustafsson, Anton. 2009. Persuasive design of a mobile energy conservation game with direct feedback and social cues. In *Proc. of the 2009 DiGRA International Conference: Breaking New Ground: Innovation in Games, Play, Practice and Theory*. Brunel University.

Barišić, Ankica, Amaral, Vasco, Goulão, Miguel, and Barroca, Bruno. 2011. Quality in use of domain-specific languages: a case study. In *Proceedings of the 3rd ACM SIGPLAN workshop on Evaluation and usability of programming languages and tools - PLATEAU '11*. New York, New York, USA: ACM Press, 65–72. DOI:http://dx.doi.org/10.1145/2089155.2089170

Barkhuus, Louise. 2003. Context Information vs. Sensor Information: A Model for Categorizing Context in Context-Aware Mobile Computing. In *Proc. of Collaborative Technologies and Systems (CTS'03)*. San Diego.

Berkovsky, Shlomo. 2005. Ubiquitous User Modeling in Recommender Systems Liliana Ardissono, Paul Brna, & Antonija Mitrovic, eds. *User Modeling 2005* 3538 (2005), 496–498. DOI:http://dx.doi.org/10.1007/11527886_68

Berkovsky, Shlomo, Bohnert, Fabian, Carmagnola, Francesca, Cheng, Doreen, Heckmann, Dominikus, Kuflik, Tsvi, Nurmi, Petteri, and Partridge, Kurt. 2010. Introduction of Pervasive User Modeling and Personalization (PUMP'10). In *Proc. of PUMP workshop at UMAP'10*. Big Island of Hawaii: Springer-Verlag.

Berkovsky, Shlomo, Kuflik, Tsvi, and Ricci, Francesco. 2007. Mediation of user models for enhanced personalization in recommender systems. *User Modeling and User-Adapted Interaction* 18, 3 (November 2007), 245–286. DOI:http://dx.doi.org/10.1007/s11257-007-9042-9

Bökemeier, Jost, and Koerber, Jon. 2013. PJB - PHP/Java Bridge. (2013). Retrieved December 30, 2016 from http://php-java-bridge.sourceforge.net/pjb/index.php

Booth, David, Haas, Hugo, McCabe, Francis, Newcomer, Eric, Champion, Michael, Ferris, Chris, and Orchard, David. 2004. Web Services Architecture. (2004). https://www.w3.org/TR/ws-arch/

Bouzeghoub, Mokrane, and Kostadinov, Dimitre. 2007. *Data personalization: a taxonomy of user profiles knowledge and a profile management tool*,

Brooke, John. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.

Brooke, John. 2013. SUS: a retrospective. *Journal of Usability Studies* 8, 2 (2013), 29–40.

Brusilovsky, Peter, and Maybury, Mark T. 2002. From Adaptive Hypermedia to the Adaptive Web. *Communications of the ACM* 45, 5 (May 2002), 30–33. DOI:http://dx.doi.org/10.1145/506218.506239

Burke, J.W., McNeill, M.D.J., Charles, D.K., Morrow, P.J., Crosbie, J.H., and McDonough, S.M. 2009. Optimising engagement for stroke rehabilitation using serious games. *The Visual Computer* 25, 12 (December 2009), 1085–1099. DOI:http://dx.doi.org/10.1007/s00371-009-0387-4

Burke, Robin. 2002. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction* 12, 4 (November 2002), 331–370. DOI:http://dx.doi.org/10.1023/A:1021240730564

Burke, Robin. 2000. Knowledge-Based Recommender Systems. In *ENCYCLOPEDIA OF LIBRARY AND INFORMATION SYSTEMS*. Marcel Dekker.

Cardoso, Bruno. 2016. *Two Steps Towards Kairos-Awareness*. Faculdade de Ciências e Tecnologia - Universidade Nova de Lisboa.

Centieiro, Pedro, Madeira, Rui Neves, Romão, Teresa, Dias, A. Eduardo, and Correia, Nuno. 2015. In sync with fair play!: delivering a synchronized and cheat-preventing second screen gaming experience. In *Proc. of the 12th International Conference on Advances in Computer Entertainment Technology - ACE '15*. New York, New York, USA: ACM Press, 1–11. DOI:http://dx.doi.org/10.1145/2832932.2832953

Centieiro, Pedro, Romão, Teresa, and Dias, A. Eduardo. 2014. From the lab to the world: studying real-time second screen interaction with live sports. In *Proc. of the 11th Conference on Advances in Computer Entertainment Technology (ACE '14)*. New York, New York, USA: ACM Press, 1–10. DOI:http://dx.doi.org/10.1145/2663806.2663843

Centieiro, Pedro, Romão, Teresa, Dias, A. Eduardo, and Madeira, Rui Neves. 2015a. A Synchronisation Mechanism based on User Feedback for Second Screen Applications. In *Proc. of 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous'15)*. Coimbra: ACM Press. DOI:http://dx.doi.org/10.4108/eai.22-7-2015.2260265

Centieiro, Pedro, Romão, Teresa, Dias, A. Eduardo, and Madeira, Rui Neves. 2015b. Synchronising Live Second Screen Applications with TV Broadcasts Through User Feedback. In Julio Abascal, Simone Barbosa, Mirko Fetter, Tom Gross, Philippe Palanque, & Marco Winckler, eds. *Proc. of IFIP TC.13 Conference on Human-Computer Interaction - INTERACT'15*. Lecture Notes in Computer Science. Cham: Springer International Publishing, 341–349. DOI:http://dx.doi.org/10.1007/978-3-319-22698-9

Chalmers, Dan, Chalmers, Matthew, Crowcroft, Jon, Kwiatkowska, Marta, Milner, Robin, O'Neill, Eamonn, Rodden, Tom, Sassone, Vladimiro, and Sloman, Morris. 2006. Ubiquitous Computing: Experience, Design and Science. In *Grand Challenges Manifesto*. version 4.

Chang, Chien-Yen, Lange, Belinda, Zhang, Mi, Koenig, Sebastian, Requejo, Phil, Somboon, Noom, and Sawchuk, Alexander A. 2012. Towards pervasive physical rehabilitation using Microsoft Kinect. In *Proc. of 2012 6th International Conference on Pervasive Computing Technologies for Healthcare and workshops (PervasiveHealth)*. ICST, 159–162.

Chang, Yao-Jen, Chen, Shu-Fang, and Huang, Jun-Da. 2011. A Kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. *Research in Developmental Disabilities* 32, 6 (November 2011), 2566–2570. DOI:http://dx.doi.org/10.1016/J.RIDD.2011.07.002

Chang, Yao-Jen, Han, Wen-Ying, and Tsai, Yu-Chi. 2013. A Kinect-based upper limb rehabilitation system to assist people with cerebral palsy. *Research in Developmental Disabilities* 34, 11 (November 2013), 3654–3659. DOI:http://dx.doi.org/10.1016/j.ridd.2013.08.021

Chang, Yen-ning, Lim, Youn-kyung, and Stolterman, Erik. 2008. Personas: from theory to practices. In *Proceedings of the 5th Nordic conference on Human-computer interaction building bridges - NordiCHI '08*. New York, New York, USA: ACM Press, 439. DOI:http://dx.doi.org/10.1145/1463160.1463214

Chmielewski, Dawn. 2014. Millennials Spend More Time Watching TV on Other Devices. (2014). Retrieved February 13, 2017 from http://www.recode.net/2014/3/26/11624950/teens-twentysomethings-spend-more-time-watching-tv-on-other-screens

Churruca, Silvana. 2013. Introduction to User Personas. *UX Lady* (2013), http://www.ux-com/introduction-to-user-person.

Cingil, Ibrahim, Dogac, Asuman, and Azgin, Ayca. 2000. A Broader Approach to Personalization. *Communications of the ACM* 43, 8 (August 2000), 136–141. DOI:http://dx.doi.org/10.1145/345124.345168

Clinch, Sarah. 2013. Smartphones and Pervasive Public Displays. *IEEE Pervasive Computing* 12, 1 (2013), 92–95. DOI:http://dx.doi.org/10.1109/MPRV.2013.16

Cohen, Philip R., and Perrault, C. Raymond. 1979. Elements of a Plan-Based Theory of Speech Acts. *Cognitive Science* 3, 3 (July 1979), 177–212. DOI:http://dx.doi.org/10.1207/s15516709cog0303_1

Conrad, Gary, and Johnsonbaugh, Richard. 1994. A real-time people counter. In *Proceedings of the 1994*

ACM symposium on Applied computing (SAC '94). New York, New York, USA: ACM Press, 20–24. DOI:http://dx.doi.org/10.1145/326619.326649

Cooper, Alan. 1999. *The inmates are running the asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity* 1 edition., Indianapolis: Sams Publishing.

Cooper, Alan, and Reimann, Robert. 2003. *About face 2.0: the essentials of interaction design* 2nd editio., New York, NY: John Wiley & Sons, Inc.

Cosmides, Leda. 1989. The logic of social exchange: has natural selection shaped how humans reason? Studies with the Wason selection task. *Cognition* 31, 3 (April 1989), 187–276.

Cosmides, Leda, and Tooby, John. 2005. Neurocognitive Adaptations Designed for Social Exchange. In David M. Buss, ed. *The Handbook of Evolutionary Psychology*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 584–627. DOI:http://dx.doi.org/10.1002/9780470939376.ch20

Coutinho, Pedro, and José, Rui. 2016. Risk Elicitation for User-Generated Content in Situated Interaction. In *Proc. of 10th International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI'16)*. Springer International Publishing, 481–486. DOI:http://dx.doi.org/10.1007/978-3-319-48746-5_49

Coutinho, Pedro, José, Rui, and Silva, Bruno. 2016. Understanding media situatedness and publication practices in place-based digital displays. In *Proc. of the 5th ACM International Symposium on Pervasive Displays - PerDis '16*. New York, New York, USA: ACM Press, 154–160. DOI:http://dx.doi.org/10.1145/2914920.2915025

Davies, Nigel, Friday, Adrian, Newman, Peter, Rutlidge, Sarah, and Storz, Oliver. 2009. Using bluetooth device names to support interaction in smart environments. In *Proc. of the 7th international conference on Mobile systems, applications, and services - Mobisys '09*. New York, New York, USA: ACM Press, 151. DOI:http://dx.doi.org/10.1145/1555816.1555832

Davies, Nigel, Langheinrich, Marc, Clinch, Sarah, Elhart, Ivan, Friday, Adrian, Kubitza, Thomas, and Surajbali, Bholanathsingh. 2014. Personalisation and privacy in future pervasive display networks. In *Proc. of the 32nd annual ACM conference on Human factors in computing systems (CHI '14)*. New York, New York, USA: ACM Press, 2357–2366. DOI:http://dx.doi.org/10.1145/2556288.2557287

Demarin, Iva, Leko, Ljubica, Škrobo, Maja, Germano, Helena, Macedo, Patrícia, and Madeira, Rui Neves. 2015. The Impact of Stuttering; How Can a Mobile App Help? In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility - ASSETS '15*. New York, New York, USA: ACM Press, 399–400. DOI:http://dx.doi.org/10.1145/2700648.2811389

Deponti, Dario, Maggiorini, Dario, and Palazzi, Claudio E. 2009. DroidGlove: An android-based application for wrist rehabilitation. In *2009 International Conference on Ultra Modern Telecommunications & Workshops*. IEEE, 1–7. DOI:http://dx.doi.org/10.1109/ICUMT.2009.5345442

Deterding, Sebastian, Sicart, Miguel, Nacke, Lennart, O'Hara, Kenton, and Dixon, Dan. 2011. Gamification: using game-design elements in non-gaming contexts. In *Proc. CHI'11 Extended Abstracts*. New York, New York, USA: ACM Press, 2425–2428. DOI:http://dx.doi.org/10.1145/1979742.1979575

Dimovska, Dajana, Jarnfelt, Patrick, Selvig, Sebbe, and Yannakakis, Georgios N. 2010. Towards procedural level generation for rehabilitation. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games - PCGames '10*. New York, New York, USA: ACM Press, 1–4. DOI:http://dx.doi.org/10.1145/1814256.1814263

Diogo, Rúben, and Banha, Diogo. 2016. *Walker - Contagem de Pessoas*, Setúbal.

Dix, Alan, Finlay, Janet E., Abowd, Gregory D., and Beale, Russell. 2003. *Human-Computer Interaction (3rd Edition)*, Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Even, Alon. 2015. Personalization: The Pillar of the Mobile User Experience. (2015). Retrieved October 8, 2015 from https://uxmag.com/articles/personalization-the-pillar-of-the-mobile-user-experience

Ferreira, Carlos, Guimarães, Vânia, Santos, António, and Sousa, Inês. 2014. Gamification of Stroke Rehabilitation Exercises Using a Smartphone. In *Proc. of the 8th International Conference on Pervasive Computing Technologies for Healthcare*. ICST, 282–285. DOI:http://dx.doi.org/10.4108/icst.pervasivehealth.2014.255326

Ferretti, Stefano, and Roccetti, Marco. 2006. AC/DC: an algorithm for cheating detection by cheating. In *Proc. of the 2006 international workshop on Network and operating systems support for digital audio and video (NOSSDAV '06)*. New York, New York, USA: ACM Press, 6. DOI:http://dx.doi.org/10.1145/1378191.1378220

Ferscha, Alois. 2012. 20 Years Past Weiser: What's Next? *IEEE Pervasive Computing* 11, 1 (2012), 52–61. DOI:http://dx.doi.org/10.1109/MPRV.2011.78

Flaherty, Kim, and Laubheimer, Page. 2017. Personas to User-Centered Design. (2017). Retrieved September 12, 2017 from https://www.nngroup.com/courses/personas/

Fogg, BJ. 2009. Creating persuasive technologies: an eight-step design process. In *Proc. of the 4th International Conference on Persuasive Technology - Persuasive '09*. New York, New York, USA: ACM Press, Article 44. DOI:http://dx.doi.org/10.1145/1541948.1542005

Fowler, Martin. 2010. *Domain Specific Languages* 1st ed., Addison-Wesley Professional.

Froehlich, Jon. 2009. Promoting Energy Efficient Behaviors in the Home through Feedback: The Role of Human-Computer Interaction. In *HCIC 2009 WINTER WORKSHOP*.

Fujisawa, Sizuka, Hasegawa, Go, Taniguchi, Yoshiaki, and Nakano, Hirotaka. 2013. Pedestrian counting in video sequences using optical flow clustering. *International Journal of Image Processing (IJIP)* 7, 1 (2013), 1–16.

Gamberini, Luciano, Björkskog, Christoffer, Salo, Marja, Aman, Pirkka, Corradi, Nicola, Zamboni, Luca, Perotti, Michela, Cadenazzi, Camilla, Mandressi, Stefano, Jacucci, Giulio, Tusa, Giovanni, and Spagnolli, Anna. 2011. Saving is fun: designing a persuasive game for power conservation. In *Proc. of the 8th International Conference on Advances in Computer Entertainment Technology - ACE '11*. New York, New York, USA: ACM Press, article 16. DOI:http://dx.doi.org/10.1145/2071423.2071443

Gamberini, Luciano, Spagnolli, Anna, Corradi, Nicola, Jacucci, Giulio, Tusa, Giovanni, Mikkola, Topi, Zamboni, Luca, and Hoggan, Eve. 2012. Tailoring Feedback to Users' Actions in a Persuasive Game for Household Electricity Conservation. In Springer Berlin Heidelberg, 100–111. DOI:http://dx.doi.org/10.1007/978-3-642-31037-9_9

Geleijnse, Gijs, Nachtigall, Peggy, van Kaam, Pim, and Wijgergangs, Luciënne. 2011. A Personalized Recipe Advice System to Promote Healthful Choices. In *Proc. of the 15th international conference on Intelligent user interfaces - IUI '11*. New York, New York, USA: ACM Press, 437–438. DOI:http://dx.doi.org/10.1145/1943403.1943487

Gonzalez, Gustavo, de la Rosa, Josep Lluis, Montaner, Miquel, and Delfin, Sonia. 2007. Embedding Emotional Context in Recommender Systems. In *2007 IEEE 23rd International Conference on Data Engineering Workshop*. IEEE, 845–852. DOI:http://dx.doi.org/10.1109/ICDEW.2007.4401075

Gorgoglione, Michele, Palmisano, Cosimo, and Tuzhilin, Alexander. 2006. Personalization in Context: Does Context Matter When Building Personalized Customer Models? In *Proc. of Sixth International Conference on Data Mining (ICDM'06)*. IEEE, 222–231. DOI:http://dx.doi.org/10.1109/ICDM.2006.125

Greenfield, Adam. 2006. *Everyware: The Dawning Age of Ubiquitous Computing*, Berkeley, CA, USA: Peachpit Press.

Gulliksen, Jan, Göransson, Bengt, Boivie, Inger, Persson, Jenny, Blomkvist, Stefan, and Cajander, Åsa. 2003. Key Principles for User-Centred Systems Design. *Behaviour & Information Technology* 22, 6 (2003), 397–409. DOI:http://dx.doi.org/10.1007/1-4020-4113-6_2

Haas, Hugo, and Brown, Allen. 2004. Web Services Glossary, W3C Working Group Note 11 February 2004. (2004). Retrieved August 24, 2016 from https://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/

Hamari, Juho, Koivisto, Jonna, and Sarsa, Harri. 2014. Does Gamification Work? -- A Literature Review of Empirical Studies on Gamification. In *Proc. of 47th Hawaii International Conference on System Sciences*. IEEE, 3025–3034. DOI:http://dx.doi.org/10.1109/HICSS.2014.377

Hashimoto, K., Morinaka, K., Yoshiike, N., Kawaguchi, C., and Matsueda, S. 1997. People count system using multi-sensing application. In *Proceedings of International Solid State Sensors and Actuators Conference (Transducers '97)*. IEEE, 1291–1294. DOI:http://dx.doi.org/10.1109/SENSOR.1997.635472

Hashimoto, Kazuhiko, Yoshinomoto, Makoto, Matsueda, Satoshi, Morinaka, Katsuya, and Yoshiike, Nobuyuki. 1997. Development of people-counting system with human-information sensor using multi-element pyroelectric infrared array detector. *Sensors and Actuators A: Physical* 58, 2 (February 1997), 165–171. DOI:http://dx.doi.org/10.1016/S0924-4247(96)01400-8

HBR, Harvard Business Review Analytic Services. 2012. *How Mobility is Changing the World*,

He, Helen Ai, and Greenberg, Saul. 2009. Motivating sustainable energy consumption in the home. In *Proc. of Defining the Role of HCI in the Challenges of Sustainability (ACM CHI'09 Workshop)*. 5 pages.

Helf, Christopher, and Hlavacs, Helmut. 2016. Apps for life change: Critical review and solution directions. *Entertainment Computing* 14 (2016), 17–22. DOI:http://dx.doi.org/10.1016/j.entcom.2015.07.001

Hernandez, Hamilton A., Graham, T.C. Nicholas, Fehlings, Darcy, Switzer, Lauren, Ye, Zi, Bellay, Quentin, Hamza, Md Ameer, Savery, Cheryl, and Stach, Tadeusz. 2012. Design of an exergaming station for children with cerebral palsy. In *Proc. of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*. New York, New York, USA: ACM Press, 2619. DOI:http://dx.doi.org/10.1145/2207676.2208652

Hernandez, Hamilton A., Ye, Zi, Graham, T.C. Nicholas, Fehlings, Darcy, and Switzer, Lauren. 2013. Designing action-based exergames for children with cerebral palsy. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*. New York, New York, USA: ACM Press, 1261. DOI:http://dx.doi.org/10.1145/2470654.2466164

Hilbert, David M., and Trevor, Jonathan. 2004. Personalizing Shared Ubiquitous Devices. *ACM Interactions* 11, 3 (May 2004), 34–43. DOI:http://dx.doi.org/10.1145/986253.986254

Hnat, Timothy W., Griffiths, Erin, Dawson, Ray, and Whitehouse, Kamin. 2012. Doorjamb: unobtrusive room-level tracking of people in homes using doorway sensors. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems - SenSys '12*. New York, New York, USA: ACM Press, 309–322. DOI:http://dx.doi.org/10.1145/2426656.2426687

Huizinga, Johan. 1971. *Homo ludens: a study of the play-element in culture*, Boston: Beacon Press.

Hurwitz, Judith, Bloor, Robin, Kaufman, Marcia, and Halper, Fern. 2009. *Cloud Computing For Dummies:*, Wiley Publishing, Inc.

IETF, Internet Engineering Task Force. 2012. The OAuth 2.0 Authorization Framework. Tech. rep. IETF. (2012). https://tools.ietf.org/html/rfc6749

Jambon, Francis, and Meillon, Brigitte. 2009. User experience evaluation in the wild. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems - CHI EA '09*. New York, New York, USA: ACM Press, 4069. DOI:http://dx.doi.org/10.1145/1520340.1520619

Jeh, Glen, and Widom, Jennifer. 2003. Scaling Personalized Web Search. In *Proc. of the twelfth international conference on World Wide Web - WWW '03*. New York, New York, USA: ACM Press, 271–279. DOI:http://dx.doi.org/10.1145/775152.775191

Johnson, Peter. 1992. *Human Computer Interaction: Psychology, Task Analysis and Software Engineering* 1st edition., McGraw-Hill.

José, Rui, and Cardoso, Jorge C.S. 2011. Opportunities and Challenges of Interactive Public Displays as an Advertising Medium. In Jörg Müller, Florian Alt, & Daniel Michelis, eds. *Pervasive Advertising*. Springer London, 139–157. DOI:http://dx.doi.org/10.1007/978-0-85729-352-7_7

José, Rui, Otero, Nuno, Izadi, Shahram, and Harper, Richard. 2008. Instant Places: Using Bluetooth for Situated Interaction in Public Displays. *IEEE Pervasive Computing* 7, 4 (October 2008), 52–57. DOI:http://dx.doi.org/10.1109/MPRV.2008.74

Junior, Plinio Thomaz Aquino, and Filgueiras, Lucia Vilela Leite. 2005. User modeling with personas. In *Proceedings of the 2005 Latin American conference on Human-computer interaction - CLIHC '05*. New York, New York, USA: ACM Press, 277–282. DOI:http://dx.doi.org/10.1145/1111360.1111388

Kabassi, Katerina, Virvou, Maria, and Tsihrintzis, George A. 2008. User Modelling Sharing for Adaptive E-learning and Intelligent Help. In Gulden Uchyigit & Matthew Y. Ma, eds. *Personalization Techniques and Recommender Systems*. World Scientific, 85–110.

Kay, Judith, Kummerfeld, Robert, and Kuflik, Tsvi. 2012. Challenges and Solutions of Ubiquitous User Modeling. In Antonio Krüger & Tsvi Kuflik, eds. *Ubiquitous Display Environments*. Cognitive Technologies. Berlin, Heidelberg: Springer Berlin Heidelberg, 7–30. DOI:http://dx.doi.org/10.1007/978-3-642-27663-7

Kay, Judy. 1995. The um toolkit for cooperative user modelling. *User Modelling and User-Adapted Interaction* 4, 3 (1995), 149–196. DOI:http://dx.doi.org/10.1007/BF01100243

Kay, Judy, Kummerfeld, Bob, and Lauder, Piers. 2002. Personis: A Server for User Models. In *Proc. of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'02)*. Springer-Verlag, 203–212.

Kellar, Melanie, Parker, Karen, Whalen, Tara, Inkpen, Kori M., Reilly, Derek, Hawkey, Kirstie, Rodgers, Malcolm, MacKay, Bonnie, Dearman, David, Ha, Vicki, MacInnes, W. Joseph, and Nunes, Michael. 2005. It's a jungle out there. In *CHI '05 extended abstracts on Human factors in computing systems - CHI '05*. New York, New York, USA: ACM Press, 1533. DOI:http://dx.doi.org/10.1145/1056808.1056959

Kobsa, Alfred. 2004. Adaptive Interfaces. In W. S. Bainbridge, ed. *Encyclopedia of Human-Computer Interaction*. Great Barrington: Berkshire Publishing, 9–16.

Kobsa, Alfred. 2007. Generic User Modeling Systems. In Peter Brusilovsky, Alfred Kobsa, & Wolfgang Nejdl, eds. *The Adaptive Web: Methods and Strategies of Web Personalization*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 136–154. DOI:http://dx.doi.org/10.1007/978-3-540-72079-9

Koch, Nora. 2000. *Software Engineering for Adaptive Hypermedia Systems*. Ludwig-Maximilians University, Munich, Germany.

Kooij, Wouter J., Stokking, Hans M., van Brandenburg, Ray, and de Boer, Pieter-Tjerk. 2014. Playout delay of TV signals: measurement system design, validation and results. In *Proc. of the 2014 ACM international conference on Interactive experiences for TV and online video (TVX '14)*. New York, New York, USA: ACM Press, 23–30. DOI:http://dx.doi.org/10.1145/2602299.2602310

Kostakos, Vassilis, and Ojala, Timo. 2013. Public Displays Invade Urban Spaces. *IEEE Pervasive Computing* 12, 1 (2013), 8–13. DOI:http://dx.doi.org/10.1109/MPRV.2013.15

Kramer, Joseph, Noronha, Sunil, and Vergo, John. 2000. A user-centered design approach to personalization. *Communications of the ACM* 43, 8 (August 2000), 44–48. DOI:http://dx.doi.org/10.1145/345124.345139

Kubitza, Thomas, Clinch, Sarah, Davies, Nigel, and Langheinrich, Marc. 2013. Using mobile devices to personalize pervasive displays. *ACM SIGMOBILE Mobile Computing and Communications Review* 16, 4 (February 2013), 26. DOI:http://dx.doi.org/10.1145/2436196.2436211

Kuflik, Tsvi, Berkovsky, Shlomo, Heckmann, Dominik, and Krüger, Antonio. 2008. UbiqUM 2008: theories and applications of ubiquitous user modeling. In *Proc. of the 13th international conference on Intelligent user interfaces - IUI '08*. New York, New York, USA: ACM Press, 438. DOI:http://dx.doi.org/10.1145/1378773.1378858

Kurdyukova, Ekaterina, Hammer, Stephan, and André, Elisabeth. 2012. Personalization of Content on Public Displays Driven by the Recognition of Group Context. In *Proc. of the third International*

Joint Conference on Ambient Intelligence (AmI 2012). Pisa: Springer Berlin Heidelberg, 272–287. DOI:http://dx.doi.org/10.1007/978-3-642-34898-3_18

Langley, Pat. 1999. User modeling in adaptive interfaces. In *Proc. of the seventh international conference on User modeling (UM '99)*. Springer-Verlag, 357–370.

Laurens, Peter, Paige, Richard F., Brooke, Phillip J., and Chivers, Howard. 2007. A Novel Approach to the Detection of Cheating in Multiplayer Online Games. In *Proc. of 12th IEEE International Conference on Engineering Complex Computer Systems (ICECCS 2007)*. IEEE, 97–106. DOI:http://dx.doi.org/10.1109/ICECCS.2007.11

Lemire, Daniel, and Maclachlan, Anna. 2007. Slope One Predictors for Online Rating-Based Collaborative Filtering. *The Computing Research Repository (CoRR)* (February 2007).

Lewis, Norman. 2004. From customization to ubiquitous personalization. *ACM Interactions* 11, 2 (March 2004), 48. DOI:http://dx.doi.org/10.1145/971258.971273

Liu, X., Tu, P.H., Rittscher, J., Perera, A., and Krahnstoever, N. 2005. Detecting and counting people in surveillance applications. In *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS'05)*. IEEE, 306–311. DOI:http://dx.doi.org/10.1109/AVSS.2005.1577286

Lozano-Quilis, José-Antonio, Albiol-Pérez, Sergio, Gil-Gómez, Hermenegildo, Palacios-Navarro, Guillermo, Fardoun, Habib, Gil-Gómez, José-Antonio, and Mashat, Abdulfattah. 2013. Virtual Reality System for Multiple Sclerosis Rehabilitation using KINECT. In *Proc. of Pervasive Computing Technologies for Healthcare (PervasiveHealth)*. IEEE, 366–369. DOI:http://dx.doi.org/10.4108/icst.pervasivehealth.2013.252208

Luna-Oliva, Laura, Ortiz-Gutiérrez, Rosa María, Cano-de la Cuerda, Roberto, Piédrola, Rosa Martínez, Alguacil-Diego, Isabel M., Sánchez-Camarero, Carlos, and Martínez Culebras, María Del Carmen. 2013. Kinect Xbox 360 as a therapeutic modality for children with cerebral palsy in a school environment: a preliminary study. *NeuroRehabilitation* 33, 4 (2013), 513–21. DOI:http://dx.doi.org/10.3233/NRE-131001

Ma, Minhua, and Bechkoum, Kamal. 2008. Serious games for movement therapy after stroke. In *2008 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 1872–1877. DOI:http://dx.doi.org/10.1109/ICSMC.2008.4811562

Macdonald, Craig Matthew. 2012. *Understanding usefulness in human-computer interaction to enhance user experience evaluation*. Drexel University.

Madeira, Rui Neves. 2012. Personalization in Pervasive Spaces towards Smart Interactions Design. In *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*. IEEE, 548–549. DOI:http://dx.doi.org/10.1109/PerComW.2012.6197568

Madeira, Rui Neves, Antunes, André, and Postolache, Octavian. 2016. just Physio kidding - NUI and Gamification based Therapeutic Intervention for Children with Special Needs. In *Proceedings of 6th EAI International Symposium on Pervasive Computing Paradigms for Mental Health (MindCare'16)*. Barcelona: Springer Berlin / Heidelberg.

Madeira, Rui Neves, Antunes, André, Postolache, Octavian, and Correia, Nuno. 2017. Serious…ly! Just Kidding in Personalized Therapy through Natural Interactions with Games. In *Proceedings of 14th International Conference on Advances in Computer Entertainment Technology Conference (ACE'17)*. London: Springer, 19.

Madeira, Rui Neves, Centieiro, Pedro, and Correia, Nuno. 2015. Adaptation to TV Delays Based on the User Behaviour towards a Cheating-Free Second Screen Entertainment. In *Proc. of Entertainment Computing Conference (ICEC 2015)*. Trondheim: Springer International Publishing, 424–432. DOI:http://dx.doi.org/10.1007/978-3-319-24589-8_35

Madeira, Rui Neves, and Correia, Nuno. 2010. An Infrastructure for Real Objects Augmentation with Additional Personalized Information Services. In *Proc. of the 7th international conference on Ubiquitous intelligence and computing (UIC'10)*. Xian, China: Springer-Verlag, 282–296.

Madeira, Rui Neves, and Correia, Nuno. 2011. Towards Personalized Pervasive Theragames in Smart Wheelchairs. In *Adjunt Proceedings of 19th International Conference on User Modeling, Adaptation and*

Personalization (UMAP 2011). Girona: Springer.

Madeira, Rui Neves, Correia, Nuno, Dias, Ana Claudia, Guerra, Marco, Postolache, Octavian, and Postolache, Gabriela. 2011. Designing Personalized Therapeutic Serious Games for a Pervasive Assistive Environment. In *2011 IEEE 1st International Conference on Serious Games and Applications for Health (SeGAH)*. IEEE, 1–10. DOI:http://dx.doi.org/10.1109/SeGAH.2011.6165465

Madeira, Rui Neves, Costa, Luís, and Postolache, Octavian. 2014. PhysioMate - Pervasive Physical Rehabilitation based on NUI and Gamification. In *Proc. of 2014 International Conference and Exposition on Electrical and Power Engineering (EPE'14)*. IEEE, 612–616. DOI:http://dx.doi.org/10.1109/ICEPE.2014.6969982

Madeira, Rui Neves, Macedo, Patricia, Pereira, Carla, Germano, Helena, and Ferreira, João. 2017. Mobile Apps to improve ThErapy - The Health Practitioner in your pocket knows you. In *Proceedings of the 15th International Conference on Advances in Mobile Computing & Multimedia (MoMM2017)*. Salzburg: ACM.

Madeira, Rui Neves, Macedo, Patrícia, Pita, Pedro, Bonança, Íris, and Germano, Helena. 2013. Building on Mobile towards Better Stuttering Awareness to Improve Speech Therapy. In *Proc. of International Conference on Advances in Mobile Computing & Multimedia (MoMM '13)*. New York, New York, USA: ACM Press, 551–554. DOI:http://dx.doi.org/10.1145/2536853.2536911

Madeira, Rui Neves, Macedo, Patrícia, Reis, Sofia, and Ferreira, João. 2014. Super-Fon: Mobile Entertainment to Combat Phonological Disorders in Children. In *Proc. of the 11th Conference on Advances in Computer Entertainment Technology - ACE '14*. New York, New York, USA: ACM Press, 1–4. DOI:http://dx.doi.org/10.1145/2663806.2663870

Madeira, Rui Neves, Mestre, Vanessa, and Ferreirinha, Tânia. 2017. Phonological Disorders in Children? Design and user experience evaluation of a mobile serious game approach. *Procedia Computer Science* 113 (January 2017), 416–421. DOI:http://dx.doi.org/10.1016/J.PROCS.2017.08.359

Madeira, Rui Neves, Postolache, Octavian, and Correia, Nuno. 2012. Gaming for Therapy in a Healthcare Smart Ambient. In *Constructing Ambient Intelligence - Proc. of AmI 2011 Workshops*. Amsterdam: Springer Berlin Heidelberg, 224–228. DOI:http://dx.doi.org/10.1007/978-3-642-31479-7_38

Madeira, Rui Neves, Postolache, Octavian, Correia, Nuno, and Girão, Pedro Silva. 2010. Designing a Pervasive Healthcare Assistive Environment for the Elderly. In *Proc. of 5th International Workshop on Ubiquitous Health and Wellness at ACM UbiComp 2010 (UbiHealth'10)*. Copenhaga: CEUR Workshop Proceedings.

Madeira, Rui Neves, Santos, Pedro Albuquerque, and Correia, Nuno. 2014. Building a Platform for Pervasive Personalization in a Ubiquitous Computing World. In *Proc. of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS'14)*. ICST, 345–346. DOI:http://dx.doi.org/10.4108/icst.mobiquitous.2014.258158

Madeira, Rui Neves, Santos, Pedro Albuquerque, Vieira, André, and Correia, Nuno. 2014. Model-Based Solution for Personalization of the User Interaction in Ubiquitous Computing. In *Proc of 2014 IEEE International Conference on Ubiquitous Intelligence and Computing*. 311–318. DOI:http://dx.doi.org/10.1109/UIC-ATC-ScalCom.2014.139

Madeira, Rui Neves, Silva, André, Santos, Catarina, Teixeira, Bárbara, Romão, Teresa, Dias, Eduardo, and Correia, Nuno. 2011. LEY! Persuasive Pervasive Gaming on Domestic Energy Consumption-Awareness. In *Proc. of the 8th International Conference on Advances in Computer Entertainment Technology - ACE '11*. New York, New York, USA: ACM Press, 1. DOI:http://dx.doi.org/10.1145/2071423.2071512

Madeira, Rui Neves, Vieira, André, and Correia, Nuno. 2012. Personalization of an Energy Awareness Pervasive Game. In *Proc. of the 2012 ACM Conference on Ubiquitous Computing - UbiComp '12*. New York, New York, USA: ACM Press, 619–620. DOI:http://dx.doi.org/10.1145/2370216.2370333

Maes, Pattie. 1994. Agents that reduce work and information overload. *Communications of the ACM* 37, 7 (July 1994), 30–40. DOI:http://dx.doi.org/10.1145/176789.176792

Mekuria, Rufael, Cesar, Pablo, and Bulterman, Dick. 2012a. Digital TV: the effect of delay when watching football. In *Proc. of the 10th European conference on Interactive tv and video - EuroiTV '12*. New York, New York, USA: ACM Press, 71–74. DOI:http://dx.doi.org/10.1145/2325616.2325632

Mekuria, Rufael, Cesar, Pablo, and Bulterman, Dick. 2012b. Digital TV: the effect of delay when watching football. In *Proceedings of the 10th European conference on Interactive tv and video - EuroiTV '12*. New York, New York, USA: ACM Press, 71. DOI:http://dx.doi.org/10.1145/2325616.2325632

Mell, Peter, and Grance, Timothy. 2011. *The NIST Definition of Cloud Computing: Recommendations of the National Institute of Standarts and Technology*,

Miaskiewicz, Tomasz, and Kozar, Kenneth A. 2011. Personas and user-centered design: How can personas benefit product design processes? *Design Studies* 32, 5 (September 2011), 417–430. DOI:http://dx.doi.org/10.1016/j.destud.2011.03.003

Mobasher, Bamshad, and Tuzhilin, Alexander. 2008. Preface to the special issue on Data Mining for Personalization. *User Modeling and User-Adapted Interaction* 19, 1–2 (November 2008), 1–3. DOI:http://dx.doi.org/10.1007/s11257-008-9060-2

Muller, Michael J. 2002. Participatory Design - The Third Space in HCI. In Julie A. Jacko & Andrew Sears, eds. *The Uuman-Computer Interaction - Development Process*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1051–1068. DOI:http://dx.doi.org/10.1201/9781420088892.ch9

Mulvenna, Maurice D., Anand, Sarabjot S., and Büchner, Alex G. 2000. Personalization on the Net using Web mining: introduction. *Communications of the ACM* 43, 8 (August 2000), 122–125. DOI:http://dx.doi.org/10.1145/345124.345165

Nielsen, Jakob. 2010. Testing Expert Users. (2010). https://www.nngroup.com/articles/testing-expert-users/

Nielsen, Lene. 2013. Personas. In Mads Soegaard & Rikke Friis Dam, eds. *The Encyclopedia of Human-Computer Interaction*. Aarhus: Interaction Design Foundation.

Nielsen Holdings. 2014. *Millennials: Breaking the Myths*, New York. Retrieved February 13, 2017 from http://www.exploremidtown.org/wp-content/uploads/2015/04/nielsen-millennial-report-feb-2014.pdf.

Notelaers, Sofie, Weyer, Tom De, Robert, Karel, Raymaekers, Chris, and Coninx, Karin. 2010. Design Aspects for Rehabilitation Games for MS Patients. In *Proceedings of Design and Engineering of Game-like Virtual and Multimodal Environment*. Berlin.

O'Mahony, Kevin, Liang, Jian, and Delaney, Kieran. 2012. User-Centric Personalization and Autonomous Reconfiguration Across Ubiquitous Computing Environments. In *Proc. of UBICOMM 2012, The Sixth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*. Barcelona, 48–53.

Oinas-Kukkonen, Harri. 2013. A foundation for the study of behavior change support systems. *Personal and Ubiquitous Computing* 17, 6 (August 2013), 1223–1235. DOI:http://dx.doi.org/10.1007/s00779-012-0591-5

Oppermann, Reinhard. 1994. *Adaptive user support : ergonomic design of manually and automatically adaptable software*, Lawrence Erlbaum Associates.

Oppermann, Reinhard, Rashev, Rossen, and Kinshuk. 1998. Adaptability and Adaptivity in Learning Systems. In A. Behrooz, ed. *Knowledge Transfer (Volume II)*. London: pAce, 173–179.

Orland, Brian, Ram, Nilam, Lang, Dean, Houser, Kevin, Kling, Nate, and Coccia, Michael. 2014. Saving energy in an office environment: A serious game intervention. *Energy and Buildings* 74 (2014), 43–52. DOI:http://dx.doi.org/10.1016/j.enbuild.2014.01.036

Osgood, Charles Egerton, May, William H., and Miron, Murray S. 1975. *Cross-cultural Universals of Affective Meaning* 1st ed., University of Illinois Press.

Palmisano, Cosimo, Tuzhilin, Alexander, and Gorgoglione, Michele. 2008. Using Context to Improve Predictive Modeling of Customers in Personalization Applications. *IEEE Transactions on Knowledge and Data Engineering* 20, 11 (November 2008), 1535–1549. DOI:http://dx.doi.org/10.1109/TKDE.2008.110

Panayiotou, Christoforos, Andreou, Maria, and Samaras, George. 2006. Using Time and Activity in Personalization for the Mobile User. In *Proc. of the 5th ACM international workshop on Data engineering for wireless and mobile access (MobiDE '06)*. New York, New York, USA: ACM Press, 87–90. DOI:http://dx.doi.org/10.1145/1140104.1140122

Panayiotou, Christoforos, and Samaras, George. 2006. Mobile User Personalization with Dynamic Profiles: Time and Activity. In Robert Meersman, Zahir Tari, & Pilar Herrero, eds. *Proc. of On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 1295–1304. DOI:http://dx.doi.org/10.1007/11915072

Papazoglou, M. 2008. *Web services : Principles and Technology, (Chapter 1 and 2)*, Pearson/Prentice Hall.

Parker, Callum, Kay, Judy, Baldauf, Matthias, and Tomitsch, Martin. 2016. Design implications for interacting with personalised public displays through mobile augmented reality. In *Proc. of the 5th ACM International Symposium on Pervasive Displays (PerDis '16)*. New York, New York, USA: ACM Press, 52–58. DOI:http://dx.doi.org/10.1145/2914920.2915016

Paterno, Fabio. 2014. User Interface Design Adaptation. In Mads Soegaard & Rikke Friis Dam, eds. *The Encyclopedia of Human-Computer Interaction, 2nd Ed.* Aarhus, Denmark: The Interaction Design Foundation.

Pazzani, Michael J., and Billsus, Daniel. 2007. Content-based recommendation systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl, eds. *The adaptive web*. Springer-Verlag, 325–341.

Pereira, Carla, Macedo, Patrícia, and Madeira, Rui Neves. 2015. Mobile Integrated Assistance to Empower People Coping with Parkinson's Disease. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility - ASSETS '15*. New York, New York, USA: ACM Press, 409–410. DOI:http://dx.doi.org/10.1145/2700648.2811394

Pereira, Carla, Madeira, Rui Neves, and Macedo, Patricia. 2016. ONParkinson – Innovative mHealth to support the triad: patient, carer and health professional. In *Proceedings of 6th EAI International Symposium on Pervasive Computing Paradigms for Mental Health (MindCare'16)*. Springer Berlin / Heidelberg.

Resnick, Paul, Iacovou, Neophytos, Suchak, Mitesh, Bergstrom, Peter, and Riedl, John. 1994. GroupLens: an open architecture for collaborative filtering of netnews. In *Proc. of the 1994 ACM conference on Computer supported cooperative work - CSCW '94*. New York, New York, USA: ACM Press, 175–186. DOI:http://dx.doi.org/10.1145/192844.192905

Ricci, Francesco, Rokach, Lior, Shapira, Bracha, and Kantor, Paul B. 2010. *Recommender Systems Handbook*, Springer-Verlag New York, Inc.

Rich, Elaine. 1979a. *Building and exploiting user models. PhD Thesis*. Pittsburgh, PA: Carnegie-Mellon University.

Rich, Elaine. 1979b. User modeling via stereotypes. *Cognitive Science* 3, 4 (1979), 329–354. DOI:http://dx.doi.org/10.1016/S0364-0213(79)80012-9

Rodden, Tom. 2008. Living in a ubiquitous world. *Philosophical Transactions of the Royal Society of London Series A* 366, 1881 (October 2008), 3837–3838. DOI:http://dx.doi.org/10.1098/rsta.2008.0146

Romão, Teresa, Correia, Nuno, and Dias, Eduardo. 2012. DEAP - Developing Environmental Awareness with Persuasive Systems. (2012). Retrieved January 29, 2017 from http://nova-lincs.di.fct.unl.pt/project/69

Saini, Sanjay, Rambli, Dayang Rohaya Awang, Sulaiman, Suziah, Zakaria, Mohamed Nordin, and Mohd Shukri, Siti Rohkmah. 2012. A low-cost game framework for a home-based stroke rehabilitation system. In *2012 International Conference on Computer & Information Science (ICCIS)*. IEEE, 55–60. DOI:http://dx.doi.org/10.1109/ICCISci.2012.6297212

Salber, Daniel, Dey, Anind K., and Abowd, Gregory D. 1998. *Ubiquitous Computing: Defining an HCI Research Agenda for an Emerging Interaction Paradigm*, GVU Technical Report (GIT-GVU-98-01), Georgia Institute of Technology.

Santos, Pedro Albuquerque, Madeira, Rui Neves, and Correia, Nuno. 2013. FCT4U -- When Private Mobile Displays Meet Public Situated Displays to Enhance the User Experience. In *Proc. of 2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing*. IEEE, 186–193. DOI:http://dx.doi.org/10.1109/UIC-ATC.2013.112

Sarwar, Badrul, Karypis, George, Konstan, Joseph, and Reidl, John. 2001. Item-based collaborative filtering recommendation algorithms. In *Proc. of the tenth international conference on World Wide Web - WWW '01*. New York, New York, USA: ACM Press, 285–295. DOI:http://dx.doi.org/10.1145/371920.372071

Sarwar, Badrul, Karypis, George, Konstan, Joseph, and Riedl, John. 2000. Analysis of recommendation algorithms for e-commerce. In *Proc. of the 2nd ACM conference on Electronic commerce - EC '00*. New York, New York, USA: ACM Press, 158–167. DOI:http://dx.doi.org/10.1145/352871.352887

Satyanarayanan, M. 2001. Pervasive computing: vision and challenges. *IEEE Personal Communications* 8, 4 (2001), 10–17. DOI:http://dx.doi.org/10.1109/98.943998

Sauro, Jeff. 2011. *A Practical Guide to the System Usability Scale: Background, Benchmarks & Best Practices*, Denver, CO: CreateSpace Independent Publishing Platform.

Schade, Amy. 2016a. Customization vs. Personalization in the User Experience. (2016). Retrieved September 1, 2017 from https://www.nngroup.com/articles/customization-personalization/

Schade, Amy. 2016b. Tips for Successful Personalization. (2016). Retrieved February 12, 2017 from https://www.nngroup.com/articles/personalization/

Schilit, Bill, Adams, Norman, and Want, Roy. 1994. Context-Aware Computing Applications. In *Proc. of the 1994 First Workshop on Mobile Computing Systems and Applications (WMCSA '94)*. IEEE, 85–90. DOI:http://dx.doi.org/10.1109/WMCSA.1994.16

Sebesta, Robert W. 2010. *Concepts of programming languages* 11th editi., Pearson Education Limited.

Sharifi, Matthew, Payne, Terry R., and David, Esther. 2006. Public Display Advertising Based on Bluetooth Device Presence. In *Proc. of Mobile Interaction with the Real World workshop at MobileHCI '06*. 52–55.

Stone, Robert J. 2008. *Human Factors Guidelines for. Interactive 3D and Games-Based Training*, Birmingham (UK).

Surie, Dipak, Partonia, Saeed, and Lindgren, Helena. 2013. Human Sensing Using Computer Vision for Personalized Smart Spaces. In *2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing*. Vietri sul Mere: IEEE, 487–494. DOI:http://dx.doi.org/10.1109/UIC-ATC.2013.24

Taylor, Matthew J.D., McCormick, Darren, Shawis, Teshk, Impson, Rebecca, and Griffin, Murray. 2011. Activity-promoting gaming systems in exercise and rehabilitation. *Journal of rehabilitation research and development* 48, 10 (2011), 1171–86.

Teixeira, Thiago, and Savvides, Andreas. 2007. Lightweight People Counting and Localizing in Indoor Spaces Using Camera Sensor Nodes. In *2007 First ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC '07)*. IEEE, 36–43. DOI:http://dx.doi.org/10.1109/ICDSC.2007.4357503

Teltzrow, Maximilian, and Kobsa, Alfred. 2004. Impacts of user privacy preferences on personalized systems: a comparative study. In Clare-Marie. Karat, Jan O. Blom, & John. Karat, eds. *Designing personalized user experiences in eCommerce*. Kluwer Academic Publishers, 315–332. DOI:http://dx.doi.org/10.1007/1-4020-2148-8_17

Tullis, Tom (Thomas), and Albert, Bill (William). 2013. *Measuring the User Experience: collecting, analyzing, and presenting usability metrics* 2nd Edition., San Francisco, CA: Morgan Kaufmann.

Turchi, Tommaso, and Malizia, Alessio. 2015. Pervasive Displays in the Wild: Employing End User Programming in Adaption and Re-Purposing. In *Proc. of 5th International Symposium on End-User Development (IS-EUD 2015)*. Madrid: Springer International Publishing, 223–229. DOI:http://dx.doi.org/10.1007/978-3-319-18425-8_20

Turner, Anne M., Reeder, Blaine, and Ramey, Judith. 2013. Scenarios, personas and user stories: User-centered evidence-based design representations of communicable disease investigations. *Journal of Biomedical Informatics* 46, 4 (August 2013), 575–584. DOI:http://dx.doi.org/10.1016/j.jbi.2013.04.006

tutorialspoint. 2016. What are Web Services? (2016). Retrieved August 10, 2016 from http://www.tutorialspoint.com/webservices/what_are_web_services.htm

usability.gov. 2013. Personas. (October 2013). Retrieved September 8, 2017 from https://www.usability.gov/how-to-and-tools/methods/personas.html

Vaquero, Luis M., Rodero-Merino, Luis, Caceres, Juan, and Lindner, Maik. 2008. A break in the clouds: towards a cloud definition. *ACM SIGCOMM Computer Communication Review* 39, 1 (December 2008), 50. DOI:http://dx.doi.org/10.1145/1496091.1496100

Veryzer, Robert W., and Borja de Mozota, Brigitte. 2005. The impact of user-oriented design on new product development: an examination of fundamental relationships. *Journal of Product Innovation Management* 22, 2 (2005), 128–143.

Videonet. 2014. Making TV more personal. (2014), http://v-tv/wp-content/uploads/Making%20TV%20M.

Vredenburg, Karel, Mao, Ji-Ye, Smith, Paul W., and Carey, Tom. 2002. A survey of user-centered design practice. In *Proceedings of the SIGCHI conference on Human factors in computing systems Changing our world, changing ourselves - CHI '02*. New York, New York, USA: ACM Press, 471. DOI:http://dx.doi.org/10.1145/503376.503460

Want, Roy. 2013. Smartphones and Pervasive Public Displays - Editor's Intro. *IEEE Pervasive Computing* 12, 1 (2013), 92–95. DOI:http://dx.doi.org/10.1109/MPRV.2013.16

Want, Roy, and Schilit, Bill N. 2012. Interactive Digital Signage. *Computer* 45, 5 (May 2012), 21–24. DOI:http://dx.doi.org/10.1109/MC.2012.169

Weiser, Mark. 1991. The Computer for the 21st Century. *Scientific American* 265, 3 (September 1991), 94–104. DOI:http://dx.doi.org/10.1038/scientificamerican0991-94

Wiemeyer, Josef, and Kliem, Annika. 2012. Serious games in prevention and rehabilitation—a new panacea for elderly people? *European Review of Aging and Physical Activity* 9, 1 (April 2012), 41–50. DOI:http://dx.doi.org/10.1007/s11556-011-0093-x

Witten, Ian H., Frank, Eibe, and Hall, Mark A. 2011. *Data Mining: Practical Machine Learning Tools and Techniques* 3rd ed., San Francisco: Morgan Kaufmann Publishers Inc.

Woerndl, Wolfgang, and Schlichter, Johann. 2007. Introducing Context into Recommender Systems. In *Proc. AAAI'07 Workshop on Recommender Systems in e-Commerce*. Vancouver.

Woodruff, Allison, and Mankoff, Jennifer. 2009. Environmental Sustainability. *IEEE Pervasive Computing* 8, 1 (January 2009), 18–21. DOI:http://dx.doi.org/10.1109/MPRV.2009.6

Xu, Chenren. 2013. Device-free people counting and localization. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication - UbiComp '13 Adjunct*. New York, New York, USA: ACM Press, 367–372. DOI:http://dx.doi.org/10.1145/2494091.2501091

Yan, Bo, and Chen, Guanling. 2011. AppJoy: personalized mobile application discovery. In *Proc. of the 9th international conference on Mobile systems, applications, and services - MobiSys '11*. New York, New York, USA: ACM Press, 113–126. DOI:http://dx.doi.org/10.1145/1999995.2000007

Yan, Jeff. 2003. Security design in online games. In *Proc. of 19th Annual Computer Security Applications Conference*. IEEE, 286–295. DOI:http://dx.doi.org/10.1109/CSAC.2003.1254333

Yan, Jeff, and Randell, Brian. 2005. A systematic classification of cheating in online games. In *Proc. of 4th ACM SIGCOMM workshop on Network and system support for games (NetGames '05)*. New York, New York, USA: ACM Press, 1–9. DOI:http://dx.doi.org/10.1145/1103599.1103606

Yan, Jianxin Jeff, and Choi, Hyun-jin. 2002. Security Issues in Online Games. *The Electronic Library* 20, 2 (2002), 125–133.

Yang, Gonzalez-Banos, and Guibas. 2003. Counting people in crowds with a real-time network of simple image sensors. In *Proceedings Ninth IEEE International Conference on Computer Vision*. IEEE, 122–129 vol.1. DOI:http://dx.doi.org/10.1109/ICCV.2003.1238325

Yeung, S.F., Lui, J.C.S., Jiangchuan Liu, and Yan, J. 2006. Detecting cheaters for multiplayer games: theory, design and implementation. In *Proc. of the 3rd IEEE Consumer Communications and Networking Conference (CCNC 2006)*. IEEE, 1178–1182. DOI:http://dx.doi.org/10.1109/CCNC.2006.1593224

Yogi Teas. 2017. Yogi Tea Bags Inspirations. (2017). Retrieved February 11, 2017 from https://www.yogiproducts.com/

York, Judy, and Pendharkar, Parag C. 2004. Human–computer interaction issues for mobile computing in a variable work context. *International Journal of Human-Computer Studies* 60, 5–6 (May 2004), 771–797. DOI:http://dx.doi.org/10.1016/j.ijhcs.2003.07.004

Yu, Zhiwen, Cheng, Doreen, Khalil, Ismail, Kay, Judy, and Heckmann, Dominikus. 2011. Theme issue on adaptation and personalization for ubiquitous computing. *Personal and Ubiquitous Computing* 16, 5 (August 2011), 467–468. DOI:http://dx.doi.org/10.1007/s00779-011-0441-x

Zhang, Jie, and Ghorbani, Ali A. 2007. GUMSAWS: A Generic User Modeling Server for Adaptive Web Systems. In *Proc. of Fifth Annual Conference on Communication Networks and Services Research (CNSR '07)*. IEEE, 117–124. DOI:http://dx.doi.org/10.1109/CNSR.2007.35

Zhao, H., and Shibasaki, R. 2005a. A Novel System for Tracking Pedestrians Using Multiple Single-Row Laser-Range Scanners. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 35, 2 (March 2005), 283–291. DOI:http://dx.doi.org/10.1109/TSMCA.2005.843396

Zhao, H., and Shibasaki, R. 2005b. A Real-Time System for Monitoring Pedestrians. In *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*. IEEE, 378–385. DOI:http://dx.doi.org/10.1109/ACVMOT.2005.10

Zichermann, Gabe, and Cunningham, Christopher. 2011. *Gamification by design : implementing game mechanics in web and mobile apps*, O'Reilly Media.

***Note**: All provided links were checked for availability by the 24th of October 2017*

A. *Personalisations Design* – XUPIL scripts

This appendix presents the XUPIL scripts used in the definitions of the prototypes' personalisations.

A.1. LEY (Study D.1)

```
<?xml version='1.0' standalone='yes'?>
<xupil> <!--NOT submitted to P2MUCA since by the time P2MUCA did not exist and this was to test CAPE -->
    <database>
        <name>testeIniCAPE</name>
        <password>testeIniCAPE</password>
    </database>
    <!-- Different application personalizations -->
    <personalizations>
        <personalization>
            <name>first screen</name>
            <type>clustering</type> <!-- In the future it could be classification or traditional recommenda-
tion -->
            <personalization_option name="status">
                <parameter_option parameter_name="combat_profile">non combat</parameter_option>
                <parameter_option parameter_name="tournament_profile">non tournament</parameter_option>
            </personalization_option>
            <personalization_option name="combat">
                <parameter_option parameter_name="combat_profile">combat</parameter_option>
                <parameter_option parameter_name="tournament_profile">non tournament</parameter_option>
            </personalization_option>
            <personalization_option name="tournament">
                <parameter_option parameter_name="combat_profile">non combat</parameter_option>
                <parameter_option parameter_name="tournament_profile">tournament</parameter_option>
            </personalization_option>
        </personalization>
        <!-- MODIFICAR!!! -->
        <personalization>
            <name>alert level</name>
            <type>clustering</type>
            <context>
                <total_context_weight>0.25</total_context_weight>
                <context_resource>
                    <name>routineHours</name>
                    <weight>0.6</weight>
                </context_resource>
                <context_resource>
                    <name>situation</name>
                    <weight>0.4</weight>
                </context_resource>
            </context>
            <personalization_option name="single_basic">
                <parameter_option parameter_name="user level">basic</parameter_option>
                <parameter_option parameter_name="competition_level">non competitive</parameter_option>
            </personalization_option>
            <personalization_option name="single_advanced">
                <parameter_option parameter_name="user level">advanced</parameter_option>
                <parameter_option parameter_name="competition_level">non competitive</parameter_option>
            </personalization_option>
            <personalization_option name="multi_basic">
                <parameter_option parameter_name="user level">basic</parameter_option>
                <parameter_option parameter_name="competition_level">competitive</parameter_option>
            </personalization_option>
            <personalization_option name="multi_advanced">
                <parameter_option parameter_name="user level">advanced</parameter_option>
                <parameter_option parameter_name="competition_level">competitive</parameter_option>
            </personalization_option>
        </personalization>
        <personalization>
            <name>house image</name>
            <type>clustering</type>
            <personalization_option name="coward">
                <parameter_option parameter_name="cowardry_level">coward</parameter_option>
                <parameter_option parameter_name="competition_level">non competitive</parameter_option>
            </personalization_option>
            <personalization_option name="coward">
                <parameter_option parameter_name="cowardry_level">coward</parameter_option>
                <parameter_option parameter_name="competition_level">competitive</parameter_option>
            </personalization_option>
```

```
            <personalization_option name="defender">
                <parameter_option parameter_name="cowardry_level">non coward</parameter_option>
                <parameter_option parameter_name="competition_level">non competitive</parameter_option>
            </personalization_option>
            <personalization_option name="competitive">
                <parameter_option parameter_name="cowardry_level">non coward</parameter_option>
                <parameter_option parameter_name="competition_level">competitive</parameter_option>
            </personalization_option>
        </personalization>
        <personalization>
            <name>status background neighbourhood</name>
            <type>clustering</type>
            <personalization_option name="isolated sun">
                <parameter_option parameter_name="competition_level">non competitive</parameter_option>
                <external_service service_name="weather">sun</external_service>
            </personalization_option>
            <personalization_option name="neighbourhood sun">
                <parameter_option parameter_name="competition_level">competitive</parameter_option>
                <external_service service_name="weather">sun</external_service>
            </personalization_option>
            <personalization_option name="isolated clouds">
                <parameter_option parameter_name="competition_level">non competitive</parameter_option>
                <external_service service_name="weather">clouds</external_service>
            </personalization_option>
            <personalization_option name="neighbourhood clouds">
                <parameter_option parameter_name="competition_level">competitive</parameter_option>
                <external_service service_name="weather">clouds</external_service>
            </personalization_option>
            <personalization_option name="isolated rain">
                <parameter_option parameter_name="competition_level">non competitive</parameter_option>
                <external_service service_name="weather">rain</external_service>
            </personalization_option>
            <personalization_option name="neighbourhood rain">
                <parameter_option parameter_name="competition_level">competitive</parameter_option>
                <external_service service_name="weather">rain</external_service>
            </personalization_option>
        </personalization>
    </personalizations>
    <!-- Set of parameters used by each personalization as data, to retrieve the recommendation -->
    <parameters>
        <parameter>
            <name>combat_profile</name>
            <option>non combat</option>
            <option>combat</option>
            <data>
                <data_value>numberCombats/loginTime</data_value>
            </data>
        </parameter>
        <parameter>
            <name>tournament_profile</name>
            <option>non tournament</option>
            <option>tournament</option>
            <data>
                <data_value>numberTournaments/loginTime</data_value>
            </data>
        </parameter>
        <parameter>
            <name>user level</name>
            <option>basic</option>
            <option>advanced</option>
            <data>
                <data_value>(0.8*loginTime+0.2*statistics)*question1</data_value>
            </data>
        </parameter>
        <parameter>
            <name>competition_level</name>
            <option>non competitive</option>
            <option>competitive</option>
            <data>
                <data_value>0.35*(numberCombats/loginTime)+0.65*(numberTournaments/loginTime)</data_value>
            </data>
        </parameter>
        <!-- Rever formula -->
        <parameter>
            <name>cowardry_level</name>
            <option>non coward</option>
            <option>coward</option>
            <data>
                <data_value>0.35*(numberRejectedCombats/(numberCombats/loginTime))+0.65*(numberRejectedTourna-
ments/(numberTournaments/loginTime))</data_value>
            </data>
        </parameter>
    </parameters>
    <external_services>
        <service>
```

```
        <name>weather</name>
        <type>weather_basic</type>
        <default>sun</default>
    </service>
</external_services>
<!-- Application data. Mostly from the interaction stream. -->
<resources>
    <resource_appdata>
        <name>login</name>
        <upd_type>increment</upd_type>
    </resource_appdata>
    <resource_appdata>
        <name>loginTime</name>
        <upd_type>sum</upd_type>
    </resource_appdata>
    <resource_appdata>
        <name>statistics</name>
        <upd_type>increment</upd_type>
    </resource_appdata>
    <resource_appdata>
        <name>numberCombats</name>
        <upd_type>replace</upd_type>
    </resource_appdata>
    <resource_appdata>
        <name>numberTournaments</name>
        <upd_type>increment</upd_type>
    </resource_appdata>
    <resource_appdata>
        <name>numberRejectedCombats</name>
        <upd_type>increment</upd_type>
    </resource_appdata>
    <resource_appdata>
        <name>numberRejectedTournaments</name>
        <upd_type>increment</upd_type>
    </resource_appdata>
    <resource_context>
        <name>routineHours</name>
        <type>hour_interval</type>
        <option>
        <name>businessHours</name>
        <property>09:00-17:00</property>
        </option>
        <option>
        <name>nonBusinessHours</name>
        <property>17:00-09:00</property>
        </option>
    </resource_context>
    <resource_context>
        <name>situation</name>
        <type>temperature</type>
        <option>
        <name>Warm</name>
        <property>21-40</property>
        </option>
        <option>
        <name>Cold</name>
        <property>0-21</property>
        </option>
    </resource_context>
    <!-- For example: Environmental responsability -->
    <resource_preference>
        <name>question1</name>
        <type>question_MC</type>
        <option>
        <name>q1radio1</name>
        <value>0.8</value>
        </option>
        <option>
        <name>q1radio2</name>
        <value>0.9</value>
        </option>
        <option default="true">
        <name>q1radio3</name>
        <value>1</value>
        </option>
        <option>
        <name>q1radio4</name>
        <value>1.1</value>
        </option>
        <option>
        <name>q1radio5</name>
        <value>1.2</value>
        </option>
    </resource_preference>
</resources>
```

```
</xupil>
```

A.2. FCT4U (Study EU.1)

```
<?xml version="1.0" standalone="yes"?>
<xupil appname="fct4u" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLoca-
tion="file:///D:/Dropbox/myPhDthesis/Desenvolvimento/modelSchema.xsd">
    <!-- Application personalizations -->
    <personalizations>
        <personalization>
            <name>mapMode</name>
            <type>clustering</type>
            <personalization_option name="publicTransport">
                <parameter_option parameter_name="transportProfile">public</parameter_option>
            </personalization_option>
            <personalization_option name="privateTransport">
                <parameter_option parameter_name="transportProfile">private</parameter_option>
            </personalization_option>
        </personalization>
        <personalization>
            <name>privacyMode</name>
            <type>clustering</type>
            <personalization_option name="studentPublicTransport">
                <parameter_option parameter_name="userTypeProfile">student</parameter_option>
                <parameter_option parameter_name="transportProfile">public</parameter_option>
            </personalization_option>
            <personalization_option name="teacherPublicTransport">
                <parameter_option parameter_name="userTypeProfile">teacher</parameter_option>
                <parameter_option parameter_name="transportProfile">public</parameter_option>
            </personalization_option>
            <personalization_option name="studentPrivateTransport">
                <parameter_option parameter_name="userTypeProfile">student</parameter_option>
                <parameter_option parameter_name="transportProfile">private</parameter_option>
            </personalization_option>
            <personalization_option name="teacherPrivateTransport">
                <parameter_option parameter_name="userTypeProfile">teacher</parameter_option>
                <parameter_option parameter_name="transportProfile">private</parameter_option>
            </personalization_option>
            <personalization_option name="studentPublicTransportAlone">
                <parameter_option parameter_name="userTypeProfile">student</parameter_option>
                <parameter_option parameter_name="transportProfile">public</parameter_option>
                <external_service service_name="company">Alone</external_service>
            </personalization_option>
            <personalization_option name="teacherPublicTransportAlone">
                <parameter_option parameter_name="userTypeProfile">teacher</parameter_option>
                <parameter_option parameter_name="transportProfile">public</parameter_option>
                <external_service service_name="company">Alone</external_service>
            </personalization_option>
            <personalization_option name="studentPrivateTransportAlone">
                <parameter_option parameter_name="userTypeProfile">student</parameter_option>
                <parameter_option parameter_name="transportProfile">private</parameter_option>
                <external_service service_name="company">Alone</external_service>
            </personalization_option>
            <personalization_option name="teacherPrivateTransportAlone">
                <parameter_option parameter_name="userTypeProfile">teacher</parameter_option>
                <parameter_option parameter_name="transportProfile">private</parameter_option>
                <external_service service_name="company">Alone</external_service>
            </personalization_option>
            <personalization_option name="studentPublicTransportAccompanied">
                <parameter_option parameter_name="userTypeProfile">student</parameter_option>
                <parameter_option parameter_name="transportProfile">public</parameter_option>
                <external_service service_name="company">Accompanied</external_service>
            </personalization_option>
            <personalization_option name="teacherPublicTransportAccompanied">
                <parameter_option parameter_name="userTypeProfile">teacher</parameter_option>
                <parameter_option parameter_name="transportProfile">public</parameter_option>
                <external_service service_name="company">Accompanied</external_service>
            </personalization_option>
            <personalization_option name="studentPrivateTransportAccompanied">
                <parameter_option parameter_name="userTypeProfile">student</parameter_option>
                <parameter_option parameter_name="transportProfile">private</parameter_option>
                <external_service service_name="company">Accompanied</external_service>
            </personalization_option>
            <personalization_option name="teacherPrivateTransportAccompanied">
                <parameter_option parameter_name="userTypeProfile">teacher</parameter_option>
                <parameter_option parameter_name="transportProfile">private</parameter_option>
                <external_service service_name="company">Accompanied</external_service>
            </personalization_option>
            <personalization_option name="studentPublicTransportAccompanied w/friends">
                <parameter_option parameter_name="userTypeProfile">student</parameter_option>
```

```
        <parameter_option parameter_name="transportProfile">public</parameter_option>
        <external_service service_name="company">Accompanied w/friends</external_service>
    </personalization_option>
    <personalization_option name="teacherPublicTransportAccompanied w/friends">
        <parameter_option parameter_name="userTypeProfile">teacher</parameter_option>
        <parameter_option parameter_name="transportProfile">public</parameter_option>
        <external_service service_name="company">Accompanied w/friends</external_service>
    </personalization_option>
    <personalization_option name="studentPrivateTransportAccompanied w/friends">
        <parameter_option parameter_name="userTypeProfile">student</parameter_option>
        <parameter_option parameter_name="transportProfile">private</parameter_option>
        <external_service service_name="company">Accompanied w/friends</external_service>
    </personalization_option>
    <personalization_option name="teacherPrivateTransportAccompanied w/friends">
        <parameter_option parameter_name="userTypeProfile">teacher</parameter_option>
        <parameter_option parameter_name="transportProfile">private</parameter_option>
        <external_service service_name="company">Accompanied w/friends</external_service>
    </personalization_option>
</personalization>
<personalization>
    <name>greetingsMode</name>
    <type>clustering</type>
    <personalization_option name="inactiveStudent">
        <parameter_option parameter_name="userTypeProfile">student</parameter_option>
        <parameter_option parameter_name="userActiveness">low</parameter_option>
    </personalization_option>
    <personalization_option name="inactiveTeacher">
        <parameter_option parameter_name="userTypeProfile">teacher</parameter_option>
        <parameter_option parameter_name="userActiveness">low</parameter_option>
    </personalization_option>
    <personalization_option name="averageStudent">
        <parameter_option parameter_name="userTypeProfile">student</parameter_option>
        <parameter_option parameter_name="userActiveness">medium</parameter_option>
    </personalization_option>
    <personalization_option name="averageTeacher">
        <parameter_option parameter_name="userTypeProfile">teacher</parameter_option>
        <parameter_option parameter_name="userActiveness">medium</parameter_option>
    </personalization_option>
    <personalization_option name="activeStudent">
        <parameter_option parameter_name="userTypeProfile">student</parameter_option>
        <parameter_option parameter_name="userActiveness">high</parameter_option>
    </personalization_option>
    <personalization_option name="activeTeacher">
        <parameter_option parameter_name="userTypeProfile">teacher</parameter_option>
        <parameter_option parameter_name="userActiveness">high</parameter_option>
    </personalization_option>
</personalization>
<personalization>
    <name>viewOrder</name>
    <type>clustering</type>
    <context>
        <total_context_weight>0.5</total_context_weight>
        <context_resource>
            <name>lunchInterval</name>
            <weight>1.0</weight>
        </context_resource>
    </context>
    <personalization_option name="weatherCommunicativeMenusNewsMapAholic">
        <parameter_option parameter_name="weatherAholic">weatherAholic</parameter_option>
        <parameter_option parameter_name="communicative">communicative</parameter_option>
        <parameter_option parameter_name="menusAholic">menusAholic</parameter_option>
        <parameter_option parameter_name="newsAholic">newsAholic</parameter_option>
        <parameter_option parameter_name="mapAholic">mapAholic</parameter_option>
    </personalization_option>
    <personalization_option name="weatherMenusNewsMapAholic">
        <parameter_option parameter_name="weatherAholic">weatherAholic</parameter_option>
        <parameter_option parameter_name="communicative">notCommunicative</parameter_option>
        <parameter_option parameter_name="menusAholic">menusAholic</parameter_option>
        <parameter_option parameter_name="newsAholic">newsAholic</parameter_option>
        <parameter_option parameter_name="mapAholic">mapAholic</parameter_option>
    </personalization_option>
    <personalization_option name="communicativeMenusNewsMapAholic">
        <parameter_option parameter_name="weatherAholic">notWeatherAholic</parameter_option>
        <parameter_option parameter_name="communicative">communicative</parameter_option>
        <parameter_option parameter_name="menusAholic">menusAholic</parameter_option>
        <parameter_option parameter_name="newsAholic">newsAholic</parameter_option>
        <parameter_option parameter_name="mapAholic">mapAholic</parameter_option>
    </personalization_option>
    <personalization_option name="menusNewsMapAholic">
        <parameter_option parameter_name="weatherAholic">notWeatherAholic</parameter_option>
        <parameter_option parameter_name="communicative">notCommunicative</parameter_option>
        <parameter_option parameter_name="menusAholic">menusAholic</parameter_option>
        <parameter_option parameter_name="newsAholic">newsAholic</parameter_option>
        <parameter_option parameter_name="mapAholic">mapAholic</parameter_option>
    </personalization_option>
```

```
        <personalization_option name="weatherCommunicativeNewsMapAholic">
            <parameter_option parameter_name="weatherAholic">weatherAholic</parameter_option>
            <parameter_option parameter_name="communicative">communicative</parameter_option>
            <parameter_option parameter_name="menusAholic">notMenusAholic</parameter_option>
            <parameter_option parameter_name="newsAholic">newsAholic</parameter_option>
            <parameter_option parameter_name="mapAholic">mapAholic</parameter_option>
        </personalization_option>
        <personalization_option name="weatherNewsMapAholic">
            <parameter_option parameter_name="weatherAholic">weatherAholic</parameter_option>
            <parameter_option parameter_name="communicative">notCommunicative</parameter_option>
            <parameter_option parameter_name="menusAholic">notMenusAholic</parameter_option>
            <parameter_option parameter_name="newsAholic">newsAholic</parameter_option>
            <parameter_option parameter_name="mapAholic">mapAholic</parameter_option>
        </personalization_option>
        <personalization_option name="communicativeNewsMapAholic">
            <parameter_option parameter_name="weatherAholic">notWeatherAholic</parameter_option>
            <parameter_option parameter_name="communicative">communicative</parameter_option>
            <parameter_option parameter_name="menusAholic">notMenusAholic</parameter_option>
            <parameter_option parameter_name="newsAholic">newsAholic</parameter_option>
            <parameter_option parameter_name="mapAholic">mapAholic</parameter_option>
        </personalization_option>
        <personalization_option name="newsMapAholic">
            <parameter_option parameter_name="weatherAholic">notWeatherAholic</parameter_option>
            <parameter_option parameter_name="communicative">notCommunicative</parameter_option>
            <parameter_option parameter_name="menusAholic">notMenusAholic</parameter_option>
            <parameter_option parameter_name="newsAholic">newsAholic</parameter_option>
            <parameter_option parameter_name="mapAholic">mapAholic</parameter_option>
        </personalization_option>
        <personalization_option name="weatherCommunicativeMenusMapAholic">
            <parameter_option parameter_name="weatherAholic">weatherAholic</parameter_option>
            <parameter_option parameter_name="communicative">communicative</parameter_option>
            <parameter_option parameter_name="menusAholic">menusAholic</parameter_option>
            <parameter_option parameter_name="newsAholic">notNewsAholic</parameter_option>
            <parameter_option parameter_name="mapAholic">mapAholic</parameter_option>
        </personalization_option>
        <personalization_option name="weatherMenusMapAholic">
            <parameter_option parameter_name="weatherAholic">weatherAholic</parameter_option>
            <parameter_option parameter_name="communicative">notCommunicative</parameter_option>
            <parameter_option parameter_name="menusAholic">menusAholic</parameter_option>
            <parameter_option parameter_name="newsAholic">notNewsAholic</parameter_option>
            <parameter_option parameter_name="mapAholic">mapAholic</parameter_option>
        </personalization_option>
        <personalization_option name="communicativeMenusMapAholic">
            <parameter_option parameter_name="weatherAholic">notWeatherAholic</parameter_option>
            <parameter_option parameter_name="communicative">communicative</parameter_option>
            <parameter_option parameter_name="menusAholic">menusAholic</parameter_option>
            <parameter_option parameter_name="newsAholic">notNewsAholic</parameter_option>
            <parameter_option parameter_name="mapAholic">mapAholic</parameter_option>
        </personalization_option>
        <personalization_option name="menusMapAholic">
            <parameter_option parameter_name="weatherAholic">notWeatherAholic</parameter_option>
            <parameter_option parameter_name="communicative">notCommunicative</parameter_option>
            <parameter_option parameter_name="menusAholic">menusAholic</parameter_option>
            <parameter_option parameter_name="newsAholic">notNewsAholic</parameter_option>
            <parameter_option parameter_name="mapAholic">mapAholic</parameter_option>
        </personalization_option>
        <personalization_option name="weatherCommunicativeMapAholic">
            <parameter_option parameter_name="weatherAholic">weatherAholic</parameter_option>
            <parameter_option parameter_name="communicative">communicative</parameter_option>
            <parameter_option parameter_name="menusAholic">notMenusAholic</parameter_option>
            <parameter_option parameter_name="newsAholic">notNewsAholic</parameter_option>
            <parameter_option parameter_name="mapAholic">mapAholic</parameter_option>
        </personalization_option>
        <personalization_option name="weatherMapAholic">
            <parameter_option parameter_name="weatherAholic">weatherAholic</parameter_option>
            <parameter_option parameter_name="communicative">notCommunicative</parameter_option>
            <parameter_option parameter_name="menusAholic">notMenusAholic</parameter_option>
            <parameter_option parameter_name="newsAholic">notNewsAholic</parameter_option>
            <parameter_option parameter_name="mapAholic">mapAholic</parameter_option>
        </personalization_option>
        <personalization_option name="communicativeMapAholic">
            <parameter_option parameter_name="weatherAholic">notWeatherAholic</parameter_option>
            <parameter_option parameter_name="communicative">communicative</parameter_option>
            <parameter_option parameter_name="menusAholic">notMenusAholic</parameter_option>
            <parameter_option parameter_name="newsAholic">notNewsAholic</parameter_option>
            <parameter_option parameter_name="mapAholic">mapAholic</parameter_option>
        </personalization_option>
        <personalization_option name="mapAholic">
            <parameter_option parameter_name="weatherAholic">notWeatherAholic</parameter_option>
            <parameter_option parameter_name="communicative">notCommunicative</parameter_option>
            <parameter_option parameter_name="menusAholic">notMenusAholic</parameter_option>
            <parameter_option parameter_name="newsAholic">notNewsAholic</parameter_option>
            <parameter_option parameter_name="mapAholic">mapAholic</parameter_option>
        </personalization_option>
        <personalization_option name="weatherCommunicativeMenusNewsAholic">
```

```
        <parameter_option parameter_name="weatherAholic">weatherAholic</parameter_option>
        <parameter_option parameter_name="communicative">communicative</parameter_option>
        <parameter_option parameter_name="menusAholic">menusAholic</parameter_option>
        <parameter_option parameter_name="newsAholic">newsAholic</parameter_option>
        <parameter_option parameter_name="mapAholic">notMapAholic</parameter_option>
    </personalization_option>
    <personalization_option name="weatherMenusNewsAholic">
        <parameter_option parameter_name="weatherAholic">weatherAholic</parameter_option>
        <parameter_option parameter_name="communicative">notCommunicative</parameter_option>
        <parameter_option parameter_name="menusAholic">menusAholic</parameter_option>
        <parameter_option parameter_name="newsAholic">newsAholic</parameter_option>
        <parameter_option parameter_name="mapAholic">notMapAholic</parameter_option>
    </personalization_option>
    <personalization_option name="communicativeMenusNewsAholic">
        <parameter_option parameter_name="weatherAholic">notWeatherAholic</parameter_option>
        <parameter_option parameter_name="communicative">communicative</parameter_option>
        <parameter_option parameter_name="menusAholic">menusAholic</parameter_option>
        <parameter_option parameter_name="newsAholic">newsAholic</parameter_option>
        <parameter_option parameter_name="mapAholic">notMapAholic</parameter_option>
    </personalization_option>
    <personalization_option name="menusNewsAholic">
        <parameter_option parameter_name="weatherAholic">notWeatherAholic</parameter_option>
        <parameter_option parameter_name="communicative">notCommunicative</parameter_option>
        <parameter_option parameter_name="menusAholic">menusAholic</parameter_option>
        <parameter_option parameter_name="newsAholic">newsAholic</parameter_option>
        <parameter_option parameter_name="mapAholic">notMapAholic</parameter_option>
    </personalization_option>
    <personalization_option name="weatherCommunicativeNewsAholic">
        <parameter_option parameter_name="weatherAholic">weatherAholic</parameter_option>
        <parameter_option parameter_name="communicative">communicative</parameter_option>
        <parameter_option parameter_name="menusAholic">notMenusAholic</parameter_option>
        <parameter_option parameter_name="newsAholic">newsAholic</parameter_option>
        <parameter_option parameter_name="mapAholic">notMapAholic</parameter_option>
    </personalization_option>
    <personalization_option name="weatherNewsAholic">
        <parameter_option parameter_name="weatherAholic">weatherAholic</parameter_option>
        <parameter_option parameter_name="communicative">notCommunicative</parameter_option>
        <parameter_option parameter_name="menusAholic">notMenusAholic</parameter_option>
        <parameter_option parameter_name="newsAholic">newsAholic</parameter_option>
        <parameter_option parameter_name="mapAholic">notMapAholic</parameter_option>
    </personalization_option>
    <personalization_option name="communicativeNewsAholic">
        <parameter_option parameter_name="weatherAholic">notWeatherAholic</parameter_option>
        <parameter_option parameter_name="communicative">communicative</parameter_option>
        <parameter_option parameter_name="menusAholic">notMenusAholic</parameter_option>
        <parameter_option parameter_name="newsAholic">newsAholic</parameter_option>
        <parameter_option parameter_name="mapAholic">notMapAholic</parameter_option>
    </personalization_option>
    <personalization_option name="newsAholic">
        <parameter_option parameter_name="weatherAholic">notWeatherAholic</parameter_option>
        <parameter_option parameter_name="communicative">notCommunicative</parameter_option>
        <parameter_option parameter_name="menusAholic">notMenusAholic</parameter_option>
        <parameter_option parameter_name="newsAholic">newsAholic</parameter_option>
        <parameter_option parameter_name="mapAholic">notMapAholic</parameter_option>
    </personalization_option>
    <personalization_option name="weatherCommunicativeMenusAholic">
        <parameter_option parameter_name="weatherAholic">weatherAholic</parameter_option>
        <parameter_option parameter_name="communicative">communicative</parameter_option>
        <parameter_option parameter_name="menusAholic">menusAholic</parameter_option>
        <parameter_option parameter_name="newsAholic">notNewsAholic</parameter_option>
        <parameter_option parameter_name="mapAholic">notMapAholic</parameter_option>
    </personalization_option>
    <personalization_option name="weatherMenusAholic">
        <parameter_option parameter_name="weatherAholic">weatherAholic</parameter_option>
        <parameter_option parameter_name="communicative">notCommunicative</parameter_option>
        <parameter_option parameter_name="menusAholic">menusAholic</parameter_option>
        <parameter_option parameter_name="newsAholic">notNewsAholic</parameter_option>
        <parameter_option parameter_name="mapAholic">notMapAholic</parameter_option>
    </personalization_option>
    <personalization_option name="communicativeMenusAholic">
        <parameter_option parameter_name="weatherAholic">notWeatherAholic</parameter_option>
        <parameter_option parameter_name="communicative">communicative</parameter_option>
        <parameter_option parameter_name="menusAholic">menusAholic</parameter_option>
        <parameter_option parameter_name="newsAholic">notNewsAholic</parameter_option>
        <parameter_option parameter_name="mapAholic">notMapAholic</parameter_option>
    </personalization_option>
    <personalization_option name="menusAholic">
        <parameter_option parameter_name="weatherAholic">notWeatherAholic</parameter_option>
        <parameter_option parameter_name="communicative">notCommunicative</parameter_option>
        <parameter_option parameter_name="menusAholic">menusAholic</parameter_option>
        <parameter_option parameter_name="newsAholic">notNewsAholic</parameter_option>
        <parameter_option parameter_name="mapAholic">notMapAholic</parameter_option>
    </personalization_option>
    <personalization_option name="weatherCommunicativeAholic">
        <parameter_option parameter_name="weatherAholic">weatherAholic</parameter_option>
```

```
                <parameter_option parameter_name="communicative">communicative</parameter_option>
                <parameter_option parameter_name="menusAholic">notMenusAholic</parameter_option>
                <parameter_option parameter_name="newsAholic">notNewsAholic</parameter_option>
                <parameter_option parameter_name="mapAholic">notMapAholic</parameter_option>
            </personalization_option>
            <personalization_option name="weatherAholic">
                <parameter_option parameter_name="weatherAholic">weatherAholic</parameter_option>
                <parameter_option parameter_name="communicative">notCommunicative</parameter_option>
                <parameter_option parameter_name="menusAholic">notMenusAholic</parameter_option>
                <parameter_option parameter_name="newsAholic">notNewsAholic</parameter_option>
                <parameter_option parameter_name="mapAholic">notMapAholic</parameter_option>
            </personalization_option>
            <personalization_option name="communicative">
                <parameter_option parameter_name="weatherAholic">notWeatherAholic</parameter_option>
                <parameter_option parameter_name="communicative">communicative</parameter_option>
                <parameter_option parameter_name="menusAholic">notMenusAholic</parameter_option>
                <parameter_option parameter_name="newsAholic">notNewsAholic</parameter_option>
                <parameter_option parameter_name="mapAholic">notMapAholic</parameter_option>
            </personalization_option>
            <personalization_option name="none">
                <parameter_option parameter_name="weatherAholic">notWeatherAholic</parameter_option>
                <parameter_option parameter_name="communicative">notCommunicative</parameter_option>
                <parameter_option parameter_name="menusAholic">notMenusAholic</parameter_option>
                <parameter_option parameter_name="newsAholic">notNewsAholic</parameter_option>
                <parameter_option parameter_name="mapAholic">notMapAholic</parameter_option>
            </personalization_option>
        </personalization>
    </personalizations>
    <!-- Set of parameters used by each personalization as data, to retrieve the recommendation -->
    <parameters>
        <parameter>
            <name>userActiveness</name>
            <option>low</option>
            <option>medium</option>
            <option>high</option>
            <data>
                <data_value>0.5*uses+0.3*interactionCount+0.2*detectionCount</data_value>
            </data>
        </parameter>
        <parameter>
            <name>transportProfile</name>
            <option>private</option>
            <option>public</option>
            <data>
                <data_value>0.7*transportType+0.2*nearPublicTransportHubs+0.1*userType</data_value>
            </data>
        </parameter>
        <parameter>
            <name>userTypeProfile</name>
            <option>student</option>
            <option>teacher</option>
            <data>
                <data_value>0.9*userType+0.1*transportType</data_value>
            </data>
        </parameter>
        <parameter>
            <name>communicative</name>
            <option>notCommunicative</option>
            <option>communicative</option>
            <data>
                <data_value>0.2*messagesRead+0.3*messagesReceived+0.5*messagesSent</data_value>
            </data>
        </parameter>
        <parameter>
            <name>menusAholic</name>
            <option>notMenusAholic</option>
            <option>menusAholic</option>
            <data>
                <data_value>1.0*menuInteractions</data_value>
            </data>
        </parameter>
        <parameter>
            <name>newsAholic</name>
            <option>notNewsAholic</option>
            <option>newsAholic</option>
            <data>
                <data_value>0.7*newsViewed+0.3*newsClicked</data_value>
            </data>
        </parameter>
        <parameter>
            <name>weatherAholic</name>
            <option>notWeatherAholic</option>
            <option>weatherAholic</option>
            <data>
                <data_value>1.0*weatherInteractions</data_value>
```

```
            </data>
        </parameter>
        <parameter>
            <name>mapAholic</name>
            <option>notMapAholic</option>
            <option>mapAholic</option>
            <data>
                <data_value>0.6*mapInteractions+0.4*mapDetailInteractions</data_value>
            </data>
        </parameter>
    </parameters>
    <!-- Application data. Mostly from the interaction stream. -->
    <resources>
        <!-- ... -->
        <resource_appdata>
            <name>detectionCount</name>
            <upd_type>increment</upd_type>
        </resource_appdata>
        <!-- ... -->
        <resource_appdata>
            <name>uses</name>
            <upd_type>increment</upd_type>
        </resource_appdata>
        <!-- Counts all interactions -->
        <resource_appdata>
            <name>interactionCount</name>
            <upd_type>increment</upd_type>
        </resource_appdata>
        <!-- Number of messages sent by a user -->
        <resource_appdata>
            <name>messagesSent</name>
            <upd_type>increment</upd_type>
        </resource_appdata>
        <!-- Number of messages sent by a user -->
        <resource_appdata>
            <name>messagesReceived</name>
            <upd_type>increment</upd_type>
        </resource_appdata>
        <!-- Number of messages marked as read by a user -->
        <resource_appdata>
            <name>messagesRead</name>
            <upd_type>sum</upd_type>
        </resource_appdata>
        <!-- Number of news that a user clicks to show the link on the mobile device and the QR Code on the
public display -->
        <resource_appdata>
            <name>newsClicked</name>
            <upd_type>increment</upd_type>
        </resource_appdata>
        <!-- Number of news that a user actually views by clicking on the previously shown link -->
        <resource_appdata>
            <name>newsViewed</name>
            <upd_type>increment</upd_type>
        </resource_appdata>
        <!-- ... -->
        <resource_appdata>
            <name>weatherInteractions</name>
            <upd_type>increment</upd_type>
        </resource_appdata>
        <!-- ... -->
        <resource_appdata>
            <name>menuInteractions</name>
            <upd_type>increment</upd_type>
        </resource_appdata>
        <!-- ... -->
        <resource_appdata>
            <name>mapInteractions</name>
            <upd_type>increment</upd_type>
        </resource_appdata>
        <!-- ... -->
        <resource_appdata>
            <name>mapDetailInteractions</name>
            <upd_type>increment</upd_type>
        </resource_appdata>
        <!-- ... -->
        <resource_appdata>
            <name>nearPublicTransportHubs</name>
            <upd_type>sum</upd_type>
        </resource_appdata>
        <!-- ... -->
        <resource_context>
            <name>lunchInterval</name>
            <type>hour_interval</type>
            <option>
            <name>preLunch</name>
```

```
            <property>08:00-14:00</property>
            </option>
            <option>
            <name>afterLunch</name>
            <property>14:00-08:00</property>
            </option>
        </resource_context>
        <!-- ... -->
        <resource_context>
            <name>company</name>
            <type>value</type>
            <option>
            <name>Alone</name>
            <property>0</property>
            </option>
            <option>
            <name>Accompanied</name>
            <property>1</property>
            </option>
            <option>
            <name>Accompanied w/friends</name>
            <property>2</property>
            </option>
        </resource_context>
        <!-- The user type is a preference. It can be a teacher/researcher/staff or a student, but by default
student is chosen -->
        <resource_preference>
            <name>userType</name>
            <type>question_MC</type>
            <option default="true">
            <name>student</name>
            <value>1.0</value>
            </option>
            <option>
            <name>teacher</name>
            <value>2.0</value>
            </option>
        </resource_preference>
        <!-- The transport type is a preference. It can be a teacher or a student, but by default student is
chosen -->
        <resource_preference>
            <name>transportType</name>
            <type>question_MC</type>
            <option>
            <name>public</name>
            <value>1.0</value>
            </option>
            <option default="true">
            <name>private</name>
            <value>2.0</value>
            </option>
        </resource_preference>
    </resources>
</xupil>
```

A.3. WeSync (Study EU.2)

```
<?xml version='1.0' standalone='yes'?>
<xupil appname="wesync" accessable="protected">
<!-- accessable=protected means other apps from same developer (team) are able to reuse the entire personali-
sation definition with all data from its users-->
    <!-- Application personalizations -->
    <personalizations>
        <personalization>
            <name>delays</name>
            <type>clustering</type>
            <personalization_option name="cheater">
                <parameter_option parameter_name="gamerProfile">high</parameter_option>
                <parameter_option parameter_name="syncerProfile">wow</parameter_option>
            </personalization_option>
            <personalization_option name="non-cheater tier-1">
                <parameter_option parameter_name="gamerProfile">low</parameter_option>
                <parameter_option parameter_name="syncerProfile">itsok</parameter_option>
            </personalization_option>
            <personalization_option name="non-cheater tier-2">
                <parameter_option parameter_name="gamerProfile">high</parameter_option>
                <parameter_option parameter_name="syncerProfile">itsok</parameter_option>
            </personalization_option>
```

```
            <personalization_option name="non-cheater tier-3">
                <parameter_option parameter_name="gamerProfile">low</parameter_option>
                <parameter_option parameter_name="syncerProfile">wow</parameter_option>
            </personalization_option>
        </personalization>
    </personalizations>
    <!-- Set of parameters used by each personalization as data, to retrieve the recommendation -->
    <parameters>
        <parameter>
            <name>gamerProfile</name>
            <option>low</option>
            <option>high</option>
            <data>
                <data_value>0.3/((totalResponseTime)/keyEvents)+0.7*(totalCorrec-
tResults/keyEvents)</data_value>
            </data>
        </parameter>
        <parameter>
            <name>syncerProfile</name>
            <option>itsok</option>
            <option>wow</option>
            <data>
                <data_value>0.5 * currentUserDelay + 0.25 * (deltaUserDelay/syncKeyEvents) + 0.25 * (to-
talUserDelay/syncKeyEvents)</data_value>
            </data>
        </parameter>
    </parameters>

    <!-- Application data. Mostly from the interaction stream. -->
    <resources>
        <resource_appdata>
            <name>totalResponseTime</name>
            <upd_type>sum</upd_type>
        </resource_appdata>

        <!--<resource_appdata>
                <name>lastResponseTime</name>
                <upd_type>replace</upd_type>
        </resource_appdata>-->

        <resource_appdata>
            <name>keyEvents</name>
            <upd_type>increment</upd_type>
        </resource_appdata>


        <resource_appdata>
            <name>syncKeyEvents</name>
            <upd_type>sum</upd_type>
        </resource_appdata>

        <resource_appdata>
            <name>totalCorrectResults</name>
            <upd_type>sum</upd_type>
        </resource_appdata>

        <resource_appdata>
            <name>currentUserDelay</name>
            <upd_type>replace</upd_type>
        </resource_appdata>

        <resource_appdata>
            <name>totalUserDelay</name>
            <upd_type>sum</upd_type>
        </resource_appdata>

        <resource_appdata>
            <name>deltaUserDelay</name>
            <upd_type>replace</upd_type>
        </resource_appdata>

        <!--<resource_appdata>
                <name>totalQuestionTime</name>
                <upd_type>sum</upd_type>
        </resource_appdata>-->

        <!--<resource_appdata>
                <name>lastQuestionTime</name>
                <upd_type>replace</upd_type>
        </resource_appdata>-->
    </resources>
</xupil>
```

A.4. WePlayTennis (Study EU.2)

```
<?xml version='1.0' standalone='yes'?>
<xupil appname="weplaytennis" refapp="wesync"/>
<!-- All Application personalizations, parameters and resources come from WeSync, who shares them  -->
<!-- All users and their data related to the resources of WeSync will be automatically used by WePlayTennis-->
```

A.5. "just Physio kidding" (Study EU.3)

```
<?xml version='1.0' standalone='yes'?>
<xupil appname="JPK">
    <!-- Application personalizations -->
    <personalizations>
        <!-- Personalization 1: referente a alterar elementos para motivar/persuadir a usar o JPK-->
        <personalization>
            <name>motivation</name>
            <type>clustering</type>
            <personalization_option name="highMotivationElements">
                <parameter_option parameter_name="predispositionProfile">low</parameter_option>
            </personalization_option>
            <personalization_option name="motivationElements">
                <parameter_option parameter_name="predispositionProfile">medium</parameter_option>
            </personalization_option>
            <personalization_option name="standardElements">
                <parameter_option parameter_name="predispositionProfile">high</parameter_option>
            </personalization_option>
        </personalization>

        <!-- Personalization 2: afetará a visibilidade dos elementos de time e score durante a execução de um
challenge JPK-->
        <personalization>
            <name>pressure</name>
            <type>clustering</type>
            <personalization_option name="timeONScoreON">
                <parameter_option parameter_name="predispositionProfile">high</parameter_option>
                <!--<parameter_option parameter_name="performanceProfile">high or low</parameter_option>-->
            </personalization_option>
            <personalization_option name="timeONScoreOFF">
                <parameter_option parameter_name="predispositionProfile">medium</parameter_option>
                <parameter_option parameter_name="performanceProfile">high</parameter_option>
            </personalization_option>
            <personalization_option name="timeOFFScoreON">
                <parameter_option parameter_name="predispositionProfile">medium</parameter_option>
                <parameter_option parameter_name="performanceProfile">low</parameter_option>
            </personalization_option>
            <personalization_option name="timeOFFScoreOFF">
                <parameter_option parameter_name="predispositionProfile">low</parameter_option>
                <!--        <parameter_option parameter_name="performanceProfile">high</parameter_option> -->
            </personalization_option>

        </personalization>

        <personalization>
            <name>competition</name>
            <type>clustering</type>
            <personalization_option name="globalRanking">
                <parameter_option parameter_name="competitiveProfile">competitive</parameter_option>
                <!--        parameter "psychoProfile" irrelevant here -->
                <parameter_option parameter_name="performanceProfile">high</parameter_option>
            </personalization_option>
            <personalization_option name="selfRanking">
                <parameter_option parameter_name="competitiveProfile">competitive</parameter_option>
                <parameter_option parameter_name="psychoProfile">OK</parameter_option>
                <parameter_option parameter_name="performanceProfile">low</parameter_option>
            </personalization_option>
            <personalization_option name="selfRanking">
                <parameter_option parameter_name="competitiveProfile">nonCompetitive</parameter_option>
                <parameter_option parameter_name="psychoProfile">OK</parameter_option>
                <parameter_option parameter_name="performanceProfile">high</parameter_option>
            </personalization_option>
            <personalization_option name="gameResult">
                <parameter_option parameter_name="competitiveProfile">competitive</parameter_option>
                <parameter_option parameter_name="psychoProfile">KO</parameter_option>
                <parameter_option parameter_name="performanceProfile">low</parameter_option>
            </personalization_option>
            <personalization_option name="gameResult">
                <parameter_option parameter_name="competitiveProfile">nonCompetitive</parameter_option>
                <parameter_option parameter_name="psychoProfile">KO</parameter_option>
```

```
                <parameter_option parameter_name="performanceProfile">high</parameter_option>
            </personalization_option>
            <personalization_option name="nothing">
                <parameter_option parameter_name="competitiveProfile">nonCompetitive</parameter_option>
                <!--        parameter "psychoProfile" irrelevant here -->
                <parameter_option parameter_name="performanceProfile">low</parameter_option>
            </personalization_option>
        </personalization>
    </personalizations>

    <!-- Set of parameters used by each personalization as data, to retrieve the recommendation -->
    <parameters>
        <parameter>
            <name>predispositionProfile</name>
            <option>low</option>
            <option>medium</option>
            <option>high</option>
            <data>
                <data_value>1/clinicalConditionSeverity * 0.2 + 1/emotionalState * 0.4 + motivationLevel *
0.4</data_value>
            </data>
        </parameter>

        <parameter>
            <name>performanceProfile</name>
            <option>low</option>
            <option>high</option>
            <data>
                <data_value>1/ranking * 0.5 + meanLast3challenges * 0.3 + lastChallengeScore *
0.2</data_value>
            </data>
        </parameter>

        <parameter>
            <name>competitiveProfile</name>
            <option>nonCompetitive</option>
            <option>competitive</option>
            <data>
                <data_value>(resultsScreenTimeSpent/resultsScreenConsults) * 0.5 + resultsScreenConsults * 0.2
+ badgesTrophiesScreenConsults * 0.3</data_value>
            </data>
        </parameter>

        <parameter>
            <name>psychoProfile</name>
            <option>KO</option>
            <option>OK</option>
            <data>
                <data_value>(6-emotionalState) * 0.5 + motivationLevel * 0.5</data_value>
            </data>
        </parameter>

    </parameters>

    <!-- Application data. Mostly from the interaction stream. -->
    <resources>
        <!-- always updated by physiotherapist -->
        <resource_preference>
            <name>emotionalState</name>
            <type>question_MC</type>
            <option>
            <name>very low</name>
            <value>1</value>
            </option>
            <option>
            <name>low</name>
            <value>2</value>
            </option>
            <option default="true">
            <name>regular</name>
            <value>3</value>
            </option>
            <option>
            <name>high</name>
            <value>4</value>
            </option>
            <option>
            <name>very high</name>
            <value>5</value>
            </option>
        </resource_preference>

        <!-- always updated by physiotherapist -->
        <resource_preference>
            <name>clinicalConditionSeverity</name>
```

```
            <type>question_MC</type>
            <option>
            <name>very low</name>
            <value>1</value>
            </option>
            <option>
            <name>low</name>
            <value>2</value>
            </option>
            <option default="true">
            <name>regular</name>
            <value>3</value>
            </option>
            <option>
            <name>high</name>
            <value>4</value>
            </option>
            <option>
            <name>very high</name>
            <value>5</value>
            </option>
        </resource_preference>

        <!-- always updated by physiotherapist -->
        <resource_preference>
            <name>motivationLevel</name>
            <type>question_MC</type>
            <option>
            <name>very low</name>
            <value>1</value>
            </option>
            <option>
            <name>low</name>
            <value>2</value>
            </option>
            <option default="true">
            <name>regular</name>
            <value>3</value>
            </option>
            <option>
            <name>high</name>
            <value>4</value>
            </option>
            <option>
            <name>very high</name>
            <value>5</value>
            </option>
        </resource_preference>

        <!-- computed and updated after each challenge execution -->
        <resource_appdata>
            <name>ranking</name>
            <upd_type>replace</upd_type>
        </resource_appdata>

        <!-- updated after a challenge execution -->
        <resource_appdata>
            <name>lastChallengeScore</name>
            <upd_type>replace</upd_type>
        </resource_appdata>

        <resource_appdata>
            <name>meanLast3challenges</name>
            <upd_type>replace</upd_type>
        </resource_appdata>

        <resource_appdata>
            <name>resultsScreenTimeSpent</name>
            <upd_type>sum</upd_type>
        </resource_appdata>

        <resource_appdata>
            <name>resultsScreenConsults</name>
            <upd_type>increment</upd_type>
        </resource_appdata>

        <resource_appdata>
            <name>badgesTrophiesScreenConsults</name>
            <upd_type>increment</upd_type>
        </resource_appdata>

    </resources>
</xupil>
```

A.6. Walker (Study D.2)

```
<?xml version='1.0' standalone='yes'?>
<xupil appname="JPK">
    <!-- Application personalizations -->
    <personalizations>
        <!-- Personalization 1: referente a alterar elementos para motivar/persuadir a usar o JPK-->
        <personalization>
            <name>motivation</name>
            <type>clustering</type>
            <personalization_option name="highMotivationElements">
                <parameter_option parameter_name="predispositionProfile">low</parameter_option>
            </personalization_option>
            <personalization_option name="motivationElements">
                <parameter_option parameter_name="predispositionProfile">medium</parameter_option>
            </personalization_option>
            <personalization_option name="standardElements">
                <parameter_option parameter_name="predispositionProfile">high</parameter_option>
            </personalization_option>
        </personalization>

        <!-- Personalization 2: afetará a visibilidade dos elementos de time e score durante a execução de um
challenge JPK-->
        <personalization>
            <name>pressure</name>
            <type>clustering</type>
            <personalization_option name="timeONScoreON">
                <parameter_option parameter_name="predispositionProfile">high</parameter_option>
                <!--<parameter_option parameter_name="performanceProfile">high or low</parameter_option>-->
            </personalization_option>
            <personalization_option name="timeONScoreOFF">
                <parameter_option parameter_name="predispositionProfile">medium</parameter_option>
                <parameter_option parameter_name="performanceProfile">high</parameter_option>
            </personalization_option>
            <personalization_option name="timeOFFScoreON">
                <parameter_option parameter_name="predispositionProfile">medium</parameter_option>
                <parameter_option parameter_name="performanceProfile">low</parameter_option>
            </personalization_option>
            <personalization_option name="timeOFFScoreOFF">
                <parameter_option parameter_name="predispositionProfile">low</parameter_option>
                <!--        <parameter_option parameter_name="performanceProfile">high</parameter_option> -->
            </personalization_option>

        </personalization>

        <personalization>
            <name>competition</name>
            <type>clustering</type>
            <personalization_option name="globalRanking">
                <parameter_option parameter_name="competitiveProfile">competitive</parameter_option>
                <!--        parameter "psychoProfile" irrelevant here -->
                <parameter_option parameter_name="performanceProfile">high</parameter_option>
            </personalization_option>
            <personalization_option name="selfRanking">
                <parameter_option parameter_name="competitiveProfile">competitive</parameter_option>
                <parameter_option parameter_name="psychoProfile">OK</parameter_option>
                <parameter_option parameter_name="performanceProfile">low</parameter_option>
            </personalization_option>
            <personalization_option name="selfRanking">
                <parameter_option parameter_name="competitiveProfile">nonCompetitive</parameter_option>
                <parameter_option parameter_name="psychoProfile">OK</parameter_option>
                <parameter_option parameter_name="performanceProfile">high</parameter_option>
            </personalization_option>
            <personalization_option name="gameResult">
                <parameter_option parameter_name="competitiveProfile">competitive</parameter_option>
                <parameter_option parameter_name="psychoProfile">KO</parameter_option>
                <parameter_option parameter_name="performanceProfile">low</parameter_option>
            </personalization_option>
            <personalization_option name="gameResult">
                <parameter_option parameter_name="competitiveProfile">nonCompetitive</parameter_option>
                <parameter_option parameter_name="psychoProfile">KO</parameter_option>
                <parameter_option parameter_name="performanceProfile">high</parameter_option>
            </personalization_option>
            <personalization_option name="nothing">
                <parameter_option parameter_name="competitiveProfile">nonCompetitive</parameter_option>
                <!--        parameter "psychoProfile" irrelevant here -->
                <parameter_option parameter_name="performanceProfile">low</parameter_option>
            </personalization_option>
        </personalization>
    </personalizations>

    <!-- Set of parameters used by each personalization as data, to retrieve the recommendation -->
    <parameters>
```

```
        <parameter>
            <name>predispositionProfile</name>
            <option>low</option>
            <option>medium</option>
            <option>high</option>
            <data>
                <data_value>1/clinicalConditionSeverity * 0.2 + 1/emotionalState * 0.4 + motivationLevel *
0.4</data_value>
            </data>
        </parameter>

        <parameter>
            <name>performanceProfile</name>
            <option>low</option>
            <option>high</option>
            <data>
                <data_value>1/ranking * 0.5 + meanLast3challenges * 0.3 + lastChallengeScore *
0.2</data_value>
            </data>
        </parameter>

        <parameter>
            <name>competitiveProfile</name>
            <option>nonCompetitive</option>
            <option>competitive</option>
            <data>
                <data_value>(resultsScreenTimeSpent/resultsScreenConsults) * 0.5 + resultsScreenConsults * 0.2
+ badgesTrophiesScreenConsults * 0.3</data_value>
            </data>
        </parameter>

        <parameter>
            <name>psychoProfile</name>
            <option>KO</option>
            <option>OK</option>
            <data>
                <data_value>(6-emotionalState) * 0.5 + motivationLevel * 0.5</data_value>
            </data>
        </parameter>

    </parameters>

    <!-- Application data. Mostly from the interaction stream. -->
    <resources>
        <!-- always updated by physiotherapist -->
        <resource_preference>
            <name>emotionalState</name>
            <type>question_MC</type>
            <option>
            <name>very low</name>
            <value>1</value>
            </option>
            <option>
            <name>low</name>
            <value>2</value>
            </option>
            <option default="true">
            <name>regular</name>
            <value>3</value>
            </option>
            <option>
            <name>high</name>
            <value>4</value>
            </option>
            <option>
            <name>very high</name>
            <value>5</value>
            </option>
        </resource_preference>

        <!-- always updated by physiotherapist -->
        <resource_preference>
            <name>clinicalConditionSeverity</name>
            <type>question_MC</type>
            <option>
            <name>very low</name>
            <value>1</value>
            </option>
            <option>
            <name>low</name>
            <value>2</value>
            </option>
            <option default="true">
            <name>regular</name>
            <value>3</value>
```

```
            </option>
            <option>
            <name>high</name>
            <value>4</value>
            </option>
            <option>
            <name>very high</name>
            <value>5</value>
            </option>
        </resource_preference>

        <!-- always updated by physiotherapist -->
        <resource_preference>
            <name>motivationLevel</name>
            <type>question_MC</type>
            <option>
            <name>very low</name>
            <value>1</value>
            </option>
            <option>
            <name>low</name>
            <value>2</value>
            </option>
            <option default="true">
            <name>regular</name>
            <value>3</value>
            </option>
            <option>
            <name>high</name>
            <value>4</value>
            </option>
            <option>
            <name>very high</name>
            <value>5</value>
            </option>
        </resource_preference>

        <!-- computed and updated after each challenge execution -->
        <resource_appdata>
            <name>ranking</name>
            <upd_type>replace</upd_type>
        </resource_appdata>

        <!-- updated after a challenge execution -->
        <resource_appdata>
            <name>lastChallengeScore</name>
            <upd_type>replace</upd_type>
        </resource_appdata>

        <resource_appdata>
            <name>meanLast3challenges</name>
            <upd_type>replace</upd_type>
        </resource_appdata>

        <resource_appdata>
            <name>resultsScreenTimeSpent</name>
            <upd_type>sum</upd_type>
        </resource_appdata>

        <resource_appdata>
            <name>resultsScreenConsults</name>
            <upd_type>increment</upd_type>
        </resource_appdata>

        <resource_appdata>
            <name>badgesTrophiesScreenConsults</name>
            <upd_type>increment</upd_type>
        </resource_appdata>

    </resources>
</xupil>
```