

Personalised Privacy in Pervasive and Ubiquitous Systems

Elizabeth Papadopoulou

Submitted for the Degree of Doctor of Philosophy

School of Mathematical & Computer Sciences
Heriot-Watt University
Edinburgh, UK

September 2015

Copyright 2015 by Elizabeth Papadopoulou

The copyright in this thesis is owned by the author. Any quotation from the thesis or use of any of the information contained in it must acknowledge this thesis as the source of the quotation or information.

ABSTRACT

Our world is edging closer to the realisation of pervasive systems and their integration in our everyday life. While pervasive systems are capable of offering many benefits for everyone, the amount and quality of personal information that becomes available raise concerns about maintaining user privacy and create a real need to reform existing privacy practices and provide appropriate safeguards for the user of pervasive environments.

This thesis presents the PERSONalised Negotiation, Identity Selection and Management (PersonISM) system; a comprehensive approach to privacy protection in pervasive environments using context aware dynamic personalisation and behaviour learning. The aim of the PersonISM system is twofold: to provide the user with a comprehensive set of privacy protecting tools and to help them make the best use of these tools according to their privacy needs. The PersonISM system allows users to: a) configure the terms and conditions of data disclosure through the process of privacy policy negotiation, which addresses the current “take it or leave it” approach; b) use multiple identities to interact with pervasive services to avoid the accumulation of vast amounts of personal information in a single user profile; and c) selectively disclose information based on the type of information, who requests it, under what context, for what purpose and how the information will be treated. The PersonISM system learns user privacy preferences by monitoring the behaviour of the user and uses them to personalise and/or automate the decision making processes in order to unburden the user from manually controlling these complex mechanisms.

The PersonISM system has been designed, implemented, demonstrated and evaluated during three EU funded projects.

ACKNOWLEDGMENTS

Many people have encouraged me in completing my thesis and I am grateful to all of them for the continuing support over the years.

To Nick Taylor, my academic supervisor, for providing me with invaluable advice, support, motivation and guidance. I am very grateful for the time he devoted to reading and reviewing my thesis so many times, for being very patient with me and for our numerous long discussions. Finally, I am very thankful for giving me the opportunity to work as a research associate and be involved in very exciting EU projects over the last ten years which I have immensely enjoyed.

To Howard Williams, my secondary academic supervisor, for all the support, advice and guidance he has provided me. His long research career has inspired me and I am grateful for his sharing his knowledge and experience with me.

To my husband, Panagiotis Verras, who always believes in me, supports and encourages me in numerous ways. He is the greatest man I have ever met and I cherish every moment I have with him.

To my ex-colleague and friend Sarah Gallacher, for her support and always comforting words. I miss the great times we had working together and the fun we had during our research project trips.

To my friends and family for encouraging me and bearing with me all these years.

To the many European partners I have worked over the years for the great discussions and debates we have had that have inspired me.

Thesis submission form

Contents

1	Introduction.....	1
1.1	The issue of Privacy in Pervasive Systems	1
1.2	The status quo in digital privacy	3
1.3	Aims and Objectives	4
1.4	Thesis Overview	6
2	Related Work	8
2.1	Pervasive and Ubiquitous Systems.....	8
2.2	Personalisation.....	11
2.2.1	User Modelling	13
2.2.2	Recommender Systems	14
2.3	Privacy.....	15
2.3.1	Privacy Concerns	16
2.3.2	Trust, Control and Informed consent	18
2.3.3	Engineering Privacy	20
2.3.4	Privacy Aware Ubiquitous Systems.....	22
2.3.5	End-User Requirements for Privacy Aware Ubiquitous systems	23
2.3.6	Privacy Preferences and Policies	28
2.3.7	Trustworthiness as a parameter for decision making.....	31
2.3.8	Server side policy matching.....	33
2.3.9	Acquiring privacy preferences	34
2.3.10	Privacy Policy Negotiation	36
2.3.11	Identity Management	40
2.4	Summary	46
3	EU Research	49
3.1	EU IST-DAIDALOS	50
3.1.1	DAIDALOS Pervasive Services Platform	52
3.1.2	Lessons Learnt	57

3.2	EU ICT-PERSIST	58
3.2.1	The Personal Smart Space.....	59
3.2.2	Some aspects of the PSS Architecture	61
3.2.3	Lessons Learnt	67
3.3	ICT-SOCIETIES	68
3.3.1	SOCIETIES concepts.....	69
3.3.2	SOCIETIES Architecture.....	70
3.3.3	Lessons Learnt	75
3.4	EU Projects Research in Relation to Thesis Research Objectives	76
3.5	Summary	77
4	Personalisation in a Pervasive System.....	79
4.1	User Preferences	80
4.1.1	Context-Dependent Nested IF-THEN-ELSE Rules.....	80
4.1.2	User Preference model implementations in DAIDALOS, PERSIST and SOCIETIES	83
4.2	Dynamic Personalisation	86
4.2.1	Preference Evaluation	87
4.2.2	Preference Condition Monitoring	88
4.3	Implicit Personalisation	91
4.3.1	Preference Learning & Merging	92
4.3.2	Preference Confidence Level	103
4.3.3	Learning service actions.....	105
4.4	Summary	105
5	Privacy in Pervasive systems.....	107
5.1	Privacy policies	107
5.1.1	The Request Policy	107
5.2	Privacy Policy Negotiation.....	109
5.2.1	The Response Policy	110
5.2.2	Negotiating with the service	111
5.2.3	The Negotiation Agreement.....	113

5.3	Digital Identities	113
5.3.1	Identity Selection	114
5.4	Access Control	115
5.5	Data Obfuscation	116
5.5.1	Data Obfuscation Limitations	116
5.6	Summary	117
6	<i>PersoNISM</i> - PERSONalised Negotiation, Identity Selection and Management...	119
6.1	PersoNISM architecture	120
6.2	Personalisation in Privacy Policy Negotiation	121
6.2.1	Privacy Policy Negotiation (PPN) Preferences	122
6.2.2	Acquiring Privacy Policy Negotiation Preferences	124
6.2.3	Applying Privacy Policy Negotiation Preferences	125
6.3	Personalisation in Identity Selection & Creation	128
6.3.1	Identity Selection Preferences	128
6.3.2	Attribute Selection Preferences	129
6.3.3	Acquiring Identity Selection preferences	130
6.3.4	Acquiring Attribute Selection preferences	131
6.3.5	Applying Identity Selection Preferences	132
6.3.6	Applying Attribute Selection Preferences	133
6.4	Personalisation in Access Control	135
6.4.1	Access Control and Data Obfuscation Preferences	136
6.4.2	Acquiring Access Control and Data Obfuscation Preferences	136
6.4.3	Applying Access Control and Data Obfuscation Preferences	138
6.5	Preference Learning for PervoNISM	139
6.6	Summary	140
7	PersoNISM Evaluation	142
7.1	Online Questionnaire on Privacy	142
7.2	Evaluation of PervoNISM system	156
7.2.1	Pre-Evaluation Step	156

7.2.2	Experiment specification.....	164
7.2.3	Evaluation Results Analysis.....	180
7.3	Summary	182
8	Conclusion	184
8.1	Future work	184
8.1.1	Need for more information.....	184
8.1.2	Including semantics to control identifiability	185
8.1.3	Monitoring service providers' activities	185
8.1.4	Graphical User Interfaces.....	185
8.1.5	Storing user data.....	185
8.1.6	Federated Privacy Policy Negotiation.....	186
8.2	Key Contributions	186
8.2.1	Negotiating Privacy.....	189
8.2.2	Personalising the use of Digital Identities.....	190
8.2.3	Controlling Data Disclosure.....	191
8.2.4	Proactive Privacy Protection	191
	Appendix A – Online Questionnaire.....	192
	Appendix B – Evaluation handout document	200
	Appendix C – PersoNISM code repository.....	208
	Publications	209
	References	215

1 Introduction

“The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it” [1]. The above quote could be the most often cited quote in the literature of Ubiquitous Environments but it marked the beginning of a new era in computing research; that of pervasive computing. For twenty five years, research on pervasive and ubiquitous environments has been ongoing but it is still far from entering our homes and our daily lives. One of the most important problems being faced is the ability of such systems to invade users' privacy. Pervasive and Ubiquitous systems perform effectively when they have access to information about the user such as preferences, interests, goals, intents, environmental (context information) as well as personal information. The one aspect of pervasive computing that concerns users most is the system's need to monitor this information about them at all times. Thus, it is imperative that any such system revolves around the user and is solely controlled by the user. Assuming that a pervasive system platform meets this requirement, it cannot exist in a vacuum. For it to be successful, it has to facilitate the interaction of users with other entities such as other users and pervasive services. This interaction often requires personal data to be disclosed.

1.1 *The issue of Privacy in Pervasive Systems*

Pervasive and ubiquitous systems will be driven by information about the users they are empowering. This thesis is primarily focused on the personal information that is accumulated about a user in such systems and how it will be disclosed, communicated and processed by those systems. There are clearly potential threats to the privacy of users when so much personal information is collected and disseminated and it is the responsibility of these systems to provide appropriate mechanisms to protect the privacy of their users. The research presented in this thesis addresses the issue of controlling disclosures that might infringe the privacy of a user. It is assumed that adequate security mechanisms to prevent unauthorised access will be present in any pervasive system.

The majority of services available in a pervasive and ubiquitous world will adapt to the preferences and contextual situation of their users. The preferences of the users could be accumulated directly from them by manually entering information into the system or by monitoring their behaviour and interactions with the system, identifying common patterns

of behaviour in specific contexts and inferring user preferences and intents based on them. Information about the contextual situation of users will be collected via sensors, networks and the devices which they use. This primary context information may also be processed to infer secondary, higher-order, contextual information. For many services, the more information they can access about the user, the more useful and appealing they will be to the user.

Personal information is valuable and can be exploited for financial benefit [2]. As an example, consider a restaurant finding service that allows the user to enter a location such as a city and then lists the restaurants in that city. Such a service would be more useful if it is able to use the current location of the user to suggest restaurants near the current location of the user. It would also be more useful if the service had information about the type of restaurant the user preferred, the type of food they would like to eat as well as the preferences of the people accompanying them to the restaurant. As the user consumes such services and discloses information such as current location, personal interests, preferences, activities etc., reasoning and inference algorithms could be used to deduce more information about the user. The more information that exists about the user, the better inference can be made about other preferences and habits of the user. For instance, it would be possible that the service monitored the location of the user and the time spent in each location to infer what restaurants the user prefers without the user explicitly providing this information. The accumulation of increasing quantities of information about the user would effectively allow the services to build profiles of their users which can be a dangerous characteristic of such systems and give rise to concern [3]. Further, the accumulation of this information without the user being aware that this information exists about them is even more concerning.

Information such as this can be used very effectively to make profit in targeted advertising [4]. Information is stored in the services' servers and processed accordingly to find similarities, habits and traits of the users in order to offer products and services that are better tailored to the users. Using the restaurant finder service again as an example, consider the scenario of two people using this service meeting regularly at a restaurant. The service could infer that these two users know each other due to the fact that they were at the same location at the same time for roughly the same period of time on more than one occasion. This information is inferred by the service and can be used later for other purposes. What is important in this example scenario is the fact that the user may be

unaware this information is stored in the service's servers and can be used for any number of purposes.

This processing of information without any restrictions, may also infer wrong information about the user or information that they would prefer not to be disclosed. Consider the scenario that a user is visiting a clinic specialising in certain diseases to request information on behalf of a friend. It could be wrongly inferred that the user is a patient in this clinic based only on the fact that the user is located at the clinic for a specified amount of time. This becomes part of the user's record in the system and could be disclosed to other parties at a later time which could be embarrassing or even harmful to the user and/or their reputation depending on the situation.

If appropriate safeguarding mechanisms do not exist in the system, the services will be allowed to access and process any kind of information about the user and use it to their benefit without considering the privacy of the user. Finally, it is also possible that information will be traded between services that have business relationships with each other. Information about transactions that have taken place with one service will be merged and processed with information about transactions with other services, thus resulting in the collection of vast amounts of information about users. In real life, users maintain a separation of their information depending on the context of their interactions. Depending on the role they undertake in each instance, they present themselves differently. For example, different interaction occurs between a user and their neighbour and a user and their colleagues or bosses. So, appropriate mechanisms must exist to allow users to implement the same interaction patterns in the digital world as they do in real life.

1.2 *The status quo in digital privacy*

In their everyday encounters with services on the World Wide Web, users are often presented with the terms and conditions and the privacy statements of the services they are using. However, a very small fraction of users take the time to read them before agreeing to them. One of the problems behind this trend of the users is the manner in which the privacy policy statements are presented on the websites. Some of the users are unable to comprehend the content of the statements because of the legal jargon they contain [5]. Other users don't bother to read them because they are not aware of the

dangers they potentially put themselves in. As for the small percentage of users that take the time to read them, the only options that are available to them are to use or not to use a service. There is no option to change the terms and conditions for using a service to fit the privacy demands of the user. If a user needs to use a service, they will therefore have to agree to the terms and conditions laid out by the service with no input from them [3]. In traditional systems, the information disclosed may not be as personal as in the world of pervasive and ubiquitous systems. In the latter, the amount of information that can be accumulated from sensors and monitoring systems and the quality and sensitivity of that pose much greater risk to privacy than in traditional systems.

Almost every service provides a document with the service's terms and conditions and a document with the privacy policies that describes the privacy practices of the service. These are actually terms of a contract between the user and the service. When the users "click" the "I Agree to the Terms & Conditions" button without reading them, they are clearly signing a contract without reading it. In the non-digital world, this action would almost never occur as people are quite reluctant to sign a contract they don't understand. However, experience shows us that in the digital world, this reluctance disappears as these agreements are not perceived as real contracts. Hence, it is very important to provide mechanisms to protect the users and attract their attention to the privacy policies and practices of the services they use and allow them measures of flexibility on how to use the services with a level of risk to their privacy that is acceptable to them.

1.3 *Aims and Objectives*

In a pervasive system, information about the user is constantly collected, processed, disclosed and shared to adapt the user's environment to meet their needs and preferences. Research into existing solutions shows that traditional access control mechanisms and privacy practice notifications lack required capabilities to address the privacy requirements of pervasive systems [6]. There is a real need to allow users to dictate the terms for disclosing their own data, to provide appropriate tools to selectively disclose information based on the benefit they receive by doing so, and, to help them maintain the level of privacy they desire using personalised suggestions regarding the handling of personal data based on past decisions. This thesis addresses this need by proposing the PersoNISM system, an intelligent privacy protection framework for pervasive

environments that amalgamates privacy enhancing technologies and personalisation practices.

The primary objective is to enable the user to remain in control of their data during and after their data is disclosed. Thus, the first step is to identify appropriate privacy enhancing technologies that can satisfy this requirement. Privacy policy negotiation allows users to change the terms and conditions for data disclosure and handling to fit their needs. It provides a solution to the “take it or leave it” approach currently in use. The use of multiple identities and context-dependent identity selection support the user in organising their information and using it depending on the current context and activity of the user. This approach prevents the accumulation of vast amounts of information linked to a single user profile as a) it segregates the user profile into smaller parts that are grouped together for a specific task and b) identities of a single user are not linked to each other so they appear to belong to different users. Finally, a dynamic access control mechanism enhanced by utilising data obfuscation algorithms allows the user to disclose different information in varying quality depending on the situation and the service they are interacting with.

When users are provided with such tools, they can become overwhelmed by the information they need to keep track of and the configurations they need to perform to protect their privacy. This leads to user fatigue and eventually may discourage users from using the tools properly. Moreover, as more user information accumulates in the system, it is not reasonable to expect users to remember what information they disclosed to whom and under what conditions. Thus, the second objective is to help users maintain the desired level of privacy for their data by personalising or automating the processes involved according to their preferences. By monitoring the user’s decisions regarding the handling and disclosure of their data, the system can learn user privacy preference rules, using a behaviour learning algorithm, that reflect their privacy requirements. These privacy preference rules can be used later to either suggest to the user suitable privacy decisions or to enforce automatically if the user wishes so.

The final objective of this thesis is to assess how well the proposed PersonISM system satisfies these requirements. Initially, an online questionnaire was used to collect information regarding users’ online data disclosure habits aiming at gauging the level of

awareness of privacy of the average user. Subsequently, a user trial was conducted, in which users evaluated the PersoNISM system by providing real personal information.

To summarise, below is a list of objectives of this thesis:

1. To enable users to remain in control of their data
 - a. To design a privacy policy negotiation algorithm
 - b. To develop a system for managing multiple identities
 - c. To design a dynamic context-aware access control mechanism with data obfuscation capabilities
2. To assist users in maintaining their privacy
 - a. To design a mechanism for dynamic context-dependent privacy preferences
 - b. To design an appropriate algorithm for learning privacy preferences
 - c. To develop a system for automating privacy decision making
3. To evaluate the proposed algorithms, mechanisms and systems
 - a. To assess the extent to which users are currently aware of privacy issues and in need of the proposed solutions
 - b. To evaluate the perceived value and usability of the solutions implemented in the PersoNISM system

1.4 *Thesis Overview*

Chapter 2 provides a review of the current state of art in pervasive systems in general and in the areas of privacy, personalisation and personalised privacy specifically in the context of pervasive systems.

Chapter 3 presents the architecture of three pervasive platforms of the EU projects DAIDALOS, PERSIST and SOCIETIES respectively in which the author has worked on the areas of personalisation and privacy protection frameworks.

Chapter 4 gives an overview of personalisation practices in pervasive systems, focusing on the use of user preferences as a tool for personalising pervasive services.

Chapter 5 discusses the use of privacy policies and multiple identities as tools for privacy protection and presents the privacy policy negotiation concept as well as the importance of identity selection and data obfuscation.

Chapter 6 presents the PersoNISM personalised privacy protection system that is based on a set of dynamic context aware user preference rules used to guide the system to perform intelligent privacy policy negotiation, identity selection and data management on behalf of the user.

Chapter 7 presents the evaluation and testing performed on the PersoNISM system. Specifically, the results of a questionnaire on user behaviour regarding privacy, and a user trial of the PersoNISM system are presented which show how well the system managed to satisfy the participant users' privacy requirements.

Chapter 8 concludes with a summary of the PersoNISM system, a discussion on the achievements of the aims and objectives of the PersoNISM system and an outline of future work that can be performed to extend the capabilities of the PersoNISM system.

2 Related Work

The work presented in this thesis relates to three different research areas: pervasive computing, personalisation and privacy protection. There has been a lot of discussion on the issue of privacy with regards to personalisation and context awareness in services of pervasive environments. The problem remains universally unsolved as the development of the technology always precedes legislation controlling its use. The fact that the services are offered on the Internet increases the difficulty of implementing current laws to safeguard the privacy of the users as countries will implement their own flavour of privacy laws. As pervasive systems are not yet mature enough to exist everywhere, it is safe to assume that laws will not be considered until these systems are in effect and their consequences on user privacy become more pronounced. This chapter provides the state of the art in pervasive and ubiquitous systems in which personalisation and context awareness are inherent, and describes the privacy protection mechanisms developed in these systems.

2.1 *Pervasive and Ubiquitous Systems*

“The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.”

Mark Weiser, 1991 [1]

Mark Weiser envisioned a world saturated by computing devices, collaborating, sharing resources and data to aid users in their everyday tasks. Mark Weiser’s vision describes an environment of invisible computing devices anticipating their users’ needs, helping users accomplish tasks without distracting from the tasks themselves. To accomplish this, the system must be context-aware. In Satyanarayanan’s words, “it must be cognizant of its user’s state and surroundings, and must modify its behaviour based on this information” [7]. Hence, a pervasive and ubiquitous system is a computing infrastructure available anywhere anytime, able to monitor the activity and behaviour of its users and the environment they inhabit and configure it according to their current needs and wishes.

Project Aura [8] from Carnegie Mellon University, targeted the integration of existing software and hardware technologies into a pervasive framework [9]. Two basic concepts

that drove the research in Aura are proactivity and self-tuning. Proactivity is a very important concept of a pervasive system and refers to the ability of the system to pre-determine the goals and expectations of a user and try to meet them by performing actions on the user's behalf. Self-tuning is the ability of a pervasive system to react to context changes and proactivity requirements to enable the system to be available always and everywhere with the best possible performance [9]. Proactivity and self-tuning are implemented through Aura's Prism layer: a task layer sitting on top of all the other layers in the architecture including the Application Layer. The Prism layer represents User Intent and communicates this to services and applications required to accomplish an intended task. The basic functionality of the Prism architecture includes: i) representation of user tasks as sets of abstract services implemented by a Task Manager, ii) context management and context event management to reconfigure tasks appropriately implemented by a Context Observer and Environment Manager and iii) event management to react to changes and reconfigure the environment implemented by an Environment Manager. Figure 1 shows the Prism Architecture.

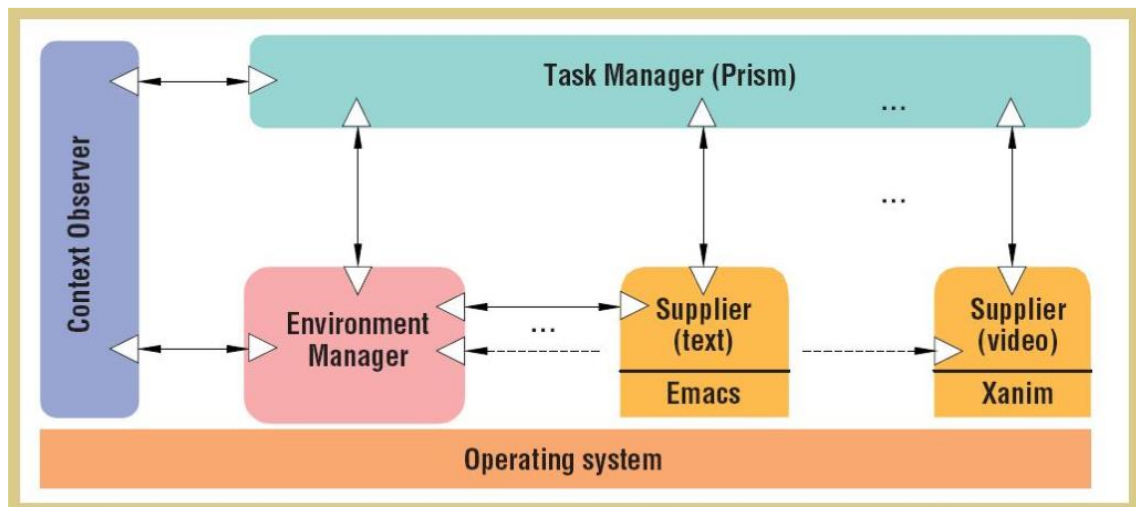


Figure 1. *Project Aura: The Prism Architecture.*

MIT's project Oxygen [10] is a large effort to construct Intelligent Environments (IE) that control a wide range of available devices, sensors, resources and services embedded in these IEs, to provide both proactive and reactive services to its users [11]. Mobility plays a crucial role for all pervasive systems as location changes have to trigger re-configuration of devices which include service re-compositions, session transfers and network handovers. The Intelligent Room is a sub project of Oxygen inside MIT research which attempts to create a multi-agent based context aware home or office environment using

context management and Artificial Intelligence practices which include speech recognition as input to infer user behaviour and implement proactivity. The Intelligent Room facilitates Intelligent Environments (IEs); an IE controls all aspects of physical and digital devices and software that affects that environment (laptops, smart phones, databases, software agents, services, etc.). The research of the Intelligent Room focused on architecting a system that reacts to the user's behaviour in a context aware fashion, automatically allocating shared resources needed for the currently performed tasks and finally enabling collaboration between multiple physical and digital spaces. The Intelligent Room architecture is based on software agents controlling specific resources arranged hierarchically depending on the task that is being performed. Each agent communicates and exchanges control and data messages regarding the resource it manages. A context management component collects data from agents, processes them, and instigates actions in the system according to the collected data. ReBA is a context reactive behavioural system facilitating the representation of context information by constructing a model of the environment based on the current activity being performed. The detected activity indicates what kind of resources are needed and how they should be used to perform the intended tasks.

Another example of an agent-based system is Digital Me (D-Me) [12], a multi-agent system supporting personalised ubiquitous interaction. Two entities are envisioned, the D-Me Context-Aware Agent representing the user, and the environment that provides resources and services with which the user interacts. The aim of the D-Me Agent is to assist the user with as little user intervention as possible, in accomplishing a set of tasks in a smart environment using context information and a to-do list defined by the user or some behaviour learning agent. The D-Me distributed architecture places the user profile data in the handheld device of the user to allow for availability of the data at all times.

The Gaia Operating System [13] [14] is a middleware infrastructure for Active Spaces. An Active Space is defined as a dynamically constructed space defined by a geographic space with physical boundaries, the set of resources and services currently available, its users and active sessions of users interacting with the resources and services. The Gaia OS provides context, event, resource and user session management to facilitate the processes in all current Active Spaces. User sessions are managed by the Gaia OS and associate user data and applications with users. The Gaia OS maintains user session information to enable seamless transition between Active Spaces as users leave one space

and enter another. To facilitate pervasive service provision in areas with no infrastructure, the Mobile Gaia OS was developed as a middleware for ad-hoc pervasive computing. The Mobile Gaia OS allows the creation of “Personal Active Spaces” connected through close proximity networks such as Bluetooth, IrDA or GPRS [15].

Microsoft’s EasyLiving project [16] is a prototype to create middleware software to manage devices and services in a home and work environment. The goal of the EasyLiving software is to move away from the desktop environment and allow users to take advantage of the environment. Using a multitude of sensors and control devices, it tries to assist the user in interacting with intelligent environments by getting interfaces to move with the user. The EasyLiving project focused on creating easy to use world models that represent objects, devices, resources and services in smart environments using geometric models [17]. The geometric models are dynamic, constantly updating as the user roams the environment, providing information about the location of the user and the sessions they are involved in. While previous projects have focused on realising Weiser’s vision for invisible computing and seamless pervasive functionality, the EasyLiving project exposes the user to the pervasive technology.

A lot of effort has been placed on the design and architecture of pervasive systems. While most of the underlying technologies already exist to implement such systems, integrating them all together proves to be a very difficult task.

The term pervasive system will be used throughout this thesis to include ubiquitous computing systems, user adaptive systems and ambient intelligent environments which vary slightly in their emphasis but an issue common to them all is privacy.

2.2 *Personalisation*

Most of the published research papers regarding personalisation refer to the personalisation or customisation of web services or mobile services, specifically addressing personalised service composition [18]. Several definitions of Personalisation exist among which, [19] defines it as: “Personalisation and customisation refers to the ability of an Internet Web site or service to be shaped or reshaped so as to better meet the individual needs or wants of a user”. Jørstad and Dustdar [18] agree with this while adapting this definition to concern mobile services instead of Internet Web sites or services, even though the architecture defined targets both stationary and mobile services.

Initially the focus of the research on personalisation concerned the personalisation of web services for marketing purposes. Much research was carried out looking for efficient ways to offer specific marketed products to users based on their previous transactions, searches and interests, all of which are parts of a profile. As a result, a number of recommender systems were developed that would benefit the user during her visits to e-commerce websites by recommending lists of products comparable with the user's preferences and budget. However, while we are interested in personalisation that improves the user experience, the focus of this type of research does not necessarily address that; spam email, spam SMS and popup browser windows are clearly an annoying side effect of this type of activity [18].

The term personalisation refers to the process of adapting a service, resource or entity in the environment to the needs of an individual user. Personalisation aims to enhance user experience. This may be in contrast to the kind of personalisation practised on the World Wide Web that is often more associated with providing targeted advertisements which is outside the scope of our research.

The term *explicit personalisation* refers to the process of customising a service manually either by the user or the service provider. This requires that the system provides appropriate interfaces for the user to tweak the parameters of the service or manually edit user preference rules to customise the system according to their wishes.

In 2001, Accenture Technology Labs developed two systems (MusicFX and GroupCast) that adapt shared physical spaces according to the interests and preferences of their users [20]. MusicFX controls the music that is played through the speakers of a fitness centre. The music is selected based on the preferred genres of the current users of the fitness centre. GroupCast is installed in an office environment and collects the identities and interests of passers-by to display content of mutual interest on a large visual display located in a shared working space. In both cases, the user data was collected by providing to the users of these systems appropriate forms to manually fill in their music genre preferences and interests respectively. In both cases, a relatively smart space was personalised to the wishes of its occupiers; however, the means for collecting the data needed to perform the personalisation cannot scale in a fully pervasive environment where not one but a large number of services will need user preference information to perform in a personalised pervasive manner.

In contrast, *implicit personalisation* requires that a mechanism continuously adapts the service to the changing environmental conditions based on information on the user's behaviour that has been collected and analysed by the system [18]. Only implicit personalisation can fully satisfy the requirements of a pervasive computing environment. Implicit personalisation implies that there must be a monitoring component that allows user preferences to be collected in the system without the manual intervention of the user.

2.2.1 User Modelling

In order to perform personalisation, systems must maintain information about the user, structured in some form to allow querying and processing. Most commonly, user modelling [21] is the approach used to represent user data for the purposes of personalisation. A user model is used to capture the user's interests, capabilities, beliefs, likes and dislikes, behaviours and other such information that can be used to personalise the user experience. In the context of pervasive computing, a user model must also be able to represent conditional behaviours, that is, behaviours that differ depending on the user's current context. In the EU IST DAIDALOS project, a message redirection system [22] made use of a set of IF-THEN-ELSE rules to control how messages and calls were redirected through the system depending on the user's current activity, location and time.

Another complexity in the context of a pervasive infrastructure is to design a user model that can be leveraged by a variety of applications and services. Only the application developer has the requisite knowledge of the inner workings of their application. Hence, the user model needs to be designed in an abstract manner regardless of the semantics of the applications that may use it. In the EU IST project Mobilife [23], a context dependent user modelling system was developed to provide contextualisation and personalisation functionality for third party services [24]. A user profile contains all the data of the user which consists of sub profiles associated with one or more applications. A sub profile also contains the context in which its behaviours should be performed. The sub profiles are created by a Recommender component that analyses records of user behaviour in the context in which they appeared to construct user models [25].

Several guidelines have been produced by standardization bodies assisting developers in formalising the representation of user data such as the 3GPP Generic User Profile [26], ETSI's User Profile Management [27] and W3C Composite Capability/Preference

Profiles (CC/PP) [28]. The 3GPP GUP specification defines the user profile as the collection of user data that resides in the user's devices, telecom operator's servers or service providers and provides a standardised specification for services to access user data stored by different entities in the network. The W3C CC/PP model is used to describe the capabilities of a device so that Web content can be delivered in the appropriate format for that device. However, the CC/PP model is not suited for pervasive computing environments as it does not allow the specification of a situational (conditional) profile.

One of the risks of automated personalisation is that the system will make a wrong decision due to the inability of the system to model the environment of the user in a manner that will make it reason in the same way that a human would. It is imperative that the system provides appropriate mechanisms that allow users to correct the information that has been collected on their behalf. This requirement affects the modelling of the user's preferences and the corresponding reasoning algorithms that are applied to the monitored data. In some cases, it may be impossible to manually amend the underlying model; for example, in the case of using a Bayesian network to model the user's behaviour and predict the user's intent, it is especially difficult to design a graphical user interface to allow users to make changes to the network. In these cases, mechanisms are required to collect feedback from the user using different means and input this into the network.

2.2.2 Recommender Systems

Recommender systems belong to a type of personalisation that is not related to the personalisation usually deployed in pervasive systems. As mentioned above, recommender systems attempt to personalise e-commerce websites in order to provide a better user experience to their customers. Recommender systems rely on sophisticated algorithms particularly those that use Automated Collaborative Filtering (ACF). ACF algorithms are based on the logic that users who expressed similar opinions in the past will likely express the same opinions in the future. Examples of ACF are memory-based algorithms such as the user-based k-Nearest-Neighbor [29] and item-based k-Nearest-Neighbor [30] where the entire set of preference information is used to generate recommendations, and model-based algorithms such as that of Breese et al. [31] that is based on Bayesian networks where preferences are fed into a model that is used to generate recommendations [32]. One of the drawbacks of using Bayesian models is the inability to view the data inside the model and make adjustments. This problem imposes

great difficulties for systems that need to offer users graphical user interfaces to view, edit or delete their preference data. Recommender systems are a small part of the wider range of personalisation capabilities and therefore, we will not focus on these.

2.3 *Privacy*

Privacy in user adaptive systems is considered the greatest barrier to its long-term success [33]. The quality of a ubiquitous system depends on the information it possesses about the user's behaviour, their activities and their preferences, data that are used to configure services and resources in the environment to fit the user's needs. The issue of privacy for the user who is monitored constantly is at stake. A trade off needs to be made that balances the need for information with the user's right to privacy [34].

Rao et al. [35], describe privacy as "Privacy in the computer age is the right of individuals to protect their ability to selectively reveal information about themselves so as to negotiate social relationships most advantageous to them." Earlier, Irwin Altman developed the Privacy Regulation Theory in which he defined privacy as "a temporal dynamic process of interpersonal boundary" [36][37]. Altman theorises that the more privacy one has is not better off. The optimum level of privacy depends on the social interactions people wish to have at any time and that different people use different behaviours to achieve them. Influenced by Altman's theories, Petronio [38] developed the Communication Privacy Management Theory in which she uses the boundary metaphor to illustrate ownership of information, separating private from public information. The majority of privacy aware systems currently try to mimic the real world and implement the trust values that people develop during their everyday life in the digital world. The problem is, however, that in the real world, transactions are happening between real persons who do not possess the abilities of computers. Human beings tend to apply the same rules of social interaction to information technology as they do to other humans [39], such as the belief that other humans will not disclose information that is very personal to them (non-disclosure), or that humans forget information they do not use over time, or humans' ability to distinguish right from wrong and act accordingly. However, computers do not possess such qualities. On the contrary, acquired information can be stored indefinitely and combined to produce a more extensive profile of online users [40] unless appropriate mechanisms are in place to limit and configure how information is disclosed, processed and disseminated.

While most of the previous work on privacy has been on trying to hide the user's information, ubiquitous systems require that information be released where appropriate so that the user benefits from personalised services in a pervasive environment [41]. Smart objects are dependent on as much information as they can possibly collect in order to serve us. However, as the amount of collected data increases, our privacy diminishes and the threat to privacy increases as more intelligent devices are integrated into our everyday life [42].

2.3.1 Privacy Concerns

Internet consumers have raised privacy concerns and have expressed their distrust and lack of confidence in the online marketplace with regards to the manner in which personal and sensitive information is accessed, retrieved, processed and stored [43] [44].

Wang et al. [45] define the types of privacy threats into the following four categories:

i) *Improper acquisition of information* including improper access, collection and monitoring. Cookies and browsing history are two examples of the means of obtaining user information without the user's consent and acknowledgement. Websites store a lot of information about the user in order to provide a personalised experience to the user. In a pervasive environment, the quality and quantity of information is magnified as it is enhanced by sensory input and information describing the context and behaviour of the user outside the typical scope of a Web Service. Pervasive services and applications provided by third parties can gain access to such information if proper mechanisms are not in place to protect it.

ii) *Improper use of information* such as inference and sharing. Information can be disclosed to third parties for specific purposes. However, any processing of this information to infer further information such as determining a user's shopping patterns without the user's explicit consent is unlawful, as is the sharing or selling of this information to others.

iii) *Privacy invasion*. Spamming in the form of e-mails, RSS feeds, and mass mailing lists is an annoying effect of improperly acquiring user information according to Wang et al. Furthermore, identity theft is another dangerous by-product of improper acquisition and selling of user information.

iv) *Improper storage and control of personal information.* If systems or services do not properly store information disclosed to them users have no means to protect their information when stored in mediums outside their control. Moreover, if these systems do not allow the user to have control of the disclosed information, users are unable to retract or alter the information that has already been disclosed. Security and privacy components should provide adequate mechanisms allowing users to enforce conditions regarding the management and storage of their information, demand access to the information after disclosure and the right to limit the period of time that information is retained in third party servers.

Although these privacy threats were identified in the context of a user's interaction with Web Services, the same privacy threats are also present in pervasive systems. These privacy threats pose greater risks in the context of pervasive systems because the available user information in such systems is more up-to-date and more critical in identifying a user. An example of such information is location information. Even if location information is the only piece of data disclosed, it can be used to identify the user (e.g. in the home environment).

In an effort to identify privacy threats and vulnerabilities against the users of Ambient Intelligence environments and enable developers and policy makers to provide appropriate privacy policies, the SWAMI (Safeguards in a World of AMbient Intelligence) Project produced several "dark scenarios" demonstrating situations where the users of pervasive systems become victims of the technology surrounding them. While the actual scenarios are not presented here (they can be found in [46]), the results of this type of research are briefly discussed below.

The SWAMI research project focused on the following key issues: privacy, identity, security, trust, digital divide, loss of control, dependency, exclusion and victimisation. The threats identified related to privacy (and not security) are surveillance and identity theft. Ubiquitous systems are vulnerable to the "little" brother phenomenon. The "little brother" refers to the commercial companies and service providers undertaking the role of the state in the "Big Brother" phenomenon and being able to conduct unauthorised surveillance on its citizens. A ubiquitous system is vulnerable to identity theft by malicious persons who will take advantage of the information made available to harm the users of ubiquitous systems. Data laundering is a new crime related to identity theft that

is discussed in the dark scenarios. Data laundering is the process of making illegally obtained personal information seem like they were obtained legally.

The SWAMI “dark scenarios” were created not to discourage users from using pervasive systems but to identify the potential threats and eliminate them so as to protect the user and guarantee their privacy as well as to point out the benefits of such a system. In the process though, these scenarios also served to remind researchers of the fine balance between the need for such a system and the dangers to which it can expose the user.

2.3.2 Trust, Control and Informed consent

Trust, control and informed consent are the three pillars for constructing privacy aware pervasive systems. Van der Geest states that “*Creating trust, giving users control, and requesting informed consent are essential conditions for solving the privacy issue*” [47].

Trust

Trust, as in everyday life, plays a crucial role and is the one criterion by which we allow other people or systems access to our personal information. General trust is noted by Geest et al. [47] as the basic type of trust that humans have towards other people and it is the existence of this basic type that allows other types of trust to develop. More importantly, Geest et al. suggest that without a general sense of trust, users would be unwilling to engage in any interaction of any kind. Research has demonstrated that lack of trust in online systems was the major reason for users not to engage in online shopping [48].

Grandison et al. define trust as “the firm belief in the competence of an entity to act dependably, securely and reliably within a specified context” [49] and has been adopted by the Wireless World Initiative (WWI). WWI emphasises the relation of trust within a specified context to draw attention to the fact that although entities may trust each other this may not be true in all contexts because different contexts pose different restrictions and are vulnerable to different types of threats.

Trust in the context of privacy in pervasive systems involves trust in the services that will access the personal information and how this information will be used. Viewing e-commerce websites from the spectrum of deployed services, it is easy to see that trust in

pervasive systems is not that different from trust in the World Wide Web. Trust can play a very important role in deciding to disclose information.

Control

Control refers to two distinct and important facets of any privacy aware system: the first is control over the information disclosure and the second is control over the usage and maintenance of the information, including the location where the data is stored and the provision of appropriate interfaces to remove disclosed information from services. Good pervasive system architectures cater for the provision of appropriate mechanisms to control the disclosure of information to services by providing user-friendly graphical user interfaces that allow users to perform access control, edit privacy preferences, read privacy policies and designate trusted third party services allowed to consume personal data. The more control a user has over what, how and when their information is disclosed, the more trust the user has in the system and therefore, the more willing they are to immerse themselves in a pervasive environment.

Moreover, in order to design efficient, privacy aware systems, the requirement to provide effective mechanisms, such as Graphical User Interfaces (GUIs), to enable the user to view, edit and delete any personal information held on them, must be catered for. In their survey study, Kobsa et al. [50] found that providing the means to edit user preferences and privacy policies does not absolutely satisfy the requirement for the provision of control. Specifically, emphasis must be placed on the appearance of the graphical user interfaces and the presentation of data in order to impart a sense of control to the user. Another matter that adds more complexity to this problem is that the user interfaces have to be designed to fit in the small size of the device screens. Presenting complex privacy policies on relatively small screens can be very challenging [51]. Manber et al. [52] and Cranor [53] also state that even if good personalisation practices are followed and user interfaces are provided and privacy statements are explicitly stated, users are unwilling to waste time and effort to read them or edit their preferences or they are unaware that any such interfaces or policies exist. This latter discovery only highlights the need for a) the automatic learning of privacy preferences, b) the personalisation of the privacy enhancing technologies by software working on behalf of the user and c) the provision of explicit mechanisms that draw the users' attention to such tools while the user interacts with the system.

Informed Consent

Under the EU Data Protection Directive of 1995 (95/46) [54] and the subsequent directive 2002/58/EC [55] on privacy and electronic communications, it is a legal requirement on a service to supply a mechanism to get the user's informed consent for the acquisition of personal information. Earlier on, the right to be given notice of data collection and usage practices can be found in the 1973 U.S. Department of Housing, Education and Welfare Fair Information Practices (FIPs) [56]. Consent refers to the user agreeing to the disclosure of their information. Informed consent refers to the user agreeing to the information being disclosed after being fully informed of any usage policies the service will practise over her disclosed personal information. Apart from the legal ramifications of such a requirement, acquiring the informed consent of the user will also grow the user's trust in the service and result in the user engaging with the service.

Traditionally, informed consent is obtained by having the users tick a checkbox to acknowledge they have read and understood the terms and conditions when installing software or signing up for a service. However, studies have shown that the terms and conditions are difficult to read, they are considered a waste of time and are hardly ever read by the average user [57], [58]. McDonald and Cranor also note that users implicitly (or subconsciously) perform a type of cost benefit analysis on reading privacy policies in terms of the time it takes to read a privacy policy and the benefit of knowledge of the content and the trust that it will be respected [5].

2.3.3 Engineering Privacy

Spiekermann and Cranor [59] mention three spheres of privacy control based on access to data; the *user sphere* includes data on devices that are entirely under the user's control, the *recipient sphere* includes service servers entirely under the control of a service provider where the user has no access and the *joint sphere* where users store data on recipients' servers where both users and data recipients have access to data (such as email, file hosting services, social networks). Users can control the information flow from the user sphere to the other spheres with appropriate and intelligent access control mechanisms. Information processing and sharing in the recipient and joint spheres must be performed according to privacy policies that the user has agreed to. Contextual Integrity [60] is a conceptual framework stemming from the social norms for expectations of privacy, namely the expectation of users that service providers will only use and share

user information for purposes that are relevant to the service. Barth et al. [61] attempt to formalise the concept of contextual integrity regarding the transmission of information from one entity to another (i.e. from one user to another, or a user to a service, or a service to another service) using first-order temporal logic. A privacy norm is defined by the current context, the role of the recipient of the information, the owner of the data and the type of data to be disclosed. A privacy norm can be positive or negative where “a positive norm permits communication *if* its temporal condition is satisfied, whereas a negative norm permits communication *only if* its temporal condition is satisfied”. Caprice [62] is a tool aimed at aiding designers of privacy critical applications to discover privacy threats and adapt their software to mitigate them in a three-step process: a) identify what context information must be monitored, b) identify the privacy threat before data is disclosed and c) provide courses of action depending on the severity of the threat and the benefit of disclosing the information [63]. The privacy threat analysis uses Barth’s contextual integrity framework to examine whether the privacy threat is severe enough to warrant appropriate action to preserve the user’s privacy.

Contextual integrity is the founding principle of the “privacy-by-policy” architecture model (also termed “notice-choice” model) in which personal information is collected by services and protected according to a set of laws and guidelines. The privacy-by-policy architecture is the predominant architecture currently in the industry. In contrast, some systems have attempted to implement a “privacy-by-architecture” model which tries to protect user privacy by not collecting personally identifiable information. One early example of privacy-by-architecture design is the Place Lab geo-positioning system developed at Intel Research in Seattle [64] which uses radio beacons to announce their location. Based on the beacons’ signal strength, devices can calculate how close they are to these locations and determine their own location. The Global Positioning System (GPS) works in a similar way with satellite signals being received in a unidirectional manner from satellite to device, hence actual location information remains in the user sphere. The SQUARE methodology developed at Carnegie Mellon [65] was developed to help designers and developers ensure that security requirements are collected and analysed properly during the development phase of any project. The goal of privacy-by-architecture design is to design systems where privacy and security requirements are satisfied in the design of the system rather than during runtime. The 9 steps of the

SQUARE methodology have been adapted for privacy requirements engineering and published in [66].

2.3.4 Privacy Aware Ubiquitous Systems

The Secure Persona Exchange (SPE) framework, described in [67] and [41], divides up the user information to create digital identities termed Personas. The SPE framework uses an Authorizing Entity acting as a third party entity that issues personas to be used with certain services based on user preferences, the type of information the service is requesting, the privacy policies of the requesting service and the reputation of the service to keep in line with its policies. The SPE framework is based on the Platform for Privacy Preference (P3P, see 2.3.6) notice-choice privacy model. Services request information and provide their privacy policies which are expressed in P3P vocabulary. P3P vocabulary requires that the privacy policy contain at least the following information: a) the purpose of data collection, b) identification of the entity collecting the data, c) identification of other entities that will have access to this data through this entity and d) the data retention policy. The P3P vocabulary includes other types of information that can be included in the request but they are not mandatory. The service's privacy policy provided when a service is requesting user information covers the notice part of the notice-choice model. The choice part that follows is the action taken by the user, or the user agent acting on behalf of the user. The privacy policy is examined and compared to the users' privacy preferences. At this point, the user may opt to use preconfigured privacy preferences, or configure them to block or release some or all of the information data requested.

In [68], Lederer et al. propose a conceptual model of everyday privacy for ubiquitous computing systems based on Adam's user perceptual model [69]. The model uses an abstraction layer in representing sets of privacy preferences using faces. Users assume a specific face depending on a situation i.e. "cocktail party", "secure shopper", "anonymous", etc. This abstraction makes it easier for users to associate their privacy preferences with a specific situation. Lederer's "faces" are comparable to the "secure personas" used in the SPE framework.

In the Privacy Awareness System (PAwS) [70], services residing in a ubiquitous environment use short range wireless *privacy beacons* to announce their data collection practices to the *mobile assistants* of users through privacy policies. Services can provide

a list of privacy policies, each one offering a different quality of service and a corresponding data collection practice. Users can express their privacy requirements using privacy preferences which are compared against the privacy policies to select the quality of service the user wants. Users can decline the use of a service or disable a monitoring device if the data collection practices of the service are not acceptable to them.

Context Fabric (ConFab) is a framework designed with a suite of privacy mechanisms to aid developers in creating ubiquitous computing applications which allow users to leverage the three pillars of privacy: trust, control and informed consent [33]. Ackerman claims that ConFab is the most advanced infrastructure for the protection of privacy at the time of publication but agrees that the supported preference model is very simple [71] [72]. ConFab offers 3 basic interaction patterns: optimistic, pessimistic and mixed-initiative which are used in specific contexts. The optimistic pattern is intended for applications that disclose user information and detect abuses. In this pattern, the user sets up preferences and access policies after a notification alerts the user of information disclosed and should be applied to information that is not sensitive. The optimistic pattern may seem to be not very privacy aware but it is useful for situations where preferences do not exist such as when the user uses a new service. The pessimistic pattern is used with applications where it is critical to prevent disclosure of personal user data. Access policies and privacy preferences are set beforehand and any disclosure of personal information must strictly adhere to the user's expressed wishes. Finally the mixed-initiative pattern is used where user intervention is needed to decide upon disclosure of information. When such a situation arises, the system explicitly asks the user what action to take. ConFab adopts an approach similar to the one taken by many Digital Rights Management Systems such as PAwS [70] to tag personal information for purposes of auditing and tracking [33]. Tagging personal information as it is circulated among third parties can be monitored and returned as feedback to the system which can be used as input in deciding future practices with regards to information disclosure.

2.3.5 End-User Requirements for Privacy Aware Ubiquitous systems

A set of end-user requirements for designing privacy aware ubiquitous systems stemming from a set of privacy laws and governmental reports in Europe [54], [55], [73], [74] and the United States [75], [76] are outlined below. Further important research on this topic has also been published in [41], [70], [33] and [77] outlining how the architecture of the

Secure Persona Exchange (SPE) system, the Privacy Awareness System (PAwS), Context Fabric (ConFab) and PlaceLab [64] respectively meet these requirements.

Value statement

Users must be explicitly shown the benefits they would get by providing contextual information in exchange for a personalised service. Brar and Kay [41] use the term “Purpose Specification”. Similarly, Langheinrich [70] identifies the “notice” principle noting the need for a mechanism to declare the collection practices of the service and “access and recourse” principle noting the need to inform the user regarding the usage of their data. Hong et al. [33] make specific mention of handling special situations such as emergencies where exceptions can be made. While it is imperative that users should be explicitly made aware of the benefits of providing personal information, there are exceptions such as emergency situations where it should be required to disclose certain types of information to the authorities if they can present that this is necessary and done lawfully. Kobsa and Teltzrow conducted a survey using a bookstore website that provided book recommendations [78]. The quality of the recommendations depended on the amount of data the users disclosed. Users were told that the more data they disclose the better the recommendations they would get from the website. The survey showed that users would disclose more personal information if they were given clear explanations of the websites’ privacy practices and what they would gain by disclosing each item of data.

Openness

All data collected should be available to view by the user. Here, Brar and Kay [41] point to the notion of invisibility of the ubiquitous computing environment and state that while this is one contradiction, the benefit of the user being able to monitor what data is collected is greater than the invisibility concept. Langheinrich [70] uses the term “access and recourse” again to refer to the need to allow the user to edit their preferences and any sensitive information held about them. He also emphasizes the requirement of providing simple and efficient graphical user interfaces for the user to easily manipulate her personal information. Hong et al. extend this requirement to include simple and appropriate feedback to the user regarding services accessing personal information. It is important to the user to know when, to whom and which information was revealed at any point in time [33]. In the context of privacy on the Web, the “WhatTheyKnow” [79] tool is a JavaScript tool embedded in a page that shows the user what third party advertisers can see through

the user's browser history. The purpose of the tool is to educate users about behavioural advertisement, raise privacy awareness and show, based on the information available, what other information can be inferred about them. Fu et al. [80] developed an application that monitored which Android applications were accessing user's location information and how frequently they did so. Twelve out of thirteen users participating in a field study using this application reported that they did not understand why certain applications (such as "WhatsApp" [81], "ESPN" [82] and "CricInfo" [83]) were frequently requesting their location information even though they did not offer location based services.

Simple and Appropriate Controls

Brar and Kay [41] emphasize that the user should have the ability to control the release of information using simple mechanisms. They propose to implement this by creating separate personas, private and public ones, where services are given access to the information the user allows them access to. Langheinrich [70] notes two principles of "notice" and "choice and consent". Notice is given through policy announcement mechanisms by the services (such as privacy beacons) and informed consent is taken using machine-readable privacy policies that are checked against the policies announced by the services. This process is common in many privacy aware systems. Providing simple and appropriate controls to enable users to leverage such practices not only protects their privacy and builds their trust regarding the system but also reveals the advantages of privacy enhanced pervasive systems [77], [33].

Limited Data Retention

For data stored by service providers, there is little that a user agent, working on behalf of a user, can do to limit data retention. However, it is possible to negotiate the release of information based on data retention policies advertised by the services. While Brar and Kay [41] explicitly state the data retention policy as a requirement, Langheinrich [70] satisfies this requirement by treating it as a standard part of the privacy policy announced by the service and does not state it as an explicit requirement or principle. Hong et al., also explicitly state this requirement and highlight potential intrusive data mining that might lead to incorrect assumptions about someone's activities.

Decentralised Control

Brar and Kay [41] mention the concern of users about systems that gather sensitive information in one centralised store that is out of their control. The SPE framework applies a distributed architecture where data is stored in the mobile device of the user and therefore, any sensitive data is under the user's control. However, this solution has a setback and raises the problem of availability. If the mobile device is not available to retrieve data from, then services will not be able to offer personalised services. Hong et al. [33] emphasise the drawbacks of centralised storage systems. In their survey, they discovered that users felt they had no overall control of their centrally stored information and they feared that if higher level employees wanted to examine their personal information, there was little they could do to stop them. Koliass et al. [84] present a high level architecture to store user profiles on client devices rather than service servers. Users specify which information is sensitive and write their own privacy preferences which are stored on client devices and then shared across different web applications according to the privacy settings. The proposed architecture allows that monitoring and processing of user data can be performed on the service server but none of the data is stored there. While the architecture suggests this, it does not provide a mechanism to prevent the service from keeping the data stored on the server.

Plausible Deniability

This requirement is only mentioned in [33] and it highlights the need for a user to avoid embarrassing situations by hiding or “obfuscating” some information. In the real world, not answering a phone call may imply that a user is busy, is out of reach, or not wanting to talk to the caller but the actual reason should not be revealed to the caller by the system. Therefore, the system should allow this functionality without the user having to take special action for it to be in effect.

Anonymity and Pseudonymity

Personalised services can be offered by providing information that does not identify the user. In the Secure Persona Exchange system, some personas created can contain pieces of user information that by themselves cannot identify the user. These personas can then be used to interact with services where user identification is not necessary. Langheinrich [70] suggests that a balance needs to be made between the virtue of anonymous virtual identities (pseudonyms) in the digital world and the need to disclose personal information.

The implementation, proposed in [70], suggests that pseudonymous interaction is possible by not disclosing any identifying information about the user but it is not explicitly stated.

The term “anonymity” refers to the concept of a user consuming a resource without having to reveal any identifying information or be authenticated in any way to use it. A good example of an anonymization tool is the Tor project [85] which provides anonymity to online users (while surfing the web, participating in forums such as for abuse victims, communicating such as for journalists with whistleblowers etc). Tor protects the user’s online privacy from Internet surveillance through traffic analysis which can reveal the location of a user at a specific time.

Pseudonymity, on the other hand, refers to the concept of a user holding a set of pseudonyms that allow them to consume a resource or use a service without identifying themselves while, at the same time, enabling the user to be held accountable for it [86]. Rao et al. [35] refer to pseudonymity and anonymity as the core building blocks of Privacy solutions in the Web. Even though Rao et al. refer to privacy in the World Wide Web, pervasive systems share a lot of commonalities with it and it is essential that anonymity and pseudonymity are available tools for the pervasive user. Thinking about how millions of users surf the web everyday proves to be a good example in order to demonstrate anonymity or pseudonymity. Websites allow the user to browse their pages without having to identify themselves to them. This is one situation that should be available to accommodate the user in pervasive systems. Zarsky suggests researchers should consider anonymity based solutions where one-time identities can serve as contact points in transactions [87]. These one-time identities will not be valid after the transaction is complete and therefore, tracing multiple transactions back to an individual user will be made almost impossible. Moreover, Zarsky points out that anonymity based solutions will not prevent marketers from collecting personal information but it will limit the information to the specific data that the users gave their consent to be disclosed.

On the other hand, anonymity breeds distrust. Zarsky points out that anonymity does not come without problems. If every user could stay anonymous during interactions with services, then users could provide false information and try to avoid being held accountable for their interactions [87]. Consequently, there is another balance needed here between allowing the user to remain anonymous but at the same time, including mechanisms that will allow accountability. The biggest disadvantage of anonymity is that

it renders personalisation obsolete. No user information will ever be held to be link to an actual individual and therefore, preferences will not be carried over from one session to another. As a result, another balance to be struck, is providing the means for anonymity but also providing the means to personalise services.

Kobsa and Schreck state that anonymisation hides the relationship or linkage between an individual user and her stored personal data [88]. They suggest that a well-designed system should allow for many levels of anonymity.

The CRUMPET project [89] targets location privacy using a Mediator Agent controlled by the user, installed either on a user's device or a trusted third party's device. The Mediator Agent hides the identity of the user from the service provider. Instead, it handles all communication using pseudonyms allowing user profiles to be constructed but not with real identities.

Beresford and Stajano [90] present a privacy protecting framework for the Active Bat system based on frequently changing pseudonyms to avoid user identification from location tracking. The use of frequently changing pseudonyms prevents applications from tracking users' activities so they are unable to infer further information or even identify them. For example, location tracking could expose the place of work of the user and then possibly identify them. Pseudonyms could, however, be linked if studied properly, hence to further enhance location privacy, the concept of mix zones is introduced. Mix zones group together users in the same geographical area. A mix zone is identified by the group of spatially connected users and is created between application areas. Location information is not disclosed within mix zones. A user changes into a new pseudonym before entering a mix zone. The more users in the mix zone the better the anonymity of the user.

2.3.6 Privacy Preferences and Policies

Privacy preferences refer to user preferences that specify the user's wishes regarding the disclosure of their personal information. Lederer et al. [91] report an interesting finding while surveying users to determine the accuracy of privacy preferences. The focus of the questionnaire was the "inquirer", i.e. the data collector, and the "situation", i.e. the context in which the inquiry is made. The result of the questionnaire showed that users were most likely to adopt the same disclosure practice for the same inquirer rather than different

disclosure practices based on the context in which the inquiry was made [91]. Lederer et al. [91] report that their findings are partially similar to Adams' in [92] which reports that the user-perceived privacy in an audio-video captured environment is a combination of the receiver's identity (same as Lederer's inquirer [91]), the usage of information data that is received as perceived by the user, the context in which the information is obtained (same as Lederer's situation [91]) and the sensitivity of the information to be disclosed.

Hong et al. [77] [33] note that different people have different levels of trust for service providers and other users. Moreover, based on their survey, it is evident that while users were willing to share information, such as their location, with friends and family members, they were concerned about offering this information to services which did not offer a tangible value in return and did not offer adequate privacy protection. Finally, they were adamant about controlling when and how this information would be disclosed and in which contexts, demonstrating for example, that they would allow their location information to be queried by their boss or their colleagues during working hours but not at any other time. It is therefore, a requirement for a privacy protection system to allow preferences to be dependent on the context of the user and allow for flexibility in different situations.

A number of languages have been developed to model preferences for privacy. There are at least two privacy policy description and enforcement models that have been accepted as standards. These are the Platform for Privacy Preferences (P3P) [93] and the eXtensible Access Control Markup Language (XACML) [94]. APPEL (A P3P Preference Exchange Language [95]) is a standard for specifying user privacy policies on the Web. APPEL is part of the Platform for Privacy Preferences [93]. APPEL allows users to enter their privacy preferences using a graphical user interface. These are then translated into a machine readable format which the P3P agents use to communicate with. P3P is a standardised set of multiple-choice questions covering all of the major aspects of a website's privacy policies. The OASIS XACML open standard is another language for specifying privacy policies. XACML was designed for use by Web Services but its specification is flexible enough to allow it to be used to express user privacy preferences. Another language for privacy policy expression is the Enterprise Privacy Authorisation Language (EPAL) [97], a proprietary language provided by IBM which allows developers to specify enterprise privacy policies; EPAL can coexist with the P3P APPEL because EPAL specifically targets specification of enterprise-internal privacy policies compared

to APPEL which can express the privacy statements of any entity. Because of this, APPEL allows a broader range of applications and offers a standard vocabulary to fit any enterprise.

In P3P APPEL, the preference evaluation algorithm produces the first rule that matches the situation “purpose, recipient, retention” regardless of whether subsequent rules match as well. Eldin and Wagenaar [100] propose a weighting system for APPEL preferences to solve the issue of preference order. The weights are assigned to individual preferences. A value that defines the level of privacy threat is determined at runtime and is compared against the weights of the privacy preferences to find the best match. Hence, APPEL privacy preferences are formulated using the set “purpose, recipient, retention” and the weight.

Other initiatives include the Common Policy [101] framework, a proposed standard defining a simple framework for creating authorisation policies to access application-specific data. The Common Policy framework has been specified in RFC 4745 since February 2007 but has not been adopted by commercial systems. The initial purpose of this framework was to combine the authorisation mechanisms developed separately to control location and presence information. The proposed framework, as specified in the RFC 4745, can handle authorisation of data in any application domain by providing mechanisms for extensibility.

Research shows that the amount of information a service requests from the user, the language in which a service’s privacy policies are described and the consequences of releasing certain data are too great a burden for the user to control themselves. Users should not be expected to maintain the knowledge of what information has been disclosed to whom and under what circumstances. Even if users managed that, it would be impossible for them to process it appropriately and act optimally [102]. The system should monitor the user’s actions, their context and the service’s practices and its compliance with its practices and provide appropriate guidance and advice on exchanging information with services. However, the user must remain in control of any transaction, and when the system is not confident about any automatic release of information, it should prompt the user to make sure that any system action has the user’s consent. HCI plays a vital role in the way certain questions are presented to the user and helping the user fully

comprehend the extent of the implications if privacy is breached or not adequately protected.

“The long-range vision is of systems that let people, agents, services and devices seamlessly interact as autonomously as possible while preserving appropriate security and privacy policies” [103]. The challenges presented by Kagal et al. focus on open and dynamic networks whose participants are not predetermined and identified and can change regularly. Kagal et al. stress the importance of interoperability when agents from different organisations and people have to communicate and agree on specific policies. Kagal et al. suggest that Semantic Web technologies such as the Resource Description Framework [104] and Web Ontology Language (OWL) [105] can be used to guarantee that parties communicate and negotiate terms based on the same understanding of the meaning of these terms. They also suggest that a declarative policy can tackle the problem of not having predetermined participants in these open and dynamic environments. Specifically they give the example of Rei, a declarative policy language that uses semantic technologies to describe policies and constraints over allowable and obligated actions on resources in a distributed environment. Kagal et al. claim that Rei can support sanctions and conditional permissions (such as the result of a policy negotiation process). Sanctions are a model of the consequences an entity will suffer if the policy agreement is violated. By conditional permissions, Rei allows a service to perform an action or get access to a resource as long as a set of conditions are met, such as retaining a record of the accessed data for a specified period of time (data retention).

Garcia and Toledo [106] propose a privacy framework combining three standards P3P, WS-Policy and OWL. The P3P vocabulary is defined as a generic privacy ontology but domain specific services can extend this to cover specific intricacies of their domain, such as healthcare, e-government, etc. Policies are defined using the WS-Policy language and point to specific ontologies. In their implementation, the service discovery process takes into account the user’s privacy policies to filter out services that do not comply with these policies.

2.3.7 Trustworthiness as a parameter for decision making

“Trust expresses the level of access to resources that can be granted based on the available information and evidence” [107].

Researchers in the area of trust management tend to concur that context plays a very important role in trust management. A trusts B to do action x but not action y. Hence, the action is the context condition that affects the trustworthiness. Trustworthiness cannot be generic so that if A trusts B then A will trust B to do anything [107].

Kagal et al. [103], suggest that one solution to the problem of unknown participants is trust-based security. In such systems, self-evident properties that can easily be proven true (such as the originating IP address corresponding to a domain “.gov”), proof of key attributes and signed statements from trusted sources are used to determine whether an entity should be given access or not. Kagal et al. refer to Privacy Aware Web Project (PAW) as a good example of a framework that implements trust-based security. It is also mentioned that PAW is more than just a trust based access control system. It is developed as a general purpose policy framework for the Web, that lets users define trust based policies in their own policy language [108].

A user study on user interactions within a simulated intelligent environment found that the disclosure of personally identifiable information depends on how much the users trust the owner of the intelligent environment [109]. Another survey found that the trust and reputation of a data collector are a determining factor (among others) for disclosing more personal data [110]. Giang et al. [111] propose an access control mechanism where disclosure of data is decided based on the trust assessment of the data requestor. The trust estimation is used as input to a predefined access control policy that defines an “allow/deny” decision.

The Mobilife project along with 3 other EU IST FP6 projects - Ambient Networks, WINNER (Wireless World Initiative New Radio) and E²R (End-to-End Reconfigurability) - joined to form the Wireless World Initiative (WWI) which went on to develop a Security and Trust Framework [112]. The security aspects of this framework will not be discussed here, but some aspects of the trust model, on which the framework was based, are relevant. Transitivity and Brokering of trust is one of the main concepts being researched by the WWI and relates to the trust brokered by 3rd parties. “If A trusts B and B trusts C, then does A trust C?” [112]. Trusted 3rd parties are needed to vouch for services that the user does not know about and thus does not trust. The transitivity of trust in such cases would mean that if a service’s policies are guaranteed by a 3rd party broker,

then the user trusts this service as much as she trusts the trust broker. Finally, the WWI highlight the challenges identified in building a trust model which include:

- A large number of brokers will be needed to service agreements between users and services or between services and other services and enable on-the-fly agreements to be created.
- A service delivered to a user may consist of a number of services composed together. For example, a VoIP service needs a network service as well which leads to complex dependencies which have to be catered for in the trust model.
- Trusting a service does not automatically imply that a service is always trusted. The nature of ubiquitous systems dictates the dynamicity of service compositions with networks and applications and therefore, a trust relationship may change over time and with different contexts.

Another trust management model used in pervasive environments is UbiSec's Pervasive Trust Management model (PTM) [113] which features a very different architecture. Trust management is distributed, supporting transactions where a central server is not always available to broker trust agreements. The model is based on the notion that services are offered by users instead of users requesting services. Pervasive devices establish their own trust agreements with other users and act as their own Certification Authority. Fuzzy logic is used to represent trust relationships where a continuous function tending to 1 would denote complete trust while one tending to 0 would imply complete distrust. The PTM also allows a risk management module to be built into the probabilistic model of trust management. Such a module would calculate a risk factor of interacting with a specific service and this could be used as input to the trust negotiation phase.

2.3.8 Server side policy matching

Carminati et al. [114] present a solution to managing the user's privacy introducing a form of privacy preferences based on third party architectures. The basic idea is that a user "outsources" the management of his personal data to 3rd party publishers. These publishers have to ensure that the user's privacy preferences are applied in every request for personal data from services. The publishers also need to ensure that any personal data they receive from the user is not accessible for viewing by the publisher and that is achieved by encrypting the data before outsourcing them.

It should be noted that depending on the type of preferences, this would be possible. However, in the context of pervasive computing, the evaluation of the preferences implies access to the context of the user which includes location among other important personal information. Therefore, it is impossible to a) ensure that the publisher only applies policies and does not have access to actual personal data and b) to persuade the end-users to “outsource” all their personal data to a third party entity. Carminati et al., imply that an advantage of outsourcing personal data is that this relieves the user from the burden of the management of their personal data while maintaining a high degree of control over them with the use of privacy preferences. This approach is quite static since the proposed Trusted Privacy Manager component introduced in their architecture matches requestor privacy practice policies (expressed in P3P) to predefined user privacy preferences (rulesets expressed in XML notation, a simpler version of APPEL) [114].

Nyre et al. [115] argue that users cannot handle privacy on their own, therefore they propose a solution in which privacy is offered by the service provider. Agents acting on behalf of the user send privacy preferences to the service provider who tries to satisfy them. The approach aims to relieve the user of managing a complex set of privacy preferences. A privacy preference provider offers a predefined set of privacy preference stereotypes (other uses of stereotype preferences have been published in [116]) that users choose from based on their data disclosure disposition during service session initiation. This approach has several problems such as a) the exclusive use of stereotype preferences cannot reflect the user’s exact privacy requirements, b) access control could not be dependent on the user’s context and trustworthiness of the service, c) the system cannot improve based on users’ disclosing behaviours as it makes exclusive use of stereotype preferences not associated with a particular user and d) while the approach tries to relieve the user from the burden of managing privacy preferences, it still requires that the user chooses from a complex set of privacy preferences to send to the service provider during every session initiation.

2.3.9 Acquiring privacy preferences

Any system should provide appropriate interfaces for the user to create privacy preferences. PrivacyBird.org [117] originally developed by AT&T Corp, was a complete P3P tool that allowed users to enter their privacy requirements. The tool allowed for custom setup of privacy preferences where a user could create privacy preferences in

detail or use one of three predefined settings “low”, “medium”, “high”. The tool was only available on the Web for P3P enabled Websites and due to the static configuration it provides, it is unsuitable for use in pervasive environments and has not been maintained for several years. A more recent (but not commercially available) P3P privacy preference generator is described in [118] which aimed at providing users with a user-friendly graphical user interface to enter privacy preferences. Like PrivacyBird.org, the tool provides beginner, advanced and expert modes for users with varying system knowledge. The tool defines twelve service types that correspond to user goals which were determined by identifying the primary purpose of collecting data by websites and several surveys e.g. [119] that offered examples of categorisation of internet sites: webmail, news portals, online shopping, banking, social media, forums, instant messaging, games, health portals, e-government services and e-learning services. Users can define privacy preferences for each service type and identify specific sets of data to transfer to a particular service type.

While it is important to provide such tools that allow users to edit their privacy preferences manually, it is equally important to provide mechanisms that learn from the user’s behaviour and create rules on their behalf to unburden them from this task and help them protect their privacy in a more efficient and accurate manner.

Schaub et al. [120] use an incremental approach using case based reasoning to acquire privacy preferences. A privacy preference is stored as a case and is matched to a context transition (the difference between two sequential context states). The decision making step is performed by evaluating the current context state against previous stored context states. A new case (or privacy preference) is then created by combining similar privacy preferences with the current context state. A similar approach is presented in [121] using case-based reasoning with a community portal providing additional information when needed. Learning of privacy preferences occurs when the system recommends courses of action when users encounter new services or services have altered their privacy policies. The information used to calibrate the current case is the user’s current location, current role, time of day, the service, the service provider (as a separate parameter) and the privacy policy of the service.

Researchers at Carnegie Mellon University experimented with case based reasoning for acquiring privacy preferences for Grey [122], a distributed smart phone based access control system. In their evaluation study with real users, they found that learnt rules were

much more accurate than rules created manually by the users who notably spent more than 5 minutes to create the initial rules in the system and then additional time to revise them while interacting with the system.

The DAIDALOS Personalisation and Learning system [123] makes use of Quinlan's C45 algorithm [124] to perform offline learning of user preferences in the form of decision trees using information coming from monitoring user actions as well as explicit feedback supplied by the user. The same approach is used for learning privacy preferences indicating when an identity should be selected by the user to interact with a service is also presented in [125].

Bootstrapping

A new user will not have privacy preferences stored in the system and hence a mechanism must exist to bootstrap the decision making and preference learning process. There are several approaches to solving the bootstrapping issue. In [120], Schaub et al. suggest using "personality-based privacy profiles". Several personality types are identified according to users' data disclosing behaviours. A personality-based privacy profile containing a list of privacy preferences is created that matches the data disclosure patterns of the corresponding personality type. The user can choose the profile that best matches his disclosure patterns and that profile can be used as a starting point for the decision making and privacy preference learning process.

2.3.10 Privacy Policy Negotiation

It is inherent in the nature of service provisioning that, in order for it to be effective in its context awareness and personalization, data must be exchanged and even collected and processed. After such data has been shared, there is nothing a data owner (the data subject) can do to control further processing, forwarding and sharing of it when the data is stored on servers owned by the services where the user has no access. Hence, legislation plays a crucial role in motivating service providers to respect the privacy preferences of the data owners/subjects and adhere to their own privacy policies to which the users consent. Hong [33] agrees that eventually the issue of privacy will be handled by a combination of among other things, legislation, technology and social norms.

Privacy policies are used by service providers to describe what data are going to be collected, how long they will be kept for and how they will be processed (shared with third parties, used for marketing purposes etc.). At the moment, service providers are adopting the "take it or leave it" approach which means that a service privacy policy is static and the user can either accept it and be allowed to use the service or not accept it and be prohibited from using the service [126][127]. This approach is not flexible for the user who has no means to express their terms for disclosure of their data but has to commit to the demands of the service provider who sets the terms and conditions of the transaction. This notice-and-choice model has failed to provide appropriate privacy protection [128]. There is a need to provide the user with some form of control over the terms and conditions of data disclosure, such as privacy agents which are able to negotiate privacy policies on behalf of the user and the service provider. Further, Preibusch [129] identified two shortcomings of static privacy policies not only for users but for service providers as well: a) the "one size fits all" solution for privacy policies is not only problematic for users but for service providers as well because users may be willing to disclose more data for less privacy, and b) the "take it or leave it" approach is a double edged sword. It is the service providers' loss when users turn away from a service whose privacy policy they don't agree with and it is the users' loss when they reluctantly accept the privacy policy even though they don't agree but have no choice.

During the design phase of the Platform for Privacy Preferences (P3P), the intention was to include a privacy policy negotiation mechanism. However, it was not included in the final specification because of its complexity and lack of interest from the industry and concerns that it would not benefit consumers, i.e. users trading privacy in return for better services instead of services competing for customers by providing better privacy features [6]. Nevertheless, it is noteworthy that several surveys and experiments have shown that *"users are willing to trade privacy for convenience or bargain the release of personal data in exchange for small rewards"* [102] [110].

Preibusch proposes a dynamic incremental approach to privacy policy negotiation. Users are presented with a first offer with minimum requirements, very limited personalisation and a small discount. If they accept, they are presented with a further offer that requires more data disclosure but offers better personalisation and further discount. This incremental approach continues until the user rejects the new offer. When the user rejects the offer, they have the option to use the previous offer or to walk away. Preibusch claims

that the sequential nature offers better understanding of the benefits of disclosing sensitive data.

Tamaru et al. [130] propose a Privacy Profile Negotiation Protocol (PPNP) to protect the user's privacy while personalising services offered in public spaces using different profiles based on the trustworthiness of these services. ContactXML, an XML data representation language is used to describe the profiles of the users and PEGASUS is the framework that implements PPNP. Even though the protocol and the implementing framework PEGASUS are very loosely described, it is clear that it is a very lightweight approach that focuses on changing the granularity of (obfuscating) the personal information that is disclosed for different services. The issue with obfuscation is that it can be applied to few context data types so it cannot be the only solution to protecting the user's privacy. The protocol fails to address automatic negotiation between the user and the services' agents, context aware privacy preferences, preference evaluation and the methods that allow the protocol to access user-assessed trustworthiness levels of the services. It places an unnecessary burden on the user who is prompted very often to confirm disclosure of information, enter the trustworthiness level of services and control every aspect of the negotiation.

Lee et al. [131][132] suggest a P3P Privacy Enhancing Agent (PEA) system based on the P3P (Platform for Privacy Preferences) specification and the JADE framework (Java Agent Development Framework) [133]. The Agent monitors the user's transactions on the Web, automatically retrieves the P3P policy from the server and analyses it against the user's privacy preferences and historic transactions. It alerts the user if the policy of the server conflicts with the user's privacy preference. This analysis is coined as the PEA's privacy policy negotiation process, even though the server is not participating in the process of negotiation. The system does not offer an option for the server to alter its policy based on the user's privacy preferences or to meet any constraints the user might want to impose. It is stated that the system makes use of user privacy preferences expressed in P3P and historic transaction information.

Walker et al. [134] present the Privacy Policy Negotiation Protocol "Or Best Offer" (OBO) similar to sellers offering goods at a fixed price but allowing buyers to bargain for a better price. The Or Best Offer protocol claims to overcome shortcomings of previous research such as i) endless exchange of proposals and counter proposals by terminating

after three rounds of negotiation and ii) the inability to determine whether proposals and counter proposals increase the chances of successful negotiation by giving hints to the server on how to satisfy the user's needs. Walker et al. mention that the idea of negotiating privacy policies in each session was rejected by P3P designers due to lack of useful application scenarios.

Another interesting approach is presented by Zhang and Todd [135] who introduce an architecture for context-aware privacy protection in pervasive systems which uses a Privacy Agent to manage policies of the user and to communicate with the data collectors concerning all data exchanges. The proposed system uses P3P to describe policies and suggests extensions to the existing P3P specification to accommodate the representation of non static data such as location, activity or status i.e. the context of the user. Data type ambiguity is handled with the use of a privacy vocabulary (ontology) based on the P3P terminology and policies. The ontology is used to represent privacy rules which are evaluated using ontology-based reasoning engines and eventually used to negotiate with the data collectors about data access. Privacy Agents reside on the network and not on the user's devices in order to save power on the mobile devices. However, this raises the issue of availability of rules over isolated spaces as well as the issue of trusting the network provider that stores the data. The solution proposed is protecting context in ubiquitous computing environments but does not leverage the power of context by embedding conditions of context in privacy rules and evaluation of privacy rules. As a result, it does not react to changes in the context of the user that affect the outcome of evaluating context-aware privacy preferences.

The Houdini Framework developed at Bell Labs [136] provides a policy management infrastructure for personalising services offered on the Web and through mobile telecommunication networks. This framework incorporates a "Privacy Conscious Personalisation" (PCP) engine that can block or filter access to user data based on the user's static and dynamic data and the current context of both the requestor and user. Hull et al. [136] bring attention to the fact that a user's willingness to disclose information may depend on their context and attempt to address this with the use of context-aware controls. Hull et al. also raise the issue of fine-grained preferences being tedious for users to manage or create and suggest presenting the users with web interfaces adapted for small and large screens so they can easily create inference rules themselves. Requestors and context conditions can be grouped together under labels such as "colleagues" or "at work"

to improve performance and aid the user in creating privacy preferences. The Houdini Framework does not leverage any trust management system and therefore has no means for evaluating requestors and treating them accordingly. The Houdini Framework provides a personalised context-aware access control system for user data that works on behalf of the user without involving the requestor in any step of the process, hence there is no data negotiation involved to aid the access control process. Therefore, it does not provide the user with control regarding the further processing and further disclosure of their data.

El-Khatib presents a Privacy Negotiation Protocol (PnP) for Web Services [137] for privacy policy generation and negotiation between two privacy agents. The protocol defines the states and semantics of the messages exchanged during the negotiation expressed using the P3P protocol specification. The presented protocol defines messages that allow a) the service provider's agent to offer a set of terms and conditions in a privacy policy, b) the user's privacy agent to accept or reject them and c) the service provider to suggest a counter offer after a rejection by the user's privacy agent. A limitation on this protocol is that the user's privacy agent is able to accept or reject a given offer but not to offer amendments that suggest the user's wishes. Like Zhang and Todd, El-Khatib introduces extensions to the existing P3P protocol specification to enable the expression of alternative terms and conditions within the privacy policy document.

Although adequate research has been conducted on the field of privacy policy negotiation and a number of protocols have been designed, there is currently no standard accepted by the standardising organisations or adopted by the industry [138].

2.3.11 Identity Management

Identity Management is a term used to describe the means by which individuals are authenticated and authorised to access services. Depending on the type, Identity Management systems involve issuing identities, managing the information associated with identities, providing identity interoperability across multiple domains and minimising data disclosure. Privacy Enhancing Technologies (PETs) cannot exist without a proper Identity Management system. Transactions with services require users and service providers to use some form of digital identity to represent themselves in the digital world. Traditional Identity Management systems are centred on organisations that use

common identifiers to share information about individuals, an approach often termed "back-office" Identity Management. Such systems may operate without the consent of the individuals and as a result jeopardise privacy. There is consensus that some form of organisation-centric Identity Management is necessary in the community and more specifically by the government for the purposes of national security, crime prevention and detection. Many governments already employ systems such as national identity cards which can be used to identify citizens when using public services. Like governments, companies also employ organisation-centric Identity Management systems for their employees to identify themselves to the system. However, organisation-centric Identity Management systems are necessary to operate only for specific purposes. In order to address concerns over the organisation-centric Identity Management systems, user-centric Identity models are currently being researched that focus on empowering the user with controlling their online identities and the disclosure of their personal information. OpenID [139], WebID [140], InfoCard [141], and Liberty Alliance [142] initiatives are examples of user-centric Identity Management and personal information management systems (UC-IPIM). The user-centric approach attempts to follow the type of identification individuals encounter in their non-digital everyday lives. Several types of Identity Management Systems are described below.

Identity “Silos”

Identity silos refer to the traditional identity schemes in which the relying party is also the Identity Provider and the identities used for these systems are only valid for authenticating to the relying parties they were created for. This category includes the millions of websites that exist in the Internet today that require the users to register with each one of them in order to use them.

Generally, the advantage of this model regarding privacy is that any personal information is only disclosed to the relying party. The disadvantage of such a scheme is that a user has to maintain too many identities leading to “password fatigue”, a term used to describe the problem of having to remember a different password for every different account opened. This usually leads to users using the same username and password combinations for different services which results in a greater security threat [143],[141]. Someone acquiring the username and password from one service, can access the account of the user in other services. Identity silos are Identity Management schemes which do not allow

interoperability between different service providers, therefore data that is maintained on one service cannot be ported onto another. The identity silo model cannot exploit the functionality of pervasive computing which offers services the option to be composed with other services to offer a greater range of services.

Common Identity

A “common identity” management scheme means that users will have a unique common identity to use in more than one website, having a unique global identifier for authentication. In this scheme, a single Identity Provider issues the identity but does not take part in the authentication process. Services have to implement their own authentication mechanisms. In terms of personalisation, this approach is ideal as it gives access to all information held about the user and enables any personalisation practice. This applies to pervasive systems as well as traditional computing systems. However, one of the many disadvantages of this approach is that it does not provide any privacy. Since the same identifier is used to access all services, different services can match the unique common identifier provided to them and acquire personal information. An example of a common identity scheme is Microsoft's Passport .NET Identity scheme which can be used to login to Hotmail, Expedia, Live Messenger etc.

Centralised Single Sign-On (SSO) Identity

The Single Sign-On Identity is an Identity Model that features an Identity Provider that provides security token issuance, identity authentication and acts as a central authority. The user is issued the identity by an Identity Provider and can use the same identity with multiple services as long as the services implement the identity scheme. A famous example is Microsoft's .NET Passport which was rejected by the Internet community [144] because it was too expensive, used proprietary software and had Microsoft at its centre acting as the identity provider, something that users felt was not appropriate. Kim Cameron, author of the Laws of Identity [145] criticised the .NET Passport identity management system for violating these laws. Later on, he became Microsoft's Chief Identity architect where he helped design the Windows Live ID system which addressed the violations of the Passport .NET identity scheme. One of the privacy concerns that were raised regarding the .NET Passport scheme, was the fact that there was omnidirectional flow of information data between the .NET accounts and the services that used .NET Passport authentication providing Microsoft and the participating service

providers with access to information about the users' interactions with services. It should be noted though that when the .NET Passport was first introduced, it was intended to be used for unidirectional flow of information data but services were too interested in gathering personal information on users for targeted marketing purposes.

Centralised Control of Multiple Identities

Microsoft's Identity Meta-System [146] incorporating the Windows CardSpace Identity module [141] (formerly InfoCard) is an example of this type of Identity Management. Windows CardSpace architecture is based on 4 requirements:

Support for any digital system. As shown in Figure 2, Cardspace (named Infocard in the diagram) works with the application on the user's terminal and passes information between the Identity provider and the relying party. Cardspace does not need to understand the technology of the security token communicated between the relying party and the identity provider. When the security token requirements are received from the relying party, Cardspace only needs to select an identity with a security token which matches the technology requested by the relying party.

Consistent user control of digital identity. Figure 3 shows the user interface designed to allow the user to select an appropriate identity for a given transaction. This step, as shown in Figure 3, takes place just after the relying party conveys the security token requirements to Cardspace. Providing users with a consistent way to select their identities reduces the possibility of confusion and error and inevitably leads to good privacy protection practice.

Replacement of password-based Web login. Username-Password logins are susceptible to "phishing" and for the user to maintain a combination of these for each website she uses is hard and usually leads to using the same username and password for all web accounts which is not good practice [147] [143]. To replace password based web logins, Cardspace offers a self-issued Identity Provider mechanism which allows the user to create their own identity cards acting both as the identity provider and guarantor, generating a private/public key pair to use for authenticating themselves to the relying party.

Improved user confidence in the identity of remote applications. Windows Cardspace provides mechanisms to verify that a relying party or an identity provider is who they

claim to be. Microsoft has acquired Credentica's U-Prove Technology [148] which claims to provide features such as selective data disclosure, identity untraceability and unlinkability.

The concept of this scheme is aligned with the principles of pervasive and ubiquitous computing. Its user-centric nature can be received well by the public who want to be in control of their online identities. However, since its inception in 2005, the scheme has not been adopted by users even though it is available on the Windows operating system. Lack of scalable business, liability and appropriate governance models are listed as potential reasons [3].

.....

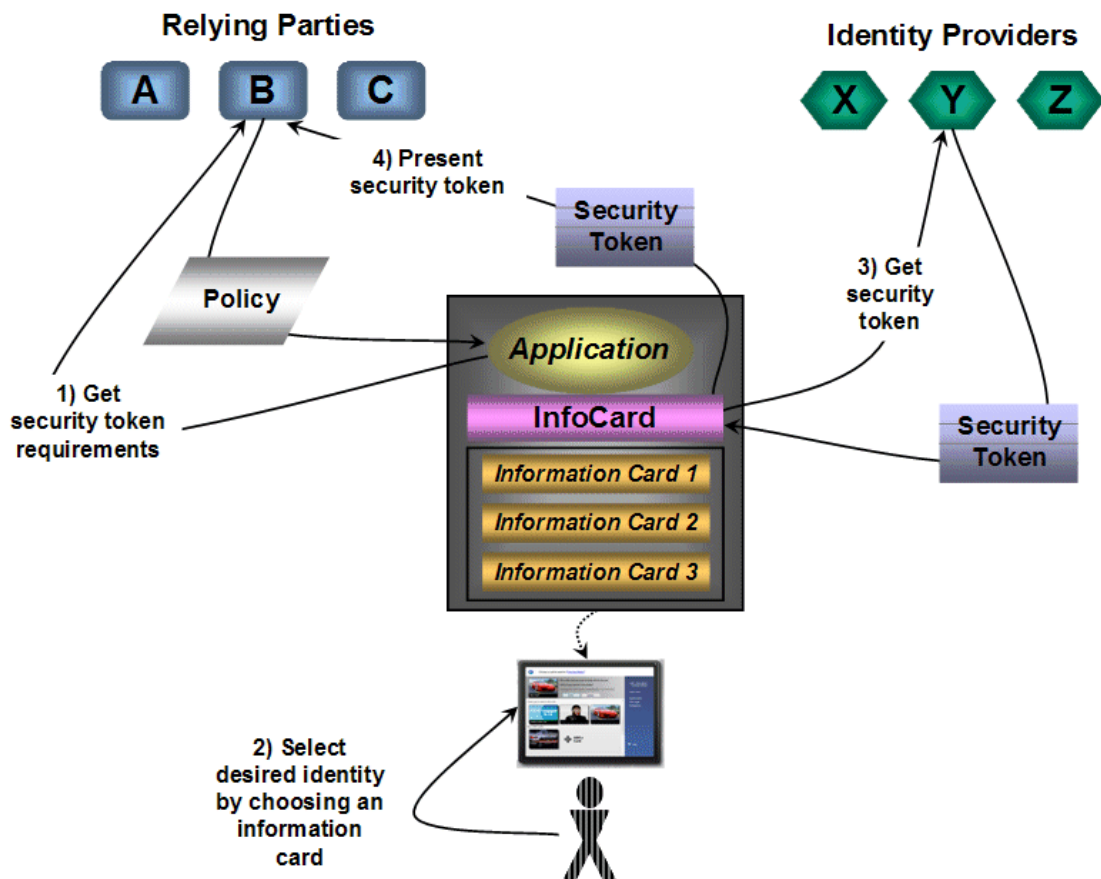


Figure 2. Interactions between the user, identity provider and relying party.

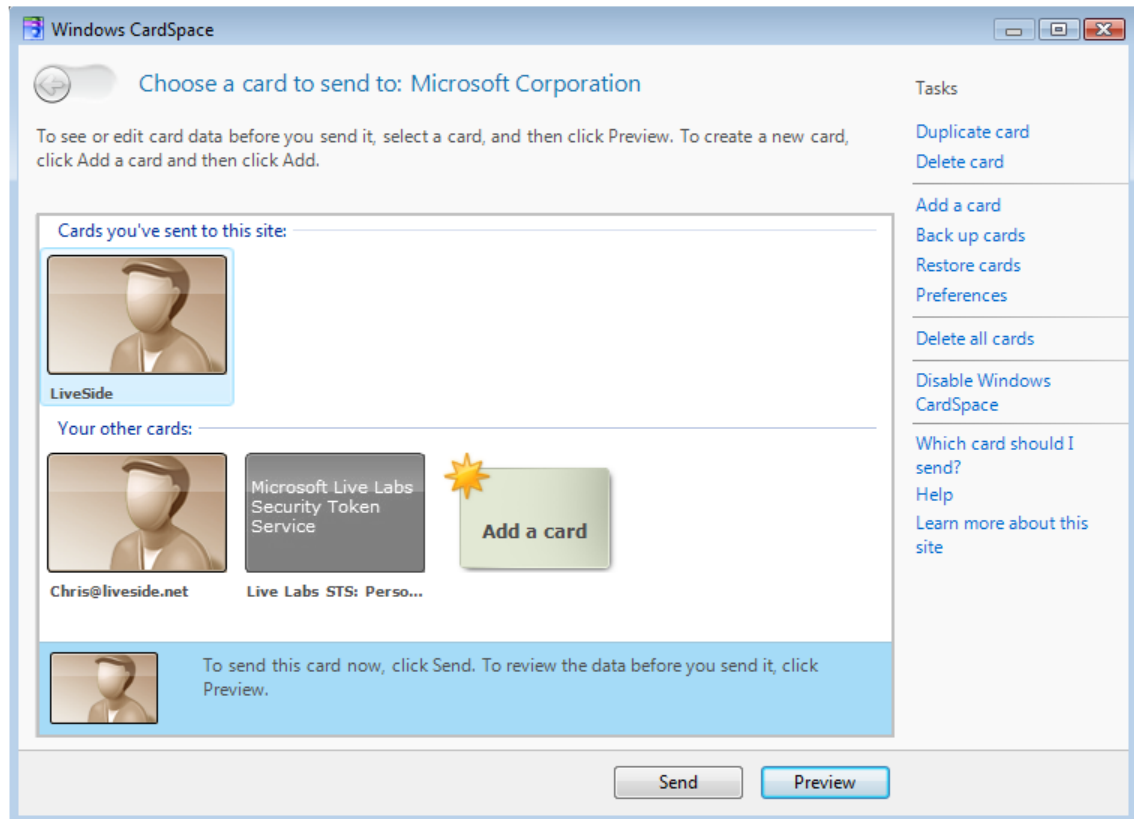


Figure 3. User selects an appropriate card to interact with service.

Decentralised Single Sign-On Identity

Bhargav-Spantzel et al. [149] present the basics on federated identity management systems (IdMs) and their disadvantage of not providing selective release of identity contents. They claim that in order to achieve that, a user should hold more than one identity which contradicts the purpose of Federated SSO (Single Sign On) IdMs, which is minimizing the management of multiple profiles by the user. Instead, to cater for such a requirement, a trust negotiation framework is preferred and presented. Bhargav-Spantzel et al. propose an approach to integrate federated IdMs with trust negotiation techniques resulting in the FAMTN (Federated Attribute Management Trust Negotiation). Key aspects of the FAMTN are a) a federated attribute need only be provided once, b) Identity verification need not be repeated by the user. (FAMTN participant components perform internal negotiations using a “circle of trust”) and c) support for temporary SSO.

The FAMTN Framework has been developed by the Liberty Alliance project [142]. It aims to provide federated identity management between multiple service providers (SPs) that implement Single Sign-On, with automated trust negotiation mechanisms for

enforcing access policies when these SPs request user information from the Identity Providers (IdP) [149]. Liberty-Enabled Services that implement Single Sign-On belong to the *circle of trust*, a federation of service providers and identity providers in the Liberty architecture. Any Service Provider in the Liberty architecture can also be an Identity Provider so the FATMN does not distinguish between the two.

FATMN allows users to login to a service provider using a Liberty account and navigate any Liberty-enabled service without having to repeat the login process. User's credentials are transferred from service provider to service provider as the user moves from one to the other. When a service provider joins the *circle of trust*, trust agreements are created which are later used as input to the negotiation process that will determine what types of information the service provider is allowed to access. User privacy preferences are also used as input to this negotiation process.

Service providers build a history of user transactions and user attributes that they have gathered during the user's interaction with the service. User attribute access is negotiated between service providers in the *circle of trust* by an Automated Trust Negotiation mechanism. Automated Trust Negotiation enables service usage without the user having to enter the same details more than once.

Shibboleth [150] is an initiative designed for resource sharing between research and academic institutions. It follows a very similar approach to the Liberty Alliance project, except that it uses a central identity provider that negotiates the federation of user attributes every time the user moves to another Shibboleth enabled domain.

Liberty and Shibboleth could prove very complex but powerful identity management systems in pervasive systems mainly because they widely support federation, a key concept in pervasive systems whose business models are mainly focused on having telecom operators taking the role of Identity Providers.

2.4 *Summary*

A pervasive and ubiquitous system is a context-aware, personalisable, dynamic, self-adaptive environment of connected devices and resources such as sensors, mobile devices, back-end servers and services. The aim of pervasive computing is to provide an environment to the user that adapts to their current needs and preferences to help them

accomplish their goals in their daily life. Information about the user such as their interests, preferences, location, environmental conditions, activities social interactions, physical capabilities or disabilities and other similar information is collected from the users, the applications they interact with and sensors attached to their environment and drives the behaviour of a pervasive system. The user and their environment are constantly monitored which raises a lot of privacy issues. This has been a source of controversy over the use of pervasive systems. The protection of this information is vital to the adoption of pervasive technologies.

There is a consensus that there is a balance to be struck between the user's right to privacy and the need to provide information to take advantage of the functionalities of a pervasive system. This balance depends on a variety of factors such as contextual integrity, i.e. the user's trust that service providers meet the users' expectations with regard to privacy, the provision of appropriate mechanisms to control the flow of information to external parties, the provision of useful and user friendly interfaces to manage the user's data, the act of data minimisation and proper data retention on the part of the service provider.

The right to privacy needs to be embedded into the design of pervasive systems to ensure that appropriate mechanisms exist to protect its users. Privacy by architecture is one means to remove the need for disclosure of information. However, this approach is very limiting for pervasive systems. Disclosing information is vital for a pervasive system to function properly but users need to be able to control who gets access to what information, when and under what conditions. Bijwe and Mead [66] express their view that users' privacy is threatened by authorised users of the system. Altman's theory [36] [37] of optimum privacy is defined by the user's preferred level of social interactions. If users can dictate when, who, under what circumstances what information is released, they could get closer to achieving the optimum level of privacy. Privacy policy negotiation and the use of multiple identities are two solutions that can be combined to equip the user to accomplish their optimum level of privacy.

As the number of services, resources and the amount of data continuously grows, the user is faced with a heavy burden in controlling all this and in some cases, does not have full knowledge of how to accomplish this properly. Privacy Enhancing Technologies (PETs) are required to help users overcome this burden. Privacy preferences can be used to express the user's wishes regarding the disclosure of their personal information. These

are rules that control how the system releases personal information to external services. Such privacy preferences must comply with the requirements of pervasive systems that are constantly adapting based on the user's context. Appropriate privacy preference models must be context-dependent to allow for different disclosures to be performed in different contexts.

It is unwise to expect users to write privacy preferences that anticipate every possible data disclosure scenario. The user's data disclosing behaviours and the context in which they were performed can be monitored and analysed using behaviour learning algorithms to automate the acquisition of privacy preferences and alleviate the user from this burden.

This chapter provided an introduction to pervasive systems and personalisation and an overview of privacy issues and approaches to these issues particularly in the context of pervasive computing. Chapter 3 gives an overview of three EU research projects DAIDALOS, PERSIST and SOCIETIES that researched pervasive and ubiquitous systems and produced working prototypes. The author was involved in researching, designing and implementing their respective personalisation and privacy frameworks.

3 EU Research

The research presented in this thesis is part of a complete framework for providing personalisation of services and protection of privacy for users of pervasive ubiquitous systems. This chapter provides background and insight into the way in which the PersoNISM system has evolved into its current form.

The research reported in this thesis began with the author's participation in the EU FP6 DAIDALOS project [151]. Two of the basic assumptions of this project were that a telecommunications infrastructure would be available at all times and that services would be provided by network operators or Internet vendors and could safely be assumed to be available from any fixed location. Based on this assumption, an architecture was developed in which privacy and security features were based on a centralised Identity Management system whose architecture impacted the design of the platform components including those providing personalisation features.

After the end of the DAIDALOS project, the focus of the research shifted to the EU FP7 PERSIST project [152] which was based on the Personal Smart Space paradigm (PSS) that used a peer to peer approach and hence could not rely on a centralised Identity Management system. Furthermore, any user could assume the role of a service provider even when they were mobile allowing interaction in an ad hoc manner as well as through existing infrastructure. As a result, the PERSIST prototype is a much more open system as it allows any user to participate as a user or as a micro-operator or both.

To achieve this, the PERSIST privacy protection framework was developed based on higher level privacy protection and using a lightweight proprietary Identity Management system, privacy policy negotiation and identity selection. To ease the user with the use of such features, privacy preferences were used in the system to guide the user as well as automate some processes on their behalf according to their wishes.

Research conducted on privacy preferences, privacy policy negotiation and identity selection in the context of the PERSIST project provided initial feedback on the feasibility and problems with this approach. The final prototype for the privacy preference data model, privacy policy data model and management and privacy policy negotiation protocol was designed and built by the author who also assisted in the design and

implementation of the lightweight Identity Management System used. This was successfully demonstrated at the final audit of the project.

After the end of the PERSIST project, the research continued under the auspices of the EU ICT FP7 SOCIETIES project. The SOCIETIES project aims to deliver a prototype of a Cooperating Smart Space (CSS) extending traditional pervasive paradigms that are focusing on the individual by leveraging the power of dynamic communities as well as social networking integration. The platform exploits the combination of the digital and the physical environment of its users to foster Pervasive Communities that feature, among others, community enhanced proactive personalisation and strong privacy protection using privacy policy negotiation, context-aware dynamic access control, trust management, dynamic data obfuscation and identity management. Like the PSS, the CSS also encourages users to become micro-operators by sharing services within communities but unlike the PSS, it does not provide a traditional peer to peer approach but instead uses the backend cloud servers to host platform services that require device resources that are not suited for mobile devices.

The DAIDALOS and PERSIST projects followed a scenario-driven approach to designing their respective architectures and demonstrating and evaluating the project outcomes. Videos of the implemented scenarios for DAIDALOS can be found in [153], and for PERSIST in [154], also more information on the PERSIST demonstration can be found in [155]. Finally, information on the results of the SOCIETIES user trial can be found in [156].

3.1 *EU IST-DAIDALOS*

DAIDALOS (Designing Advanced network Interfaces for the Delivery and Administration of Location independent, Optimised personal Services) was a multidisciplinary EU FP6 IST project that ran from January 2004 until December 2008 in two phases with participation of over forty partners from industry and academia. This section concentrates on the architecture developed during the second phase of DAIDALOS. The vision of DAIDALOS was to integrate multiple existing and new heterogeneous network technologies in a Beyond 3G pervasive infrastructure that provides access to services and content supported by context and personalisation. DAIDALOS developed this infrastructure centred on 5 key concepts:

- MARQS: Mobility Management, AAA[Authentication, Authorisation and Accounting], Resource Management, Quality of Service and Security
- VID: Virtual Identity
- USP: Ubiquitous and Seamless Pervasiveness
- SIB: Seamless Integration and Broadcast
- Federation: enabling provision of services across multiple domains and networks

The VID and USP concepts had a larger impact on the design of the DAIDALOS architecture than the rest. The Virtual Identity concept is at the core of the DAIDALOS architecture, affecting all layers and nodes of the system and it is the cornerstone of the privacy protection technologies designed for DAIDALOS. DAIDALOS users use Virtual Identities to represent themselves to the platform and to the services they interact with. The DAIDALOS Virtual Identity framework was designed to provide users with multiple identities for use in different situations and with different service providers. This concept requires that the identities owned by a single user cannot be linked together.

The Ubiquitous and Seamless Pervasiveness concept refers to two requirements. a) ubiquitous access to services: the ability of the system to provide its services and networking capabilities irrespective of the location, network technology, device architecture or service type and b) seamless pervasiveness: the ability of the system to monitor the user in their environment, anticipate their needs and act appropriately to aid them in their tasks in a seamless manner. The USP concept is realised in the DAIDALOS Pervasive Service Platform (PSP) layer while the Virtual Identity Model is realised in all of the layers of the DAIDALOS architecture as it is used to create Virtual TCP/IP stacks per Virtual Identity [157].

As mentioned previously, a scenario driven approach was followed for performing requirements collection, analysis, prototype design and demonstrating the outcome of the DAIDALOS project. The Nidaros scenario [153] focuses heavily on demonstrating the use of preferences in a mobile networked environment. Intelligent interface selection (IIS) depends on user preferences to perform network handovers. User preferences define the parameters that should be taken into account such as Quality of Service (QoS), cost,

current identity, etc. Network handovers can be performed between UMTS, DVB and WLAN networks according to the user's preferences and current context conditions.

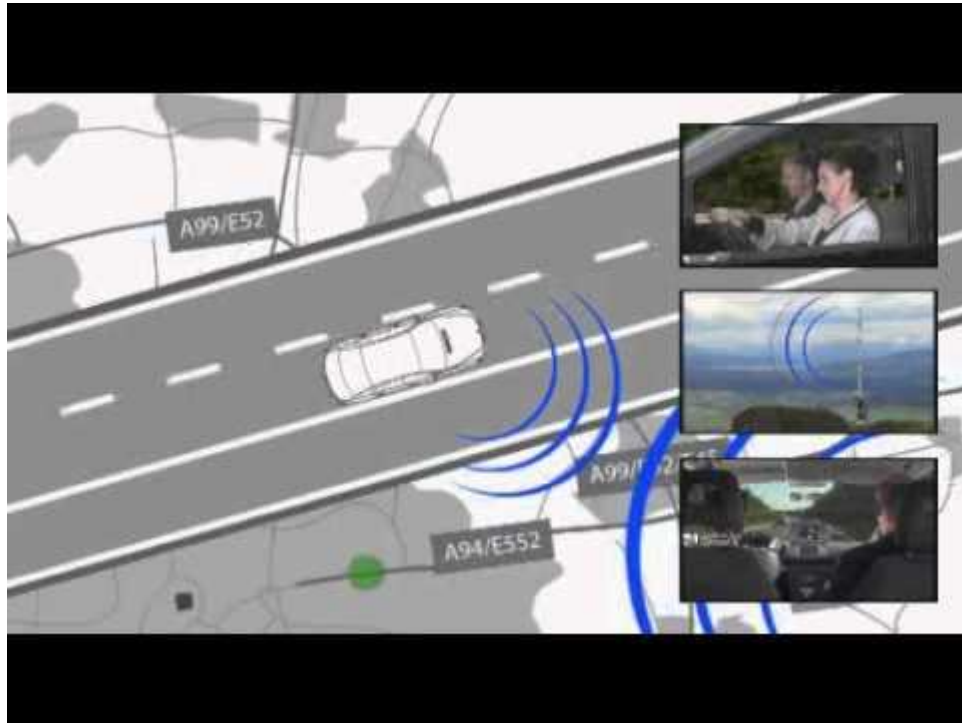


Figure 4. *DAIDALOS "Nidaros" scenario screenshot*

Further, the Nidaros scenario demonstrates context aware personalisation when a student's mobile phone automatically mutes itself when the student enters a classroom while a lecture is taking place. The phone will automatically unmute itself as soon as the student leaves the classroom.

3.1.1 DAIDALOS Pervasive Services Platform

The DAIDALOS Pervasive Services Platform (PSP) has two layers. The Service and Identity Management layer is the lower layer responsible for service management including service selection, ranking, composition and re-composition and identity management such as creating, deleting, amending and selecting appropriate identities for use with services. The Service and Identity Management layer hides its functionality from the upper User Experience management layer and makes sure that the platform operates with multiple identities for each user without being able to link together the identities owned by the same user. The User Experience Management layer includes the platform services that enable the system to operate in a pervasive, personalised and context-aware fashion. The main building blocks of the User Experience Management layer are the Personalisation and Learning system and the Context Management system. The Service

and Identity Management layer provides the basic functionality for pervasive services to be deployed and instantiated in the DAIDALOS platform and therefore realises the ubiquitous access to services requirement while the User Experience Management layer provides the necessary enhancements for pervasive services to adapt to an individual user based on their context and user preferences and realises the requirement for seamless pervasiveness.

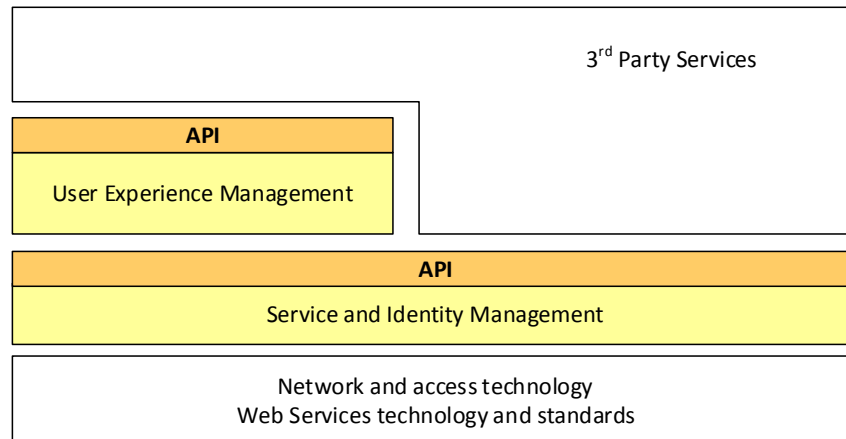


Figure 5. *DAIDALOS PSP two layer architecture approach.*

DAIDALOS assumes the existence of infrastructure and backend servers that handle the processing and storage of information. These backend servers are owned by network operators and service providers which can also act as Identity providers (IdPs). This raises a privacy issue as the user is not fully in control of the storage of their information and there is no provision for storing personal information on the users' devices. Instead, all information is stored and processed in centralised servers owned by Identity or content providers.

Personalisation in DAIDALOS

The User Experience layer of the DAIDALOS PSP platform provides personalisation and context-awareness for DAIDALOS enabling services (platform services) and third party services. The aim of personalisation is to adapt the functionality and behaviour of the system so that it performs differently according to the user's context, preferences and the available services and resources at the time. DAIDALOS personalisation utilises user preferences to personalise the following functionalities.

Service Ranking and Filtering. Context-dependent user preferences are used to filter and rank the results of a service discovery query to provide users with a list of services that are relevant to their context (i.e. based on location and proximity) and preferred by them (i.e. based on cost and quality of service). Personalised Service filtering discards services that are not preferable or relevant to the user and personalised service ranking sorts the list of discovered services based on personalised parameters such as distance, cost, quality, provider preference etc.

Service Selection. User preferences can be created to dictate specific services to be used in specific situations and for particular tasks.

Service management. User preferences are used to trigger the system to initiate, terminate or recompose a session in different context conditions. A simple example of personalised service management is to turn on the air conditioning service when a user enters their office building.

Service adaptation. The most common use of user preferences is to personalise the parameters of a service. This can range from setting the wallpaper of a user's desktop to adapting the heating temperature of a room based on room occupancy.

Communication Redirection. User preferences are used to redirect calls and messages to different devices depending on the context of the user.

Network Selection. Context-aware user preferences are used to select the network and network interface that should be used at any instant depending on network availability, network quality of service (QoS) and cost.

Identity Selection. The user is able to create user preferences that define which identity should be used with specific service types, services and in specific situations.

The DAIDALOS Personalisation subsystem exposes appropriate interfaces to allow services (DAIDALOS enabling services and third party services) to request preferences in order to personalise themselves at any time. However, it also acts as a proactive system by constantly monitoring the context of the user, evaluating the user preferences that are

affected by the changes and re-personalising the services according to the result of the evaluation.

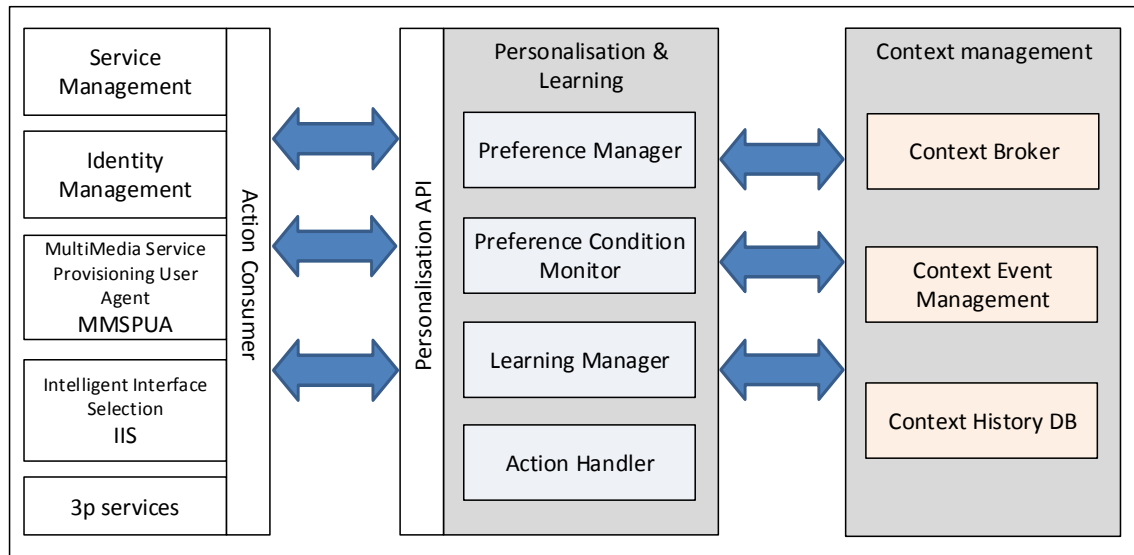


Figure 6. *Personalisation Subsystem interactions with services.*

Privacy Protection in DAIDALOS

The Privacy Enhancing Technologies (PETs) researched, designed and implemented for the DAIDALOS architecture stem from the design of the DAIDALOS Virtual Identity Framework [158]. At the core of the VID Framework lies the concept that the user profile (consisting of all the data that exist about the user in the system) can be fragmented both physically and conceptually. The user's data reside on different servers owned by infrastructure vendors such as identity providers or content providers. A Virtual Identity is linked to selected data from various sources and used to represent the user to services including platform services.

Users interact with services in the system by first exchanging their identities through a process called VID exchange. The VID exchange includes negotiating anonymously a set of terms and conditions before either party selects what VID to use. During the negotiation, the service provider reveals some information about themselves to provide assurances to the user of the service that they can provide. The service provider may also request information from the user such as user age or location if the service can only be provided to adults or to a specific geographic area. After such requests have been satisfied and based on the information that has been exchanged, each party selects a VID to represent themselves.

Any user can have multiple VIDs to represent themselves to different services and in different situations. Deciding when and with which service to use a specific VID is provided by the DAIDALOS personalisation system which allows users to create user preferences to indicate under which circumstances (under specific context) a VID should be used.

The DAIDALOS Virtual Identity Framework

The basic concept of an Identity Management system is the user profile. The DAIDALOS Virtual Identity model defines the Entity Profile as the set of all data that exists about the user in all the federated domains that he is registered with. The term "entity" refers to a user, a group or organisation. The Entity Profile (EP) is a logical concept that refers to the set of all the data of a user as stored anywhere. The Entity Profile consists of Entity Profile Parts (EPP) where each Entity Profile Part is the set of data stored in a single domain. Network operators, service and content providers create EPPs for their users. EPPs contain personal information, context information and preference information. An Entity Profile View (EPV) is a partial view of the Entity Profile and as such it is linked to one or more Entity Profile Parts as seen in Figure 7. The EPV is also termed the Virtual Identity (VID). During the VID creation phase, the user can select the Entity Profile Parts they want to associate with their VID. This linking does not automatically mean that by giving a VID to a service, it can access any of the data held in the EPPs linked to this VID. Access control rules are configured to control data disclosure on a per request basis. An EPV with configured access control rules is a filtered EPV (fEPV). This enables the user to use the same identity with more than one service but assign different access permissions for each service.

There are two basic components that constitute the Virtual identity Framework: the Identity Management component and the Identity Broker component. The Identity Management component is part of the DAIDALOS PSP. Its functionality is VID creation, deletion and management that it delegates to the appropriate IDBroker components, and VID selection for use with services. It is the only component that holds the necessary information to link together the VIDs of the user. The Identity Broker component is deployed on the operator domain and handles requests for managing (creating, deleting, amending) Entity Profile Parts that are created on that domain. Every domain has an identity broker component to handle the management of the EPPs on that domain. Its

main responsibility is the mapping of Entity Profile Parts to VIDs, routing and proxying the requests for data held in EPPs. Moreover, the IDBroker is not able to view the actual data stored in the EPPs but can only link VIDs with the URIs of the Entity Profile Parts with which they are associated.

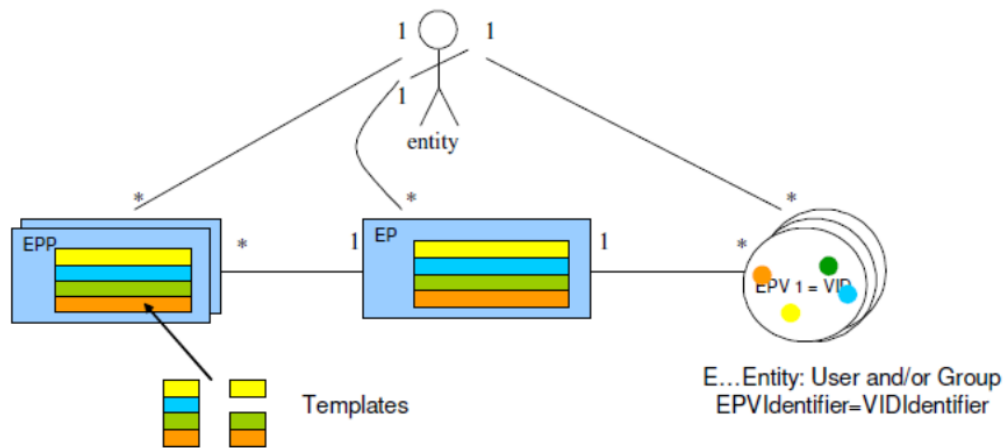


Figure 7. *DAIDALOS Virtual Identity Model*

3.1.2 Lessons Learnt

Fragmented user profile

Any system that attempts to protect the user's privacy has to be flexible enough not to restrict the performance of other components of the system. The DAIDALOS Virtual Identity model had an innovative design for an identity model. However, the fact that the identities of a single user could not be linked together inside the system that supported them was very restrictive for personalisation and learning purposes. Behaviour history was accumulated separately for each identity the user created and therefore, the learning capabilities of the system were severely impeded. Preferences could not be imported for use with new identities as the system was handling the different identities of the same user as multiple users. The same problem affected the use of context inside the system. As different context information was accessible using different identities of the user, the personalisation components were unable to use this information to create rich context dependent preferences or react appropriately to changes in the context of the user.

Infrastructure requirement

DAIDALOS could only operate with a fixed infrastructure and placed a lot of emphasis on the role of telecom operators and network providers. These entities were entrusted to

provide both secure storage capabilities for information as well as behaving as the hosts of the system and therefore enabling the functionality of the system. The fact that the system was not focused on the user but was driven by requirements set by the telecommunications industry impacted on the users' ability to have ultimate control over their data. The DAIDALOS system did not provide any means by which a user could host and manage their data on devices they own themselves. Instead, the DAIDALOS system required that all data would be stored and managed centrally.

Anonymous negotiation

User anonymity is a critical requirement in a negotiation process as any data the user discloses during the negotiation should not be linked to the real user until the negotiation succeeds. Even then, anonymity might also be maintained if needed. While it is important that the user remains anonymous during a negotiation process, there is no reason for a service provider to remain anonymous. On the contrary, the user needs as much information about the service provider and their privacy practices as possible to be able to provide informed consent about disclosing their data.

3.2 *EU ICT-PERSIST*

PERSIST (Personal Self Improving Smart Spaces) was an EU FP7 ICT project that ran from April 2008 until October 2010 with participation from industry and academia. The project was given an excellent review and was considered a flagship project of the EU when the project completed. The vision of PERSIST is of a Personal Smart Space encompassing the devices, portable and stationary, a user makes use of in their daily routine. The Personal Smart Space moves around with the user, providing context-aware pervasiveness in a ubiquitous and seamless manner. The Personal Smart Space surrounds the user at all times catering for their needs, adapting the devices and the environment they inhabit at any time and learning from the user as it constantly tries to improve the user experience.

The objective of PERSIST was to develop Personal Smart Spaces (PSSs) that expose a minimum extendable set of functionalities that can be enhanced by interactions between users of PSSs during their everyday activities. Learning and reasoning, personalisation using preference and user intent models, context-awareness and proactive behaviour are at the core of the PSS architecture. PSSs encourage users to interact with other PSSs to

enhance their pervasive experience. Services, resources and information can be exchanged and shared with others with the aim of helping users to accomplish tasks in their daily routines.

3.2.1 The Personal Smart Space

A Personal Smart Space (PSS) is a collection of devices that belong to a single user. A Personal Smart Space is a logical entity rather than a physical one. It is based on a network of devices that communicate with each other and are aware that they are part of a group of devices aiming to aid their owner in their daily tasks. A PSS can interact with other PSSs and promotes the provision and consumption of services and exchange of information. Services installed in any of the devices of a PSS can be advertised on the network so that other PSSs can consume them when needed. This section provides a view of the architecture of the Personal Smart Space and describes the components that are necessary for the operation of the personalisation and privacy subsystems.

The architecture of a PSS-enabled device consists of five layers each of which incorporates various components and component blocks essential to the operations of the PSS in a pervasive environment. The PSS Framework is situated on the four higher layers while the lowest layer represents the underlying system software and hardware. Figure 8 provides a simple view of the PSS framework architecture. The Personalisation and Privacy subsystems depend on some of the other PSS Framework subsystems to function properly. The Context Management is the most significant subsystem with regard to the personalisation and privacy subsystems because it adds dynamic context awareness to the preferences as well as being used as a permanent storage medium for preference data.

As in the case of the DAIDALOS architecture, the PERSIST Identity Model affects the entire PSS Framework as all operations are performed based on the identity used by the user at any time. Preferences are identity dependent allowing personalisation to be performed differently for each identity of the user but can be shared between multiple identities to allow the same functionality to be applied with different identities if needed. The Identity Model is at the heart of the Privacy Protection Framework because any disclosure of data is performed according to the identity of the PSS requesting the data and the identity of the PSS owning the data.

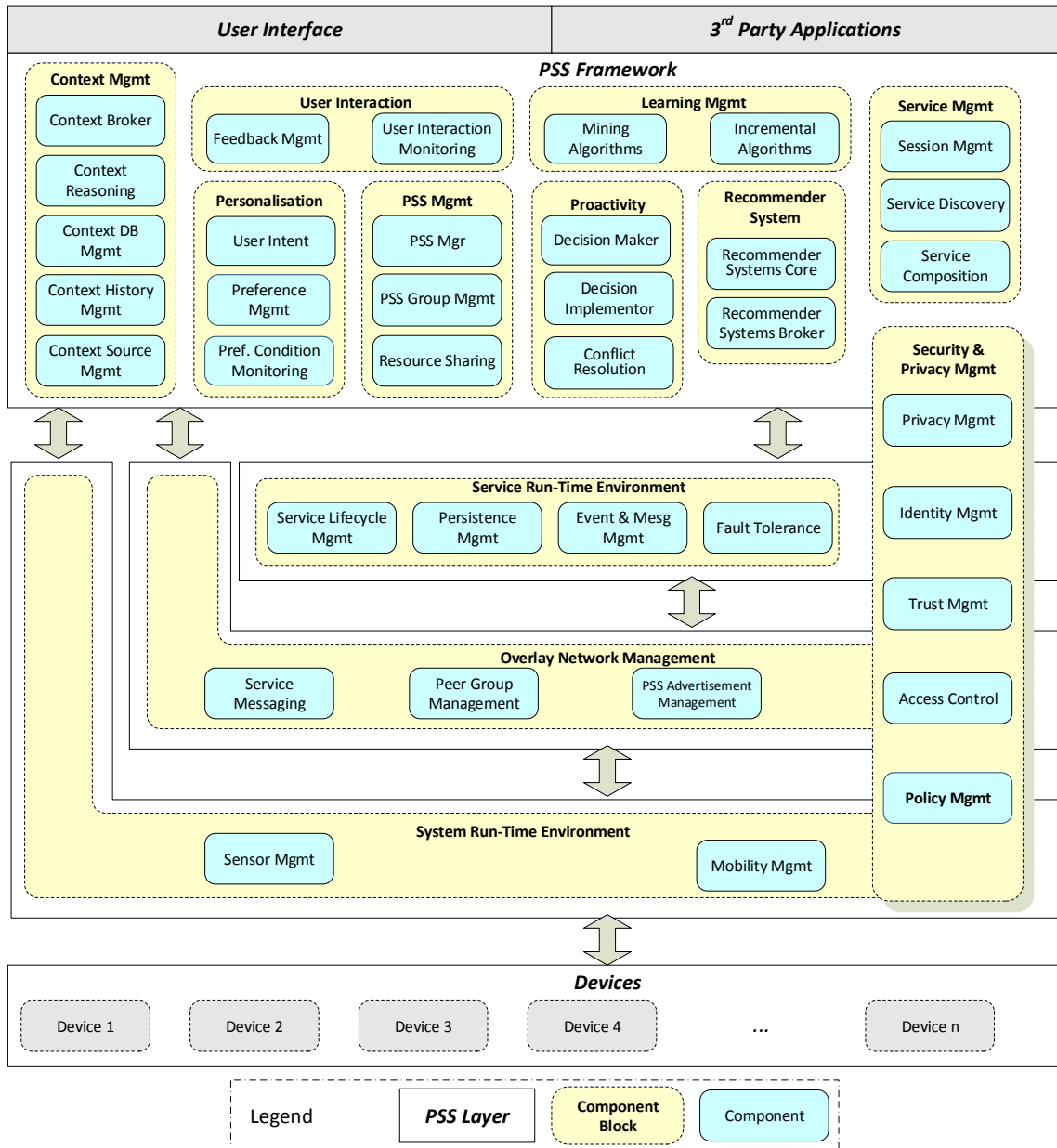


Figure 8. *PERSIST High level architecture diagram.*

PERSIST also used a scenario-driven approach to collect user and system requirements while designing the architecture of the Personal Smart Space platform. The focus of the scenarios were on demonstrating the personalisation and learning functionality embedded in the Personal Smart Space platform. Contrary to the DAIDALOS approach, the effect of the personalisation and learning were demonstrated using a visualisation tool that showed a room in which actuators applied the personalisation in an evident manner. The room showed furniture that could be moved or hidden to configure the room according to its current use. An office room could be configured into a play room or a bedroom depending on the current user’s context and their role in the room.



Figure 9. *PERSIST* visualisation – Reconfigurable room in “office” mode.



Figure 10. *PERSIST* visualisation – Reconfigurable room in “entertainment” mode.

3.2.2 Some aspects of the PSS Architecture

User Context

The Context Management System provides modelling, management and eventing for user context information. The data managed and stored in the context database are used for modelling the user's environment. Information acquired from sensors attached to the user, their devices and the environment is collected and stored in the context management system. The Context Management system enables internal and external components controlled access to context and profile data.

User Context is modelled using three principal model types; Entity, Attribute and Association. Entities model an object of the physical or the conceptual world. For example, a person, a device or a service can be defined as an Entity. Entities have properties and these are defined using the Attribute construct. For example, as demonstrated in Figure 11, if an Entity is modelling a person, some of this Entity's Attributes can be modelling the person's age, name, telephone, location, heart rate etc. Associations are used to link Entities with each other. For example, an Entity "Person" can be linked to another Entity "Person" using an Association of type "IsFriendsWith". Each instance of an Entity, Attribute and Association has a unique context identifier. This

identifier includes, among other information, the identity of the user. For example, the context identifier of an attribute referring to the user's location can be represented as: `pss://user%5B5e0ed9b9-e5d1-4af3-b26e-b33be80de73b%5D@ac3fc8a2a4490bd6a8/ENTITY/person/1/ATTRIBUTE/location/3`. As can be seen, the identity (in bold) is embedded in the context identifier. When context information is disclosed to another PSS, the context identifier given to other PSSs has to be altered to contain the identity that was selected to communicate with that PSS. Thus, unlinkability between identities is maintained and partitioning the profile data of the user is performed so that none other than the user of the PSS can gain access to more information than the user wishes to disclose.

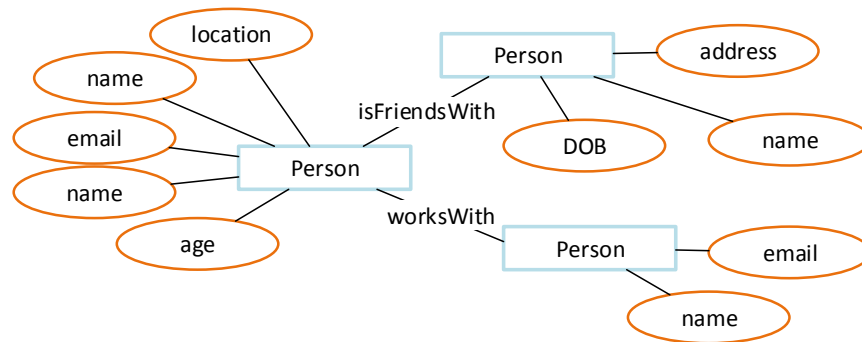


Figure 11. *PERSIST Context Model*

The context management system is a group of components that handle the management of the context data such as storing, retrieving, collecting sensed data and logging the history of each context attribute. The Context Broker component offers interfaces to enable other components to store and retrieve information from the Context Management Database. The Personalisation and Privacy Protection Systems make use of these interfaces in order to store and retrieve user preferences, privacy preferences, and privacy policies for the services provided by the PSS and their respective preference registries as well as requesting context data for the purpose of conducting preference evaluation. They also provide event management functionality for informing listeners of a change in the context of the user or their environment. Any component can register as a listener with the Context Event Management subsystem for a specific context attribute and be informed of any changes to that attribute's value.

User Interaction

The User Interaction subsystem comprises the User Interaction Monitor and the Feedback Management components. These components are necessary for retrieving information from the user and the services currently being used.

The User Interaction Monitor is responsible for monitoring and logging the interactions of the user with services. This component has very similar functionality to the Action Handler component of the DAIDALOS architecture. As users interact with services, their actions and a data representation of the context in which the action occurred are stored. The collection of all actions and their context forms the user's behaviour history. This is the manner in which the user behaviour history is collected in a PSS, identical to the DAIDALOS monitoring technique. The history of the user's behaviour is used by the Learning subsystem to learn user preferences. Another responsibility of the User Interaction Monitor is to post an event to the Event Management system to inform other components that the user has performed an action. The Personalisation subsystem relies on this event to trigger a behaviour learning cycle.

The Feedback Management component provides a list of graphical user interface templates that allow components to prompt the user to provide feedback on an action that is about to be performed. The Proactivity subsystem and the Privacy Subsystem are examples of subsystems that make use of this component when it is not possible to decide what action to take based on the existing user preferences and privacy preferences respectively. The user's response is communicated back to the appropriate component which uses this information to update its data model. This way, the PSS is able to improve itself, by adapting its behaviour based on what the user wants in different situations.

Learning Management

The Learning Management subsystem includes implementations of several learning algorithms suited for learning different behaviours. As in DAIDALOS, the Mining Algorithms component includes an implementation of the C45 learning algorithm [124] that is used to learn user preferences. However, the PSS Framework architecture has a fundamental difference in the way learning is performed. In DAIDALOS, learning was performed on the user's entire user behaviour history. This history can grow immensely after some time has passed and the system will accumulate a lot of information about the

user's behaviour. The C45 Learning component exposes an interface that allows other components to trigger a learning cycle. A learning cycle can be limited to a specific period of time and to a specific type of user action. This allows the component that requests a learning cycle to ask for a specific preference to be learnt based on actions that occurred in a specific timeframe. This is efficient for two main reasons; a) the amount of historical data to be processed is considerably less than the entire behaviour history and therefore the learning process requires less time to produce a result and b) the user behaviour may change over time, causing older behaviour to become obsolete and hence the preference models must adapt quickly to give priority to recent actions over older ones. The Preference Merging component is an example of a component that makes use of the C45 Learning component to trigger a learning cycle. The C45 Learning component retrieves the user behaviour history from the Context History database and attempts to find common patterns of behaviour. The objective is to discover the same action occurring in the same conditions many times. The output of the C45 Learning algorithm is a preference or a list of preferences that are communicated back to the component that triggered the learning cycle.

Service Run-Time Environment & Overlay Network Management

The Event Management component mentioned in the previous section is part of the Service Run-Time Environment subsystem. It provides an interface for components to publish events to the system and for listener components to subscribe for events. There are a number of events that the Personalisation and Privacy subsystems need to be notified about such as when the status of a service changes, for example starting, stopping, installing, uninstalling a service, or when a user action is received by a service.

The Service Messaging component is part of the Overlay Network Management subsystem (ONM). It provides the underlying service messaging infrastructure for intra and inter PSS communication. This component is used by the Personalisation subsystem to relay messages between components in the same PSS such as personalising services located on remote devices and by the Privacy subsystem to relay messages between the components in different PSSs in order to perform the Privacy Policy Negotiation process. In general, it is the component used to send messages to components residing on a different device.

Service Management

The Service Management Subsystem includes components that are responsible for advertising services to other PSSs, discovering other PSSs in the vicinity and services from other PSSs in the vicinity and managing the active service sessions for atomic and composite services. The Session Management component publishes service related events such as when services start, stop, and recompose. The Service Discovery component publishes events when a new PSS is discovered and new services are discovered from PSSs in the vicinity. The Personalisation and Privacy Subsystems need to be notified of these events when they occur.

PERSIST Digital Personal Identifiers

The design of the Identity Management System and the Identity model has an impact on all aspects of the PERSIST framework. The Identity Model used in PERSIST is a simpler version of the DAIDALOS Virtual Identity Model. Unlike DAIDALOS, The PERSIST Identity Model does not have as a requirement that an identity has to be issued by a telecommunications operator. The Identity Management System of the PSS framework is solely responsible for issuing and managing the user's identities. These are called Digital Personal Identifiers (DPIs) and are used in order to represent the PSS to other PSSs. Other types of identities such as the ones issued by a government, an institution or a workplace are not in the same level as the PSS Digital Personal Identifiers. These other identities are easily handled inside the PSS privacy protection framework by treating them as personal information stored as context attributes in the user's profile.

As in the case of the DAIDALOS VID Framework, the PSS Privacy Protection Framework allows for multiple identities to be active simultaneously. This means that the PSS can use several services at the same time using different identities to represent itself to the PSSs providing these services. Some of the identities of the PSS are created by default and have specific functions. These are the private DPI, the public DPI and the local DPI. Consumer DPIs are created on the fly with the purpose of consuming services from other PSSs.

Private Digital Personal Identifier: The private DPI is used internally in the PSS to enable profile data (such as context data) to have a default identifier and is never exposed outside the PSS platform. This means that every item of data inside the context management

system (as all data are stored in the Context Management database) is accessible using the private DPI. The private DPI is never used to consume or provide any services locally or to another PSS. There is only one private DPI for each user PSS.

Public Digital Personal Identifier: The public DPI is used to advertise the PSS on the network. The public DPI has very little amount of data attached to it. The user can decide what they wish to make public with their public DPI. The PERSIST Overlay Network Management System that manages PSS network connectivity, uses the public DPI to advertise PSSs services to other PSSs. The Identity Management component supports changing the public DPI of the PSS according to the user's context and preferences, therefore a PSS can have as many public DPIs as the user wishes to have but only one can be active at a particular point in time. The initial design of the Identity Management component included allowing multiple public DPIs to be active simultaneously but the requirement was not met at the Overlay Network Management layer due to limitations to the underlying JXTA platform [159]. However, according to the specification of JXTA, it is possible to alter the current JXTA implementation for the purposes of allowing multiple PSS advertisements to be sent to the network each one with a different public DPI embedded in them. This would require multiple instances of the JXTA platform to be running at the same time each one serving exactly one public DPI. The benefit of having multiple public DPIs active simultaneously is to be able to offer different services with each public DPI and the client PSSs interacting with them as different PSSs.

Local Digital Personal Identifier: The Local DPI of a PSS is used internally to use services that are installed and running on the user's PSS on behalf of same PSS. The risk for data disclosure in this case is minimal to non-existent as services provided by the user's PSS are not exchanging information with other PSSs and any data operation stays local to the PSS.

Consumer Digital Personal Identifier: Consumer DPIs are used to consume services provided by other PSSs. Consumer DPIs as a concept, partition the user's profile data into smaller separate profiles in order to limit the amount of information that can be disclosed to other PSSs. As described in section 3.2.2, every context attribute, entity and association has a context identifier that embeds the private DPI in them. Before disclosing a piece of context data such as a context attribute, the DPI in their context identifier is replaced by the consumer DPI selected to use with that service. The original context identifier stays

intact inside the context database but the context identifier given to the provider offering the service is "disguised".

The Identity Management (IdM) System manages all the identities and the mapping of context data to these identities. The IdM also takes part in the Identity Selection phase which succeeds the negotiation phase and is responsible for finding identities that match a service's requirements.

3.2.3 Lessons Learnt

Privacy Preference Format

The use of Privacy Policy Negotiation preferences for performing access control as well as driving the negotiation process proved very inefficient. Privacy Policy Negotiation preferences do not need to be context dependent as data disclosure does not occur during negotiation. Therefore, an evaluation of a PPN preference during the negotiation could yield a different outcome than during a request for data during the usage of the service. Therefore, it would be more efficient to provide the user with two different types of preferences; one type to help with privacy policy negotiation that is not context dependent but does depend on the trustworthiness of the service provider and the conditions included in the privacy policy of the service they are offering and one type to control the access permissions for service providers which should be context dependent, trust dependent and also take into account the negotiated conditions.

Learning privacy preferences

During the monitoring and learning phase, a privacy action was accompanied by the context information describing the user's situation during the privacy policy negotiation process. The privacy action embedded information such as the privacy statements (obligations) that were contained in the service's privacy policy. This meant that almost all actions were distinct from each other. Thus, the window for learning privacy preferences for privacy policy negotiation is very narrow with service providers having very different privacy policies according to their requirements and the context of the user changing very frequently. The C45 learning algorithm tries to find common actions occurring under the same circumstances; it proved very inefficient in learning privacy preferences that could be used both for future privacy policy negotiations as well as performing access control. Therefore, a different learning method needs to be explored

that is a lot more flexible and allows privacy preferences to be learnt faster and in a more generic fashion so they can be reused in new negotiations.

Negotiation from the viewpoint of the service provider

In the DAIDALOS architecture, the service provider is an entity such as an organisation or company. During the privacy policy negotiation, both parties exchange statements that have to be verified by the user. In the case, that preferences exist in the system for both client and provider, the negotiation can be performed automatically. In the case that the service provider is presented with a statement that the system cannot process due to lack of rules for this particular statement, manual intervention is required to respond to this statement. Given that the service provider is non-human, the system must be able to cope with this situation. In PERSIST, this issue was less problematic as PSSs were run by individuals acting as service providers but nonetheless, even a user acting as a service provider cannot always be available to respond to user requests. Therefore, an automated solution is still needed.

Informed Consent

A number of questions were raised during the design of both DAIDALOS and PERSIST privacy protection systems. By law, the user has to agree explicitly with the terms and conditions of a service and give informed consent. The first question is whether there is a clear benefit for using a fully automated system with such complicated policy generation algorithms to do privacy policy negotiation. The second question is what are the tasks that should be automated and what are the tasks that the user should always be involved in. Further, giving the user the option to decide which parts to automate and which parts they always want to be prompted for must also be explored.

3.3 *ICT-SOCIETIES*

SOCIETIES (Self Orchestrating Community ambiEnT IntelligEnce Spaces) is an EU FP7 ICT Integrated Project with participation from both industry and academia that started in October 2010 and ended in April 2014. The vision of SOCIETIES is to develop a complete, integrated Cooperating Smart Space (CSS), which extends pervasive systems beyond the individual to dynamic communities of users [160]. The functionality of SOCIETIES is based on the three concepts; Discover, Connect, Organise.

Discover: The SOCIETIES platform provides functionality to discover resources, services, users and communities that might be relevant to the user based on information from user profile, past behaviour, current context, future behaviour predictions and social network information such as interests, social interactions and relationships.

Connect: The SOCIETIES platform provides functionality to connect users who share similar interests, have common goals, or are collocated by suggesting communities that they can join, create or manage.

Organise: The SOCIETIES platform provides functionality to adapt the user's digital and physical environment using user and community information captured in the system. Resources and services provided to communities of users are adapted accordingly, to the communities' content and context.

The objective of SOCIETIES is to bridge the gap between pervasive and social computing using Pervasive Communities where users can communicate using both the physical and digital environment. SOCIETIES builds upon earlier research into Personal Smart Spaces (PSS) conducted during the PERSIST project to deliver to each individual user a Cooperating Smart Space (CSSs). Using their own CSS, users take advantage of the SOCIETIES functionality such as dynamic community orchestration, proactive personalisation and strong privacy protection technologies to aid them in achieving their daily tasks efficiently. Community Interaction Spaces (CISs) are pervasive communities in which users can connect with other users through their respective CSSs, communicate, interact, share services, devices, resources and information to support them as individual users as well as a community.

One fundamental difference between the PERSIST PSS and SOCIETIES CSS is that a PSS can function in an ad hoc manner, without access to infrastructure, while a CSS needs connectivity to cloud services for some of its functionality.

3.3.1 SOCIETIES concepts

Cooperating Smart Spaces (CSSs)

A CSS represents a single participant (user or organisation), and includes their information, and services within a distributed collection of CSS Nodes where a node is a

device that runs a version of the SOCIETIES platform. It provides both a pervasive capability and a social networking capability in an integrated form. A CSS can interact, communicate, or share services, resources and information directly with other CSSs, both in a peer to peer fashion as well as within a Community Interaction Space (CIS).

Community Interaction Spaces (CISs)

A CIS is a representation of a pervasive community administered by a single CSS. CISs can be defined by a logical as well as a physical space in which CSSs can share resources, services and information and organise community tasks in a dynamic and context-aware fashion. CIS components such as community context, community personalisation and community orchestration enhance pervasive communities by assimilating context information that is derived by the collective behaviours and information that stems from CSS to CSS interactions within the CIS. Figure 12 shows the architecture of the SOCIETIES platform.

3.3.2 SOCIETIES Architecture

The SOCIETIES core services are distributed within four vertical layers depending on the type of user they are serving. The lower two layers – Node and Participant Components - are operating on behalf of individual users, providing communication functionality between the devices managed by the same CSS, service management, device management, user personalisation, proactivity, behaviour learning, intelligent privacy protection, intelligent community orchestration and social networking connectivity to the CSS. The Community Components operate on behalf of a CIS providing functionality such as community context management, community personalisation and community behaviour learning. Finally, the top layer Umbrella Components operate independently of CSSs and CISs, employing a bird's eye view on the CSSs and CISs available inside the SOCIETIES system and provide community recommendations, CSS and CIS directory services, Identity Administration functionality and finally a Marketplace similar to Google Play Store [161] and Apple's App Store [162] that hosts SOCIETIES enabled applications and services.

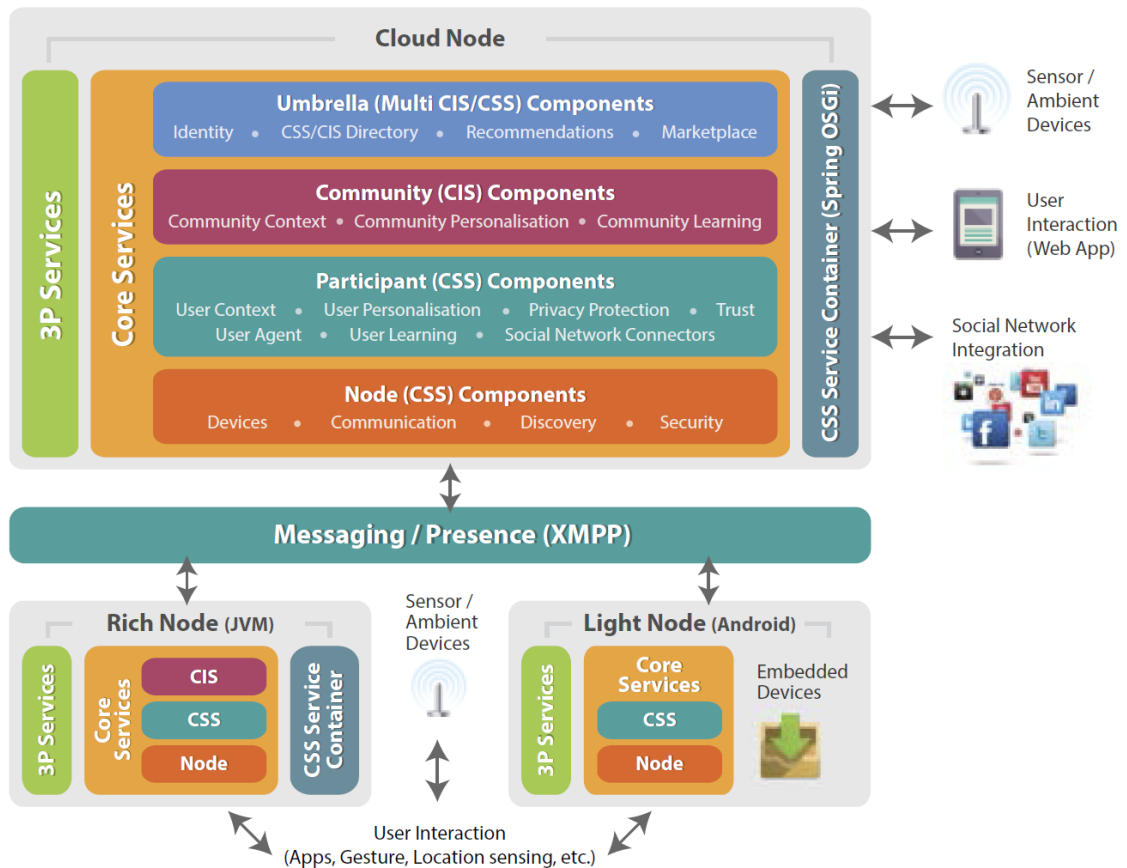


Figure 12. *SOCIETIES Architecture*

Personalisation in SOCIETIES

Both on the individual and the community level, the SOCIETIES Personalisation system supports complete dynamic management of preferences and behaviour models, proactive personalisation of services and the environment of the user. The Personalisation system plays a significant role both in the pervasive and the community aspect of the SOCIETIES platform. By monitoring user and community behaviour and applying appropriate learning algorithms, preferences and behaviour models are built automatically offering a seamless personalisation experience to the user as such mechanisms can relieve the user from the burden of manual service configuration. Essentially, the SOCIETIES Personalisation system provides mechanisms that can drive a system in a proactive manner based on the user's wishes. The SOCIETIES Personalisation System employs two different user preference models; one based on IF-THEN-ELSE rules and one based on preference based dynamic incremental neural network (DIANNE [163]) and two different user intent models; one context aware user intent model based on the Hidden Markov Model [164] and one user intent model based on Conditional Random Fields (CRIST) [165]. SOCIETIES utilised multiple different approaches to personalisation aiming to

study their strengths and weaknesses of each approach in the same situation and examine which algorithm is best suited for a specific situation.

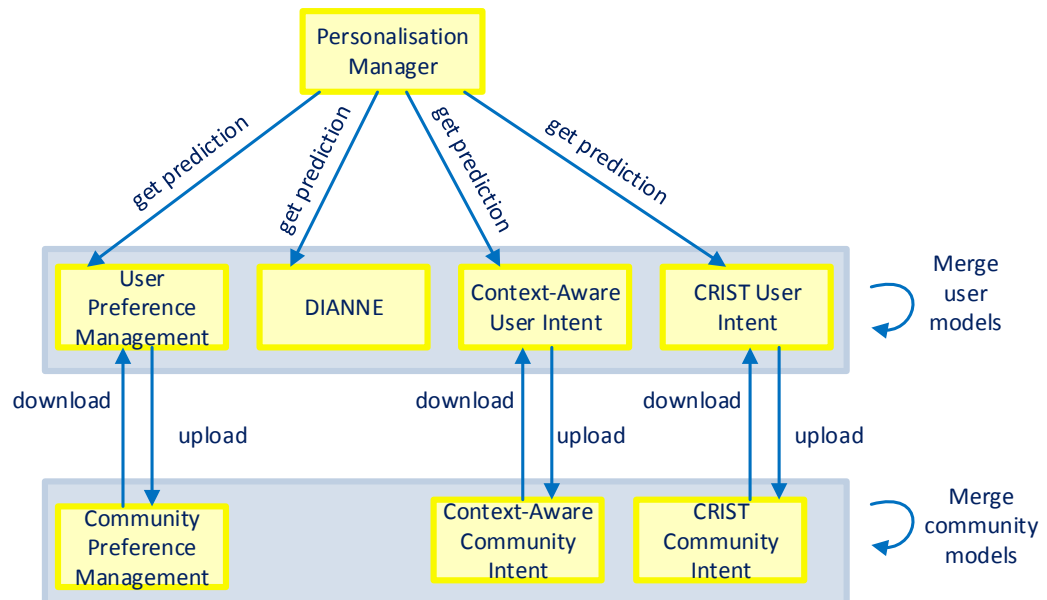


Figure 13. *SOCIETIES Personalisation Interactions*

The Personalisation Manager component acts as a coordinator between the different sources of personalisation predictions produced by the four personalisation algorithms. The four personalisation algorithms indicate to the Personalisation Manager which context attributes would be affecting their predictions so it can monitor any changes to these context attributes. When either a context event or a behaviour action is received, the Personalisation Manager requests an evaluation of the behaviour models from each of the four personalisation algorithms based on the event received. The Personalisation Manager then inspects the preference predictions and intent predictions separately. Conflict resolution – if needed – is performed between the preference based predictions and then the intent based predictions and delivers to the User Agent one list of preference based predictions and one list of intent based predictions in the form of Action objects. Responsibility for performing conflict resolution – if needed – between these lists of actions lies with the User Agent component.

Depending on the confidence level for the actions received, the User Agent will notify the user of an impending personalisation action in three different forms; explicit, implicit and multiple choice feedback requests:

- **Implicit feedback:** This notification dialog box appears for ten seconds to notify the user that a personalisation action will be performed and allows the user to abort it within 10 seconds. This form of feedback is used when the confidence level of the action to be implemented is very high. If the user does not respond within 10 seconds, the User Agent automatically implements the action.
- **Explicit feedback:** This notification dialog box requires the user's explicit feedback so the action will not be performed until the user explicitly indicates if they wish the action to be implemented or not. This form of feedback is used when the confidence level is high enough to notify the user but not as high as to implement it automatically.
- **Multiple choice feedback:** This notification dialog box asks the user to select from more than one conflicting actions to be performed. The user has the option to select none of the actions if they wish to do so. This form of feedback is used when the conflict resolution algorithm is unable to make a clear decision as to what action to perform and requires the user's input.

Depending on the user's feedback, the User Agent performs the action or aborts the personalisation. But most importantly, it informs the Personalisation Manager of the user's input in the form of a Feedback Event which specifies which action was implemented or aborted, and whether the action was or was not successfully implemented including the reason.

The Personalisation Manager holds a confidence level for each of the personalisation algorithms it employs which is updated according to the success or failure of the actions they predict. Upon receiving the feedback from the User Agent, the Personalisation Manager will update the confidence levels accordingly and will also notify each of the personalisation algorithms of the user's feedback so that each of them can update their behaviour models accordingly.

Privacy Protection in SOCIETIES

The Privacy and Trust system for the SOCIETIES platform was designed to provide privacy protection both for the Cooperating Smart Space (CSS) as well as the Community Interaction Space (CIS). The aim of the SOCIETIES Privacy and Trust System is to protect the privacy of the user, aid the users in making informed decisions about

disclosing data to other CSSs, services and CISs as well as sharing resources and services within communities and finally protect the resources, information and shared services of a community.

The SOCIETIES Privacy and Trust system is a policy based privacy protection framework utilising four distinct technologies; privacy policy negotiation, access control, data obfuscation and trust management.

Privacy Policy Negotiation. The system provides a non-automated privacy policy negotiation mechanism that allows users to modify the terms and conditions of data disclosure using an online form when they install a service or join a CIS.

Access Control. Permissions for allowing or denying access to specific data are managed by an Access Control component. Permissions are set during a service installation or during the process of joining a CIS. A permission defines whether a specific subject (or requestor) can access a specific data item and what kind of operation it can perform on the data (read, write, create, delete).

Data Obfuscation. Data obfuscation is the process of modifying data to make them less specific. The most common example of data obfuscation is location obfuscation. Not all of the data can be obfuscated and depending on the type of data, different obfuscation algorithms can be applied. Data types such as name, age, date of birth, temperature, employment, location, ethnicity etc. can all be obfuscated as long as the appropriate obfuscation technique is available for that data type and appropriate obfuscation rules exist to perform it.

Trust Management. The SOCIETIES Trust Management system is designed to evaluate the trust of a requestor as a CSS and a CIS. The system supports the evaluation of direct and indirect trust. Direct trust is the evaluation of the trustworthiness of an entity based on interactions of the user with that entity whereas indirect trust is based on trust evaluations of an entity performed by other trusted CSSs.

Personalising privacy in SOCIETIES

Privacy Policy Negotiation, Access Control and Data Obfuscation take advantage of personalisation, specifically custom privacy preferences tailored to each individual

technology's requirements. The aim of adapting the user's privacy using personalisation techniques is to help the user to make better decisions about disclosing data to third parties as well as unburden them from having to respond to frequent prompts from the system about what action to take in each case.

In the case of privacy policy negotiation, the system employs customised privacy preferences to suggest to the user how to handle the negotiation. These preferences are created from decisions the user has made during previous negotiations with the same requestor. Personalisation is applied to aid the user in negotiating with the services or CIS administrators but not to complete the negotiation on behalf of the user.

In the case of access control, the system uses customised access control preferences to perform access control on behalf of the user. As in the previous case, access control preferences are created based on the user's responses when asked to disclose data or after the negotiation process completes. For all the static data (such as name, age, ethnicity etc. which, unlike context data, do not change depending on the environment) requested in the privacy policy, access control preferences are created that indicate that these data can be disclosed if the negotiated conditions are met. If the type of data is contextual that describes the user's environment, access control preferences are created by asking the user if they want to disclose the data under the current circumstances.

3.3.3 Lessons Learnt

Fine granularity in Privacy Policy Negotiation Preferences

In both PERSIST and SOCIETIES, the user's input to the privacy policy negotiation process was translated into privacy policy negotiation preferences. Each preference indicated a decision regarding the terms and conditions for allowing a specific service provided by a specific service provider to access a specific data item. Defining these preferences with such fine granularity resulted in their inability to be reused in future negotiations with services and service providers with whom the user has not interacted with. Therefore, extra privacy policy negotiation preferences can be created that define courses of action without defining specifically the service and service provider but instead, use their trust level as a parameter to implementing such decisions. In this manner, in future negotiations, the system can recommend decisions depending on the trustworthiness of the service providers.

Lack of Multiple Identities

Initially, the SOCIETIES architecture was designed to allow users the flexibility of using multiple identities to represent their Cooperating Smart Space. However, the choice of using XMPP as the communication protocol and subsequently further choices in the integration of XMPP and the SOCIETIES platform proved too time consuming to implement this feature in an appropriate manner. Hence, the final architecture of SOCIETIES was based on a single identity with which the Cooperating Smart Space was using to represent itself. As a result, this demeaned the quality of the protection of privacy that the SOCIETIES platform offered due to the fact that the user profile could not be segregated since all data disclosed by the CSS were linked to that single identity.

3.4 *EU Projects Research in Relation to Thesis Research Objectives*

Each of the three EU projects has influenced the research objectives of this thesis and had a profound impact on the design of the PersonISM system. More specifically:

Enabling users to remain in control of their data.

The design of the DAIDALOS platform was carried out in a top-down approach, influenced mainly by the telecommunications industry. The research was heavily business-oriented which impacted the design of several components. The centralised architecture was never questioned or debated. The entire Pervasive Services Platform was designed to be provided by telecom operators which meant that all the privacy critical components that related to the storage, processing and disclosing of user data were not under the control of the user. Despite the fact that the Virtual Identity model merits special attention in the research on privacy, its design impeded the personalisation and learning processes. In contrast, the PERSIST and SOCIETIES projects used a combination of a top-down approach to steer the research directions of the projects and a bottom up approach for collecting requirements for each individual functionality of the Personal Smart Space and Cooperative Smart Space concepts. Privacy management, context management, personalisation and learning components were designed from the start as components that should be tightly coupled to provide to the user a pervasive experience as well as protect their privacy.

Assisting users in maintaining their privacy.

The difficulty in maintaining one's multiple identities became apparent during the DAIDALOS project. It was difficult to imagine how a user could manage the associations of several data to identities and the use of each identity for a specific purpose without proper assistance from the system. Due to restrictions posed by the Virtual Identity architecture, the personalisation components had no access to the right information in order to be able to perform personalised identity selection. Hence, a personalisation component had to be created to be embedded inside the Identity Management system in order to have access to all the information it required. By contrast, from the start of the design phase of the Personal Smart Space platform, the personalisation and learning components were designed to help users maintain their privacy. The author was very heavily involved in designing and implementing all aspects of the privacy protecting framework that availed itself from personalisation and learning mechanisms [166].

3.5 *Summary*

This chapter provided an overview of three pervasive platforms developed under the EU projects DAIDALOS, PERSIST and SOCIETIES respectively. All three pervasive platforms placed a lot of emphasis on utilising dynamic personalisation and strong privacy protection techniques.

The DAIDALOS Pervasive Services Platform provides context awareness and dynamic personalisation of services in a heterogeneous networking environment. Using context dependent user preferences, the DAIDALOS PSP platform focused on the personalisation of DAIDALOS enabled third party services and several system functionalities such as service discovery and composition, network interface selection, communication redirection and identity selection. Privacy protection technologies were heavily dependent on the underlying Virtual Identity model which defined much of the implementation of the DAIDALOS PSP platform components.

The PERSIST Personal Smart Space platform takes a different approach based on a peer to peer platform with the goal of bridging isolated islands of pervasiveness separated by a lack of connectivity and pervasive infrastructure. Context and personalisation played an important role in PERSIST by offering personalisation and proactive automation of functionality; internally in the Personal Smart Space, in the areas of service management

and resource sharing as well as in PSS to PSS interactions in the areas of resource and service discovery, selection and adaptation, privacy negotiation, identity selection and access control. The peer to peer nature of the identity model utilised in PERSIST gave greater flexibility in designing the PSS privacy protection features and allowed the user ultimate control over their personal data in terms of storage and disclosure to third parties.

The SOCIETIES platform combined the concept of a personal smart space with the presence of infrastructure. It implemented concrete functionality for interactions of smart spaces within community smart spaces to promote native social networking inside the SOCIETIES platform and integrations with existing social networking sites. SOCIETIES provides a comprehensive approach to personalisation by leveraging four diverse types of personalisation; if-then-else preferences, dynamic incremental associative neural network, context aware user intent and conditional random fields based user intent. Each method has strengths and weaknesses and the purpose for using these in a combined approach was to use the strength of one approach to offset the weakness of another. The SOCIETIES privacy framework was designed to offer users appropriate safeguards against unnecessary and unwanted data disclosure as well as helping them in configuring their privacy as they see fit.

4 Personalisation in a Pervasive System

The purpose of personalisation in a pervasive system is to adapt the environment of the user according to its user's wishes. The user's environment includes the pervasive platform itself, the services running on that platform, the interactions with other entities, the available resources and data. To accomplish this task, a personalisation system needs a way to represent the wishes of the user that is both machine and human readable to enable the user to make changes if they wish. As the user performs their daily routine, their environment (or more specifically their context) changes frequently. As their context changes, services and resources need to be reconfigured to adapt to the new environment as per the user's wishes. This chapter presents an approach to personalisation in pervasive systems based on the author's work during the participation of the three EU projects DAIDALOS, PERSIST and SOCIETIES.

A pervasive system is characterised by the presence of devices, services, sensors, resources and data in the environment of the user which can be monitored, configured and used to aid the user in their everyday tasks. The design for a personalisation system within a pervasive system should satisfy a number of basic requirements that stem from the nature of pervasive computing as well as the prime requirement that a pervasive system serves its users.

Environment monitoring. The personalisation system needs information about the current context of the user and their surroundings to be able to make good decisions about adapting the services and resources available to the user at any time.

Proactive adaptation. Appropriate mechanisms should be employed to automatically reconfigure services, resources and the system itself when changes in the context of the user are detected.

User behaviour monitoring and learning. Learning user preferences by monitoring the users' behaviour and mining behavioural history is required to alleviate the user from the burden of manually creating user preferences.

Dynamic self-improvement. The behaviour models employed must be flexible because they need to be constantly updated to reflect changes in the users' behavioural patterns.

Self-improvement mechanisms must combine behavioural history information as well as direct feedback from the user.

Manual configuration. The benefit of allowing the user to manually edit their user preferences is twofold. First, it gives the user the ultimate control over how the system should behave on their behalf. The user can see how the system will behave under different circumstances and make changes accordingly. Second, as 100% accuracy in any prediction algorithm cannot be guaranteed, manual configuration is the only failsafe way to correct a false or inaccurate prediction.

4.1 *User Preferences*

User Preferences define the wishes of the user in machine readable format. DAIDALOS, PERSIST and SOCIETIES use a user preference model based on nested if-then-else statements. The User Preference model is a generic model that allows any type of preference to be represented in the same way. Therefore using the same data structures and algorithms, the system is able to apply all the different types of personalisation needed in each platform. The most important factor that led to adopting nested IF-THEN-ELSE rules for user preferences was their simplicity and ability to be represented textually so they can be manipulated using a Graphical User Interface (GUI). This is in contrast to other solutions to personalisation such as Neural or Bayesian network models that are nearly impossible to represent textually or allow the user to edit.

4.1.1 **Context-Dependent Nested IF-THEN-ELSE Rules**

A Preference contains a set of conditions and outcomes. A Preference outcome defines an action to be implemented in the system. This can be either setting a parameter for example setting "volume = 0" or calling a method in a service for example "method = 'turn_on_lights' ". It is in the discretion of the service developer how to represent a personalisable parameter or action in the system since only the service is able to interpret the semantics of the action. The set of conditions define the context in which the outcome must be implemented in the system. A Preference Condition is a statement that can be evaluated. Conditions can be context conditions such as the user's current location (e.g. 'location == home') or activity (e.g. 'activity == jogging'), or service conditions such as the state of a service (running, not running, paused or any state allowed by the service

model employed). Here is an example of a user preference in the form of a nested IF-THEN-ELSE rule:

```

IF location == home AND activity == reading           set of conditions
    THEN temperature=23                               outcome
ELSE IF location == home AND activity == sleeping    set of conditions
    THEN temperature = 20                             outcome
FI
ELSE    temperature=19                               outcome
FI

```

The Backus-Naur form of the user preference model is shown below.

```

<preference> ::= IF <condition> THEN <preference> FI
              | IF <condition> THEN <preference> ELSE <preference> FI
              | <outcome>

<condition> ::= <simple condition> <logOp> <condition>
              | (<condition>)
              | NOT (<condition>)
              | <simple condition>

<logOp> ::= AND | OR

<simple condition> ::= <attribute> <relOp> <value>
                  | <predicate method call>

<relOp> ::= <|<=>|>|=|<>

<outcome> ::= <outcome_name> = <outcome method_call>
            | <parameter name> = <value>

<attribute> ::= <string>

<value> ::= <string>|<integer>|<real>

<outcome method_call> ::= <name>(<parameters>)

<predicate method_call> ::= <name>(<parameters>)

<parameters> ::= <parameters>,<parameter>
              | <parameter>

<outcome name> ::= <string>

<parameter> ::= <value>

<parameter name> ::= <string>

```

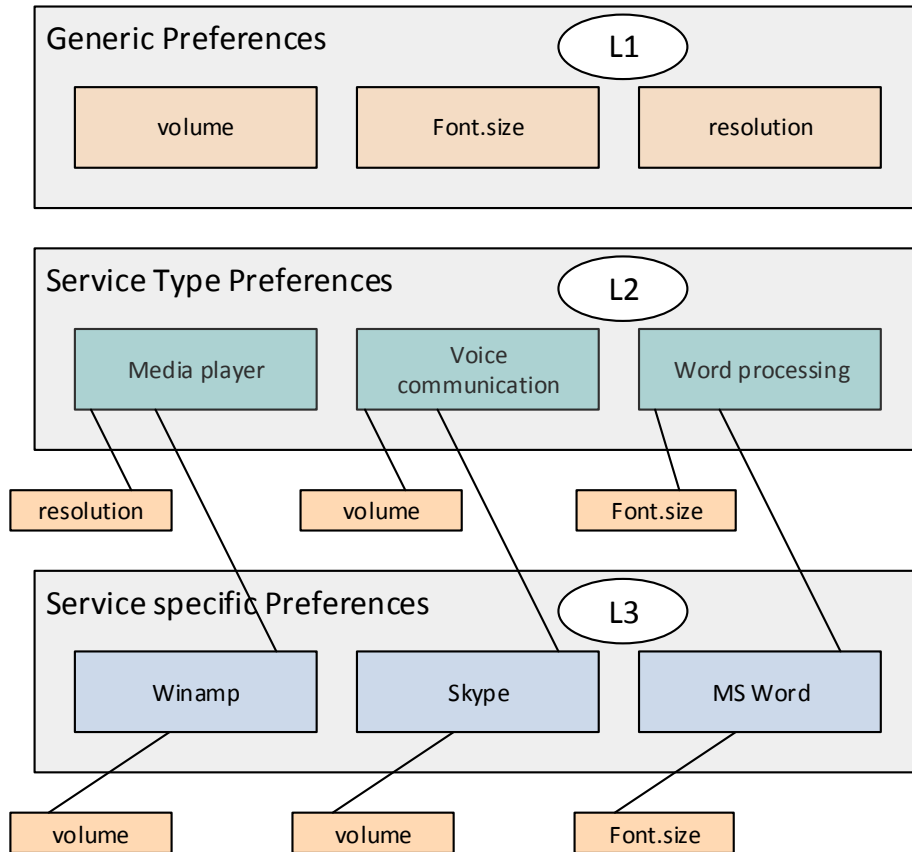
```

<integer> ::= <digit> { <digit> }
<string> ::= <char> { <char> } " "
<char> ::= <letter> | <digit>
<letter> ::= A | B | C | E | G | H | J | K | L | M | N | P | R | S | T
| V | W | X | Y | Z | a | b | c | d | e | f | g | h | i | j | k | l |
m | n | o | p | q | r | s | t | u | v | w | x | y | z
<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

```

Figure 14. *BNF notation of the context-aware user preference model.*

A User Preference defines the wishes of the user for a specific personalisable parameter. For example, the preference for “volume” is defined in a separate preference to that for “genre”. User Preferences are indexed by the type of service and/or an identifier for a specific service to which they refer. Service type preferences are generic and used to personalise new services based on their service type. For example, assuming that there are stored preferences for service type “multimedia” and the user installs a new VoIP application of this type on their device, the application can be personalised using the generic preferences relevant to service type “multimedia”. By monitoring the use of the VoIP application, the learning algorithms can subsequently learn specific preferences for that particular application and store them. The next time the VoIP application is launched, the VoIP application will be personalised using the newly learnt VoIP service specific preferences. User preferences can also be service agnostic in which case, they are not related to a specific service or service type. These are used very rarely, such as when personalising a newly installed service whose service type has not been encountered before and for which no generic service type preferences exist in the system. The hierarchy of these different preference types is illustrated in Figure 15. The generic preferences are stored at the top in level 1. The preferences specific to a service type are stored in level 2 and can be retrieved by providing the service type as a string. The service specific preferences are stored in level 3 and are retrieved using the specific service identifier and its service type.

Figure 15. *Preference Hierarchy*

4.1.2 User Preference model implementations in DAIDALOS, PERSIST and SOCIETIES

A preference model employed in a pervasive and ubiquitous system for the purposes of personalisation is required to represent any type of condition along with an outcome that is to be effectively implemented by the system. As mentioned in the previous sections, DAIDALOS, PERSIST and SOCIETIES used nested IF-THEN-ELSE rules to represent user preferences. A preference has two parts, a conditional expression (a Preference Condition) and an outcome (a Preference Outcome) that expresses an action to be implemented by the system. The condition part of a preference can include testing of attributes of the user's context such as 'location' or 'activity'.

The DAIDALOS preference model differed from the PERSIST preference model in its implementation. The former is implemented using nested Java objects expressing three different constructs. The first construct allowed context-independent personalisation, represented by a preference outcome only; the second construct provided a simple IF-THEN construct with a conditional expression and an outcome while the third one

implemented a full IF-THEN-ELSE form where further preferences could be nested under the THEN or ELSE parts of the preference adhering to the BNF shown in section 4.1.1. While this approach made it easy to visually present these rules to the user, it was a complicated and inefficient format to use for merging existing preferences with new preferences acquired using user behaviour learning techniques. This was not an issue in DAIDALOS as the Learning algorithm processed the entire history of the user behaviour and there was no need to perform preference merging. However, in PERSIST and SOCIETIES, this is a very crucial requirement for the user preference model as the learning is performed on the user's recent history only and therefore it is necessary to be able to merge two models in an easy and efficient manner. Another reason for not replacing the old model with the newly learnt model and instead merging them, is to preserve any modifications the user has manually made to the existing preference. The approach adopted in PERSIST, and subsequently in SOCIETIES, to satisfy this requirement, uses a tree data structure to represent a preference object whose branches correspond to conditional expressions and the leaves of the tree are used to store the outcomes. Both processes of preference merging and preference evaluation proved to be more efficient and easier to implement with the latter user preference model implementation than the complex structure of the DAIDALOS preference model.

An example Preference Tree is shown in Figure 16. The branches of the tree represent context conditions and the leaves of the tree express an action (Preference Outcome) that has to be applied in a service or in the system. An outcome will be applicable if all the conditions on the path between the root node and the leaf node (that holds the outcome) evaluate to true. The root node of the tree can be empty, or can represent a condition (which requires that the root node has at least one child node that represents a nested condition or an outcome). If the root node is empty, then it should be treated as multiple trees joined as children of an empty root node. This means that there are two or more actions that depend on a set of conditions that are disjoint. More specifically, if \mathbf{A}_x is the set of conditions that action x depends on to be implemented and \mathbf{B}_y is the set of conditions that action y depends on to be implemented, then the intersection of sets \mathbf{A}_x and \mathbf{B}_y is empty ($\mathbf{A} \cap \mathbf{B} = \emptyset$). This implies that, at a certain point in time, two or more conflicting actions may become applicable. In order to resolve such conflicts during preference evaluation (at runtime), preference outcomes carry a confidence level that is calculated based on information from previous applications of that outcome. This confidence level

can be used to determine which outcome is the most appropriate using a conflict resolution algorithm (see section 4.3.2). In cases where the conflict resolution algorithm is unable to decide, the user can be prompted to select the outcome. This selection can also contribute to the determination of subsequent confidence levels. The information used to calculate the confidence level of an outcome includes the following attributes:

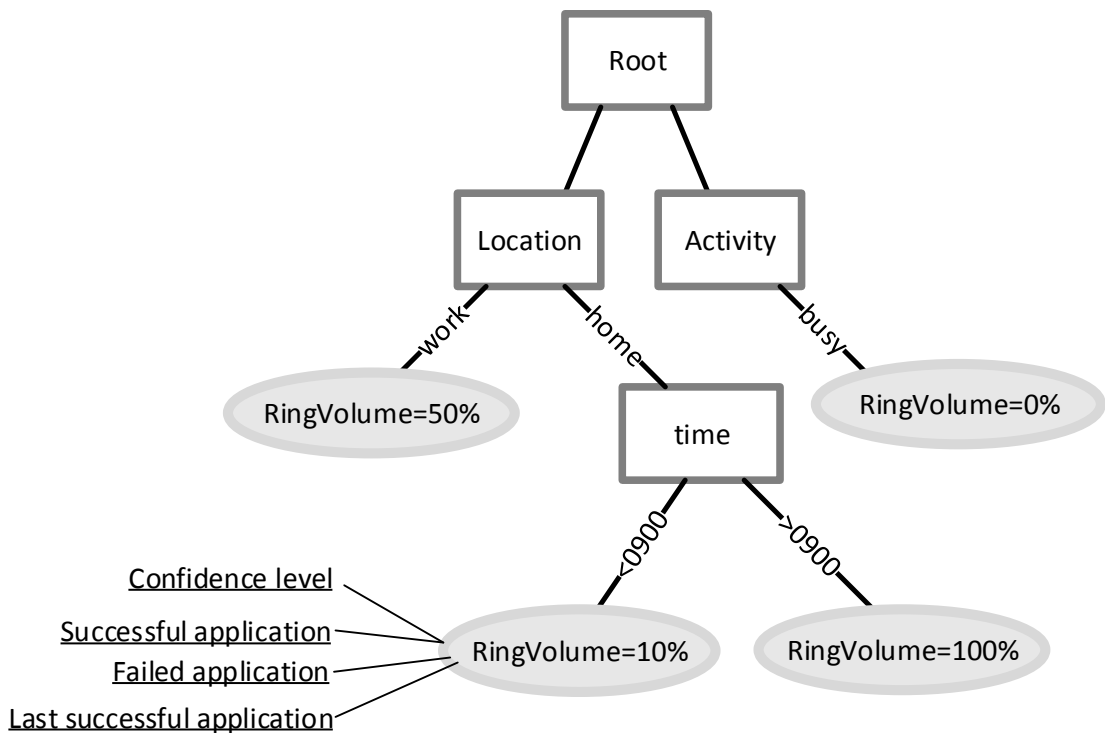


Figure 16. Preference represented in a tree format

Confidence level: The confidence level of a preference outcome expresses how strong a preference is. When the learning algorithms produce new preferences, they are merged with the existing preferences (if they exist) and their confidence level is adjusted based on what has been learnt.

Successful Application: This field states how many times this action has been applied successfully, i.e. not overridden by the user.

Failed Application: This field indicates how many times the system attempted to apply this action but was aborted by the user.

Last Successful Application: This field holds a timestamp to declare the last time the application of this outcome was successful.

The information is updated when a preference is applied and user feedback is received and the confidence level is recalculated according to the newest values. The confidence level is needed during the evaluation of preferences.

4.2 *Dynamic Personalisation*

Personalisation can be performed in a static or a dynamic manner. The term *static personalisation* refers to situations where services explicitly request user preferences to personalise themselves. This form of personalisation does not react to changes in the context of the user and thus is not adequate for a pervasive system. A pervasive system is expected to adapt the environment of the user according to the user's wishes and the availability of services and resources around them in an automated seamless manner. The term *dynamic personalisation* refers to personalising the environment of the user in a proactive manner, by reacting to changes in the context of the user (location, activity, time of day). As shown in Figure 17, the personalisation system can receive environment information in the form of events that indicate changes in the context of the user or the state of the services in the system. User preferences are evaluated against the current context of the user to infer what outcomes must be implemented to adapt the environment to serve the user's needs.

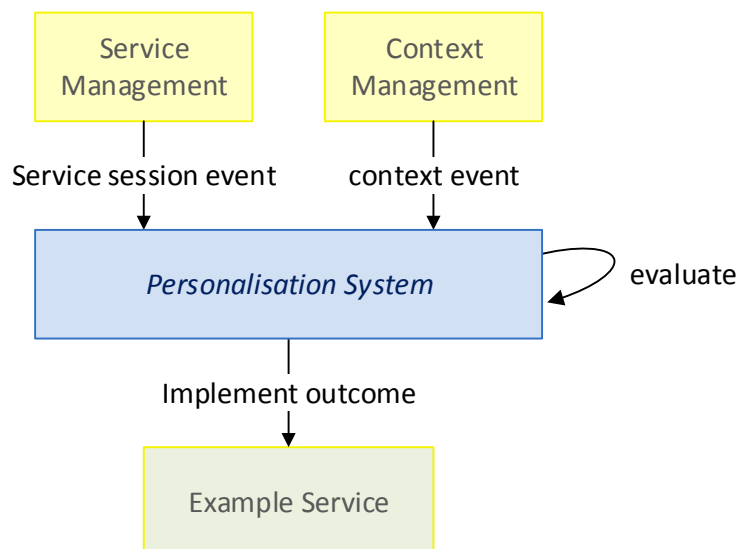


Figure 17. *Dynamic Personalisation*

4.2.1 Preference Evaluation

Preferences are evaluated when services explicitly request the outcomes of certain preferences to personalise themselves or when a change in the context of the user is detected. As described at the end of section 4.1.2 the user preference model is a nested IF-THEN-ELSE model in the form of a tree. The branches represent the conditions of the preference and the leaves represent the outcomes. A preference is evaluated by traversing the tree using depth-first pre-order traversal and evaluating the condition in each branch against the current values in the Context Management System.

Figure 18 shows the preference evaluation algorithm used in PERSIST and SOCIETIES. The preference evaluation algorithm evaluates a preference tree beginning at the root of the tree and descending, visiting each of the child nodes recursively. First, it checks to see if the current node is a branch or a leaf. If it is a leaf, then it contains an outcome and so the algorithm returns that outcome. If it is a branch, it checks to see if the branch contains a condition. If a condition is present (representing an IF statement), it is evaluated against the current context of the user and if it evaluates to true, it continues to evaluate the child nodes of the current node (representing the THEN statement). If the condition evaluates to false, any child nodes are ignored and the algorithm visits this node's siblings. If all nodes of the same level evaluate to false, the algorithm returns a null value. If a condition is absent, that means that the tree under this node is split; meaning it contains disjoint sets of conditions that may lead to multiple, and possibly conflicting, outcomes. At this point, the algorithm initialises a list of outcomes to hold the potentially conflicting outcomes that will be produced by evaluating the children of this node. For each direct child node of this node, the algorithm checks if it is a leaf node and if so, it stores this as a default outcome for this level in the tree. If the node is a branch, it recursively calls the algorithm again passing each child node of this branch as a parameter. If the algorithm returns a non-null node (an outcome node), it is added to the outcomes list. When all the nodes have been exhausted, the algorithm exits the loop and inspects the outcome list. If the outcome list is empty, the default outcome of this level is returned. If the outcome list contains only one item, that outcome is returned. If the outcome list contains two or more items, the algorithm performs conflict resolution to detect the most applicable outcome. As explained in section 4.1.2, each outcome has a confidence level that indicates how confident the system is that this outcome should be implemented under a set of conditions. Hence, the conflict resolution algorithm compares

the confidence level of each potential outcome and returns the one with the highest confidence level. The confidence level calculation algorithm is explained in section 4.3.2.

```

Outcome evaluateNode(Node node){
    if (node.isLeaf()){
        return node;
    }

    if (evaluateCondition(node.getCondition())==false){
        return null;
    }

    List outcomesList = new List();
    for (childNode : node.children()){
        outcome = evaluateNode(childNode);
        if (outcome!=null){
            outcomesList.add(outcome);
        }
    }

    switch (outcomesList.size()) {
        case 0:
            return null;
            break;
        case 1:
            return outcomes[0];
            break;
        default:
            return resolveConflicts(outcomesList);
    }
}

```

Figure 18. *Preference Evaluation Pseudocode*

The size of the preference tree depends on the number of context attributes and their corresponding values. Since each preference refers to a single parameter (i.e. volume), the type of action does not affect the size of the tree. Tree traversal is inexpensive since the algorithm will never traverse the entire tree. The tree under the branches that evaluate to false is ignored. If the tree is not split, only one branch will evaluate to true in each level of the tree and the children nodes of that branch will be visited. If the tree is split, the tree traversal will be as expensive as the number of nodes that split the tree.

4.2.2 Preference Condition Monitoring

Preference condition monitoring refers to the process of listening for changes in a) the context of the user that affect the outcome of the evaluation of the user's preferences, b) the state of the services available to the user and c) the preferences due to user intervention or preference learning. Monitoring the state of the services is needed for three reasons; first, service status information can be represented as a preference condition so it can be considered as another form of context; second, knowing which services are running reduces the number of preference evaluations needed as only the preferences of currently

running services should be evaluated; and third, knowing which services are currently running and which context affects the preferences of the currently running services reduces the number of context event registrations. Hence, a preference condition monitoring mechanism must exist to make sure that only the relevant preferences are evaluated when an event is received and only affecting context is monitored so that the system performs personalisation as efficiently as possible.

Receiving service status events

By correlating the currently running services and the context that affects their preferences, the preference condition monitoring component can request the evaluation of the preferences that are affected and can be currently applied. A “service started event” triggers loading the preferences relevant to the service that was started. The context conditions are extracted from the preferences and for each context type, the preference condition monitoring component registers for events of this context type with the Context Broker. The preferences are also evaluated against the current context and if the evaluation yields a result, the outcomes of these preferences are implemented.

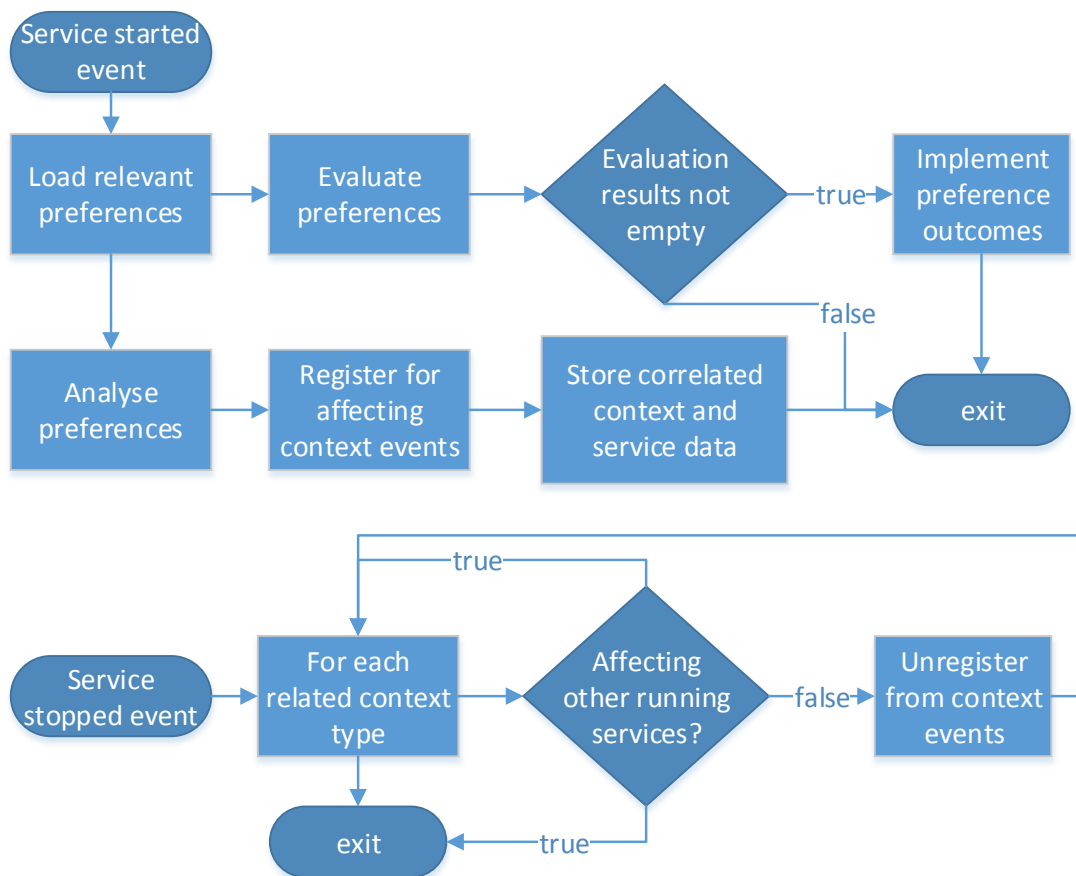


Figure 19. Service status (started – stopped) events received

If a “service stopped event” is received, the information about the preferences of the stopped service is removed and for each of the context types that affected only the preferences of this service, the preference condition monitor unregisters as a listener for events of this type. Figure 19 shows the workflow of receiving service status events.

Receiving context events

A context event indicates that something in the environment of the user has changed. After receiving a context event, the preferences affected by this type of context are retrieved and evaluated. Each outcome produced by the preference evaluation process must be implemented and an appropriate feedback event must be posted through the event management system to notify listeners if that outcome was implemented correctly or not. Figure 20 shows the workflow of receiving a context event.

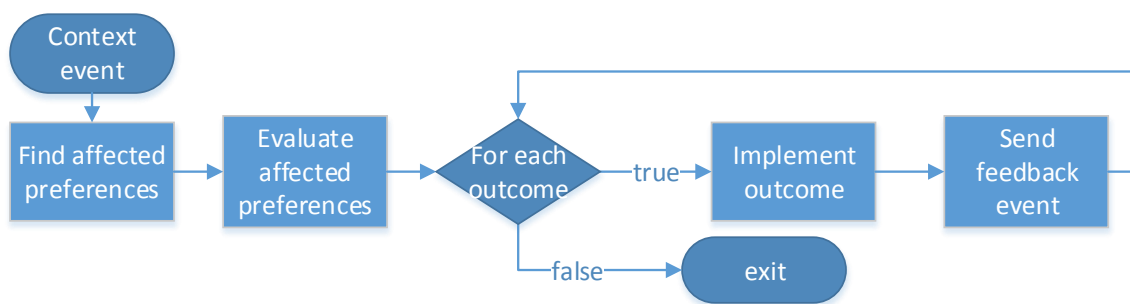
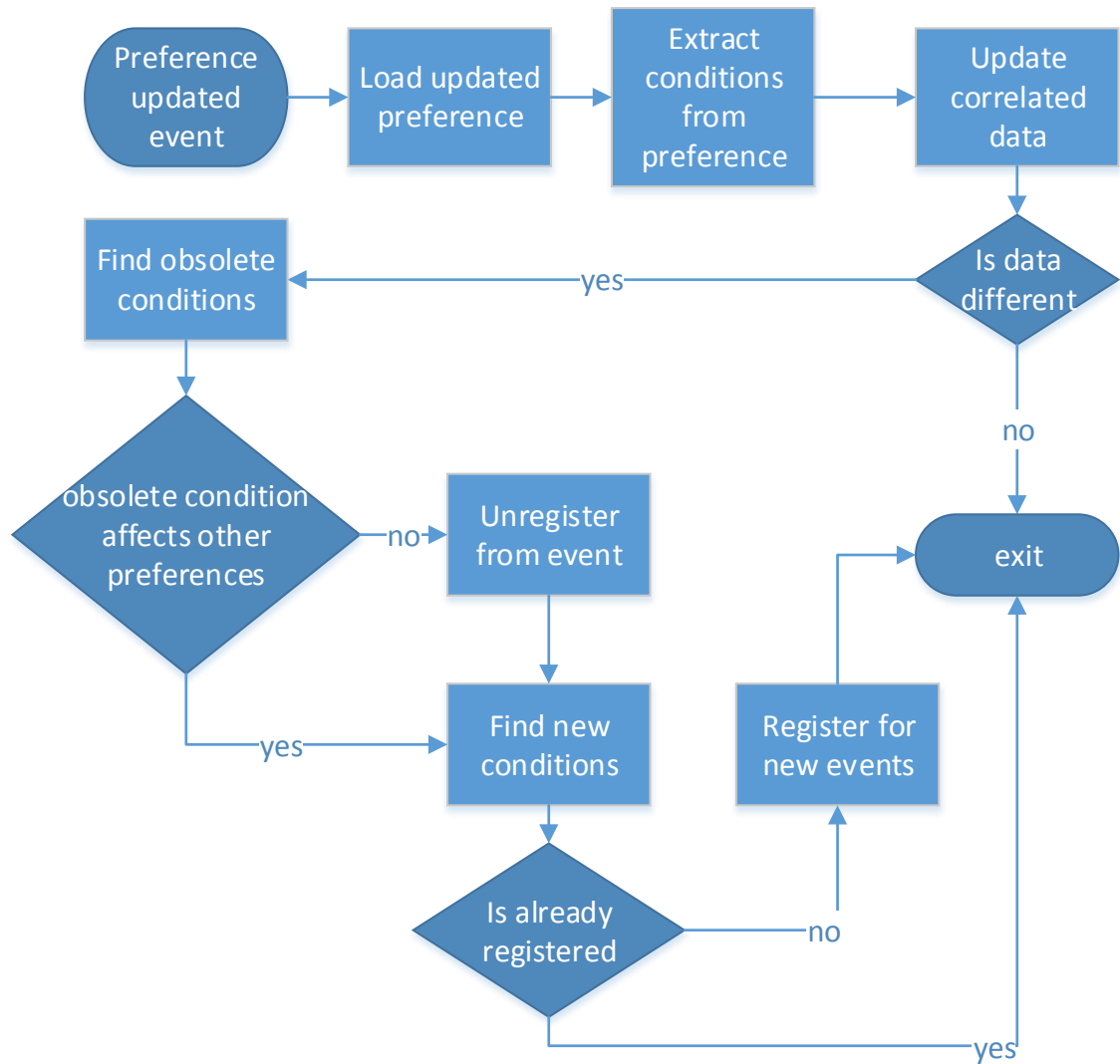


Figure 20. *Context event received.*

Receiving “preference updated” events

When a user manually edits a preference or when the learning algorithm learns a preference, an event is posted to notify listeners that a preference has been updated. The preference condition monitoring component receives these events and must ensure that the current correlated information between affecting context data and affected preferences and services is in accordance with the current preference set. The updated preference is retrieved and the conditions affecting the preference are extracted. The correlated data are updated accordingly with the extracted conditions. If a new context type affects the preference, it registers for events of the new context type or if a context type no longer affects this preference and that context type does not affect any other currently loaded preferences, it unregisters from events of this context type. The corresponding workflow is given in Figure 21.

Figure 21. *Preference updated event received.*

4.3 *Implicit Personalisation*

Maintaining the user preference set manually can become an arduous task as the number of services used by a single user grows. While it is beneficial to provide users with a graphical user interface to create and edit their user preferences, a pervasive approach to personalisation should also support the learning of user preferences through behaviour and situation monitoring. DAIDALOS, PERSIST and SOCIETIES make use of the C45 algorithm to mine behavioural information. Figure 22 shows the workflow of implicit personalisation in PERSIST and SOCIETIES.

Services volunteer user actions to the Personalisation & Learning System that indicate that a user has interacted with the service in some way. As soon as such an action is received, the Personalisation & Learning System retrieves a selection of context attributes that describes the user's context at the time when the action occurred. This is called the

context snapshot. The action and the escorting context snapshot are then stored in the User Behaviour database. The system waits until a sufficient number of actions have been received before triggering a learning cycle.

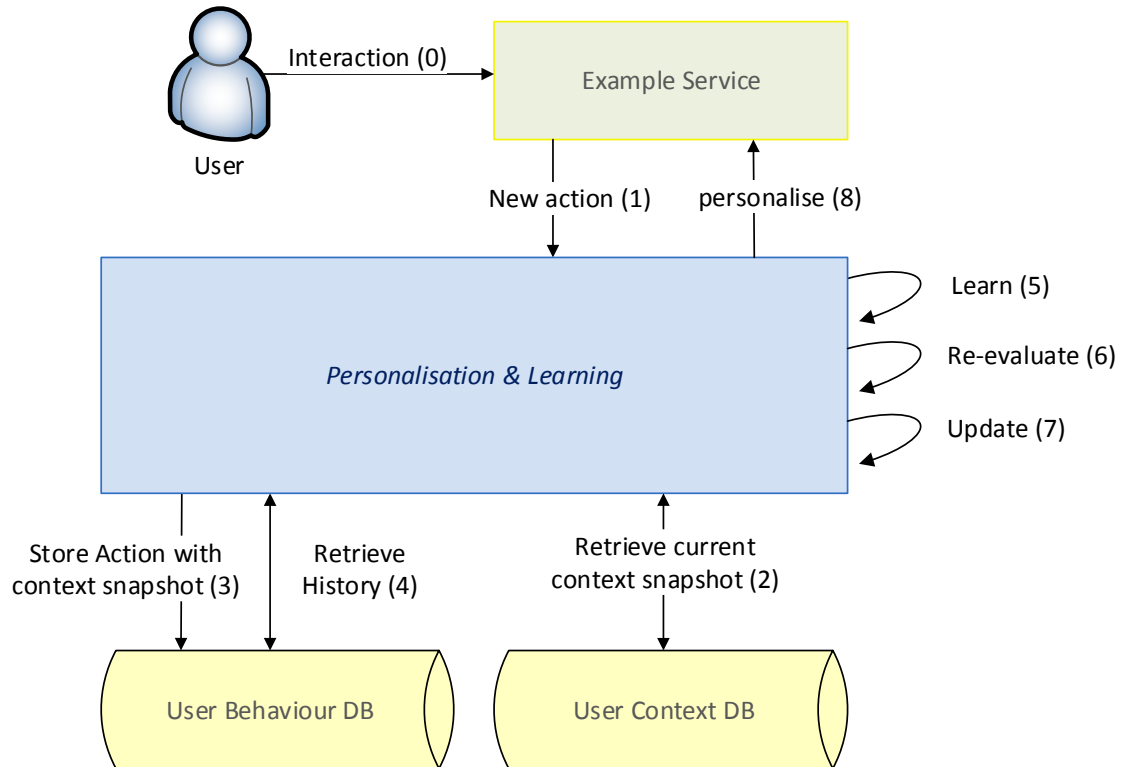


Figure 22. *Implicit Personalisation*

4.3.1 Preference Learning & Merging

The learning algorithm produces user preferences that are based on the user's recent behaviour history. These new preferences must be embedded within the existing preferences using an appropriate merging algorithm. There are four different situations that can occur when merging preference trees depending on how the preference conditions and outcomes of the new preference match those of the existing preferences.

- Situation 1: Identical preference conditions with identical preference outcomes.
- Situation 2: Identical preference conditions with different preference outcomes.
- Situation 3: Different preference conditions with identical preference outcomes.
- Situation 4: Different preference conditions with different preference outcomes.

Situation 1: Identical preference conditions - identical preference outcomes.

This is the case where the new preference matches the old preference entirely. Both preferences state that:

IF A==a THEN X=x FI.

The algorithm has to simply increase the confidence level of the preference. The confidence level is re-calculated using the confidence level calculation algorithm (described in section 4.3.2).

Situation 2: Identical preference conditions - different preference outcomes.

This happens when the user performs two different actions in the same context. In this case, the existing preference states that:

IF A==a THEN X=x FI but the new preference learnt states that:

IF A==a THEN X=y FI

These preferences cannot be merged because the two outcomes contradict each other and the preference evaluation algorithm will not be able to distinguish which of the two it should implement. To avoid having the preference evaluation algorithm solve this problem, the preference merging algorithm has to solve this problem in the preference merging phase. There are three solutions to this problem:

- a) Between the old and new preference, discard the preference with the lowest confidence level and decrease the confidence level of the preference with the highest confidence level using an appropriate algorithm.
- b) Instruct the learning component to run another learning cycle with a longer history of actions than the current history used for learning this new preference.
- c) Prompt the user to select which preference should remain and which one should be discarded or allow the user to edit the preference explicitly. This would set the confidence level of the preference to the highest value.
- d) Keep both conflicting actions in the tree and ensure that their respective confidence levels are up-to-date. This will be very helpful information to maintain when users change their behaviour.

The solution adopted in PERSIST and SOCIETIES uses a combination of solutions b and c in which the learning manager is instructed to re-mine the history of actions to yield a non-conflicting preference but it might again reach the same preference as previously. In this case, the user is prompted to edit the preference and the confidence levels is set to the

highest value because the user explicitly stated their wish. Using only solution a, could potentially discard a preference whose confidence level is not as low as to ignore it. The use of solution b is preferable as the new preference will be based on a larger history, hence it will take into account more information. Between solutions a and c, solution c is preferable as an appropriate graphical user interface can give the user the option to edit the preference and expand it to include other conditions that explain why the conflict emerged, a prospect that would not be possible with solution a.

Situation 3: Different preference conditions - identical preference outcomes.

Situation 3 has four subcategories in which the condition types and/or their values can be the same or different or a mixture of these differences. These are:

Situation 3.1: Different condition types. This situation arises if none of the conditions of the new preference is present in the existing preference tree. For example:

Assuming the existing preference states that:

IF A==a THEN X=x FI

And the new preference states that:

IF B==b THEN X=x FI

To merge this preference, the two conditional parts are merged with an OR operation:

IF A==a OR B==b THEN X=x FI.

The tree constructed from the merged preference looks like the tree in the following diagram.

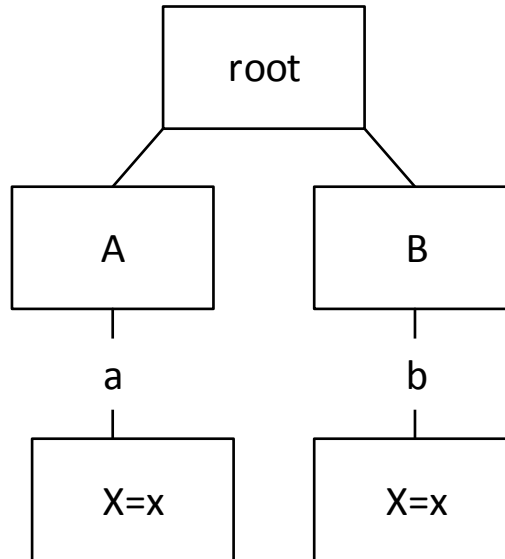


Figure 23. *Situation 3.1*

Situation 3.2: Identical condition types but different values. In this case, there is a complete match in the types of conditions present in the new preference with the condition types found in the existing preference but the values differ. The algorithm performs the same operation to merge the preference trees as in Situation 3.1:

Assuming the existing preference states that:

IF A==a THEN X=x FI

And the new preference states that:

IF A==aa THEN X=x FI

then the merged preference is:

IF A==a OR A==aa THEN X=x FI where the two conditional parts are merged with an OR operation. However, the tree shown in the Figure 24, shows a different picture than in Situation 3.1 because the conditions are the same.

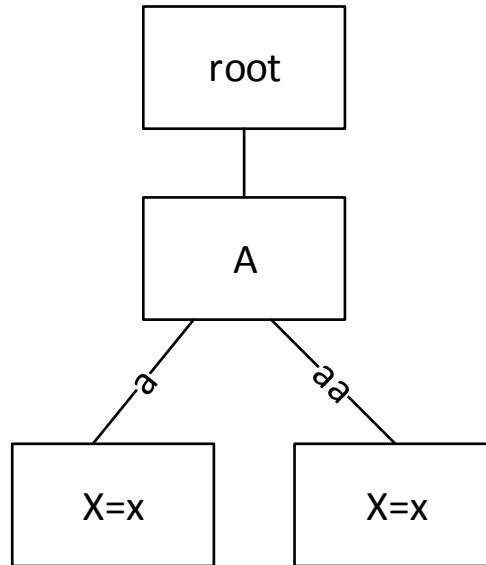


Figure 24. *Situation 3.2*

Situation 3.3: Combination of different and same types - different values. In this situation, some conditions exist in both the existing and the new preference with different values and some conditions exist only in one of them. For example:

Assuming the existing preference states that:

IF A==a OR B ==b THEN X=x FI

And the new preference states that:

IF A==aa AND C==c THEN X=x FI

The conditions are joined with an OR operation:

IF A==a OR B==b OR (A==aa AND C==c) THEN X=x FI

The tree of the merged preference is depicted in the figure below.

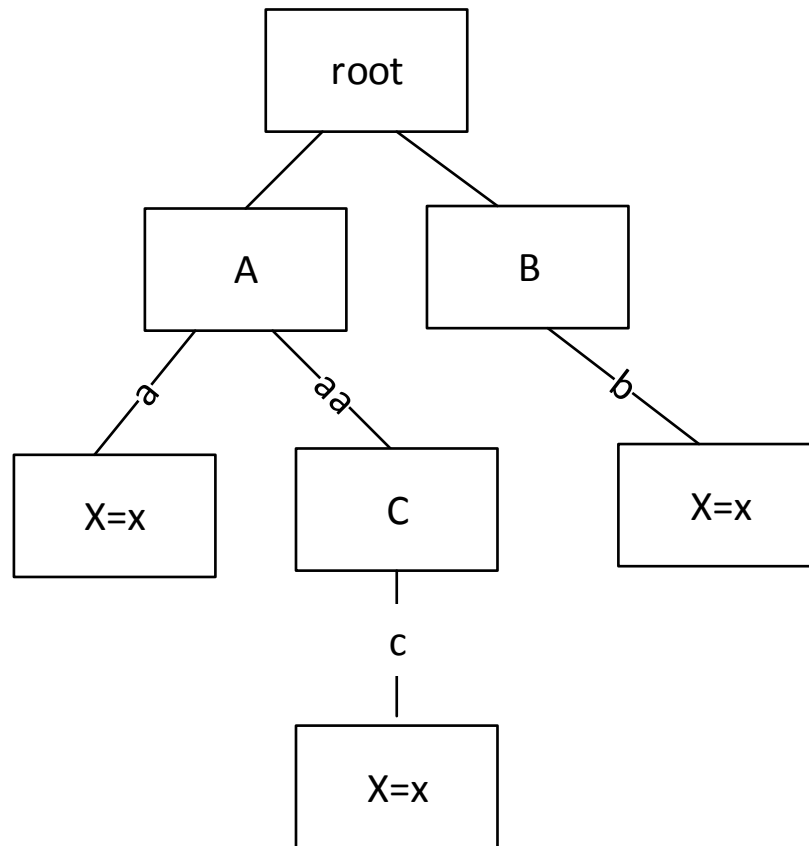


Figure 25. *Situation 3.3*

Situation 3.4: Combination of different and same types - identical values in the same condition types. In this situation, some conditions exist in both existing and new preferences and their values are the same and some conditions exist only in one of them. For example:

Assuming the existing preference states that:

IF A==a OR B==b THEN X=x FI

And the new preference states that:

IF A==a AND C==c THEN X=x FI

The new preference constructed by merging the two trees results in:

IF A==a OR B==b OR (A==a AND C==c) THEN X=x FI

The tree is depicted in the following diagram:

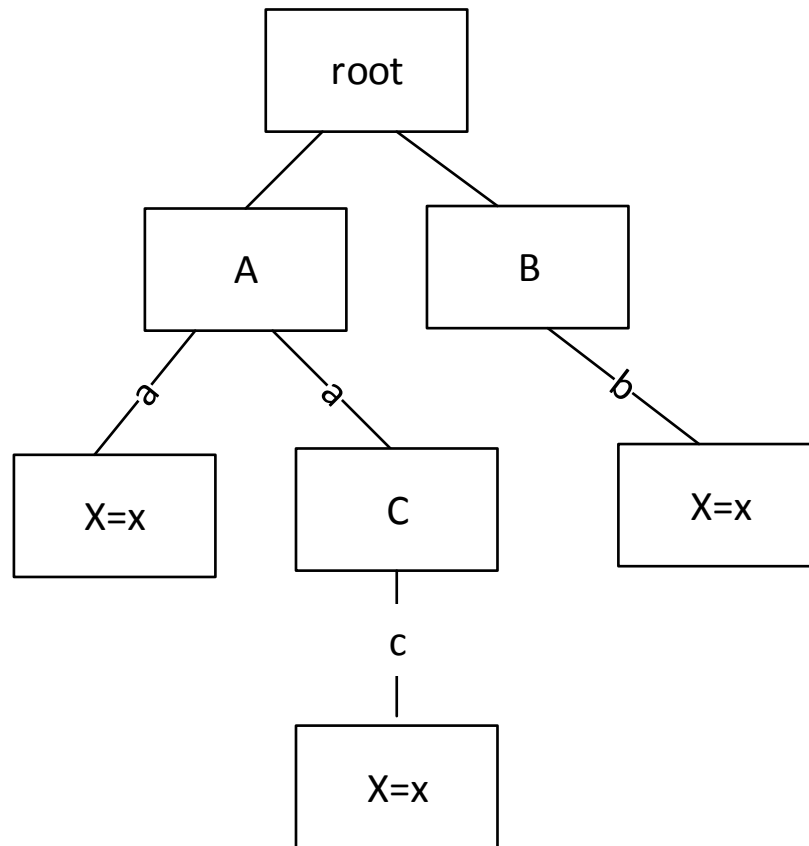


Figure 26. *Situation 3.4.*

The above expression can be simplified by removing the unnecessary Condition c using algebraic laws and the resulting expression is:

IF A==a OR B==b THEN X=x FI

However, it is preferable to keep as much information in the preference as possible because the tree could be updated frequently if the user performs the particular action (x) in different circumstances and retaining the Condition c can prove useful in future merging operations.

Situation 4: Different preference conditions - different preference outcomes.

This situation occurs when the outcomes are different and the combination of conditions varies. This situation is also broken down in four categories.

Situation 4.1: Different condition types. This situation arises if none of the conditions of the new preference is present in the existing preference tree. For example:

Assuming the existing preference states that:

```
IF A==a THEN X=x FI
```

And the new preference states that:

```
IF B==b THEN X=y FI
```

The merged preference is constructed as:

```
IF A==a THEN X=x
```

```
ELSE
```

```
IF B==b THEN X=y FI
```

```
FI
```

One problem that is immediately apparent is that it is possible that both $A==a$ and $B==b$ conditional expressions evaluate to true at the same time. There are two solutions to this problem:

- a) The algorithm leaves the tree as is and the preference evaluation algorithm takes into account the confidence level of the preference during the evaluation in order to decide what action to implement.
- b) The algorithm triggers a new learning cycle to mine a longer history of actions. As described earlier, by default, the learning cycle only learns from the last n actions and n is defined by a threshold to wait until a learning cycle can be triggered. By default, $n = 3$ so after receiving 3 actions, a new learning cycle is triggered. However, this can be changed if needed.

The implemented algorithm in PERSIST and SOCIETIES uses solution a) and allows the preference evaluation algorithm to solve the conflict if it exists. The reasons for which solution a) was adopted are because it is preferable to maintain as much information as possible in the preference tree for future merging operations along with the fact that the conflict will not necessarily occur all the time and it is unlikely that both conditions will evaluate to true during every evaluation cycle. Also, by continuously mining the history, learning more information about the preference and updating the confidence level, it is possible that one of the branches will become obsolete depending on the interaction of the user with the service. Figure 27 depicts the constructed preference tree.

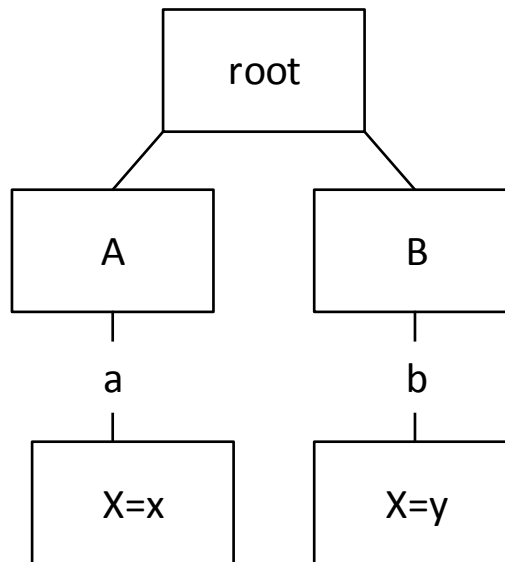


Figure 27. *Situation 4.1*

Situation 4.2: Identical condition types - different condition values. In this case, there is a complete match in the types of conditions present in the new preference with the condition types found in the existing preference but the values differ. For example:

Assuming the existing preference states that:

IF A==a THEN X=x FI

And the new preference states that:

IF A==aa THEN X=y FI

Then the result of merging the two preferences is:

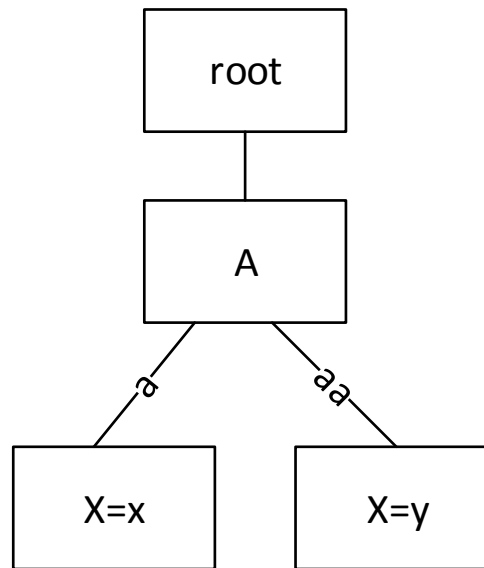
IF A==a THEN X=x

ELSE

IF A==aa THEN X=y FI

FI

The tree that depicts the preference is shown below:

Figure 28. *Situation 4.2*

Situation 4.3: Identical condition types - different condition values. In this situation, some conditions exist in both the existing and the new preference with different values and some conditions exist only in one of them. For example:

Assuming the existing preference states that:

IF A==a OR B ==b THEN X=x FI

And the new preference states that:

IF A==aa AND C==c THEN X=y FI

The result of merging these two preferences would yield:

IF A==a OR B==b THEN X=x

ELSE

IF A==aa AND C==c THEN X=y FI

FI

The same problem arises here as in situation 4.1. Expressions A==a OR B==b can evaluate to true at the same time as A==aa AND C==c because B==b can be true and A==aa be true. The solutions are the same as presented under situation 4.1 and again, the implemented solution in the algorithm updates the confidence levels of each preference outcome and eventually the preference evaluation algorithm can decide which action to implement.

The merged tree is depicted in the figure below:

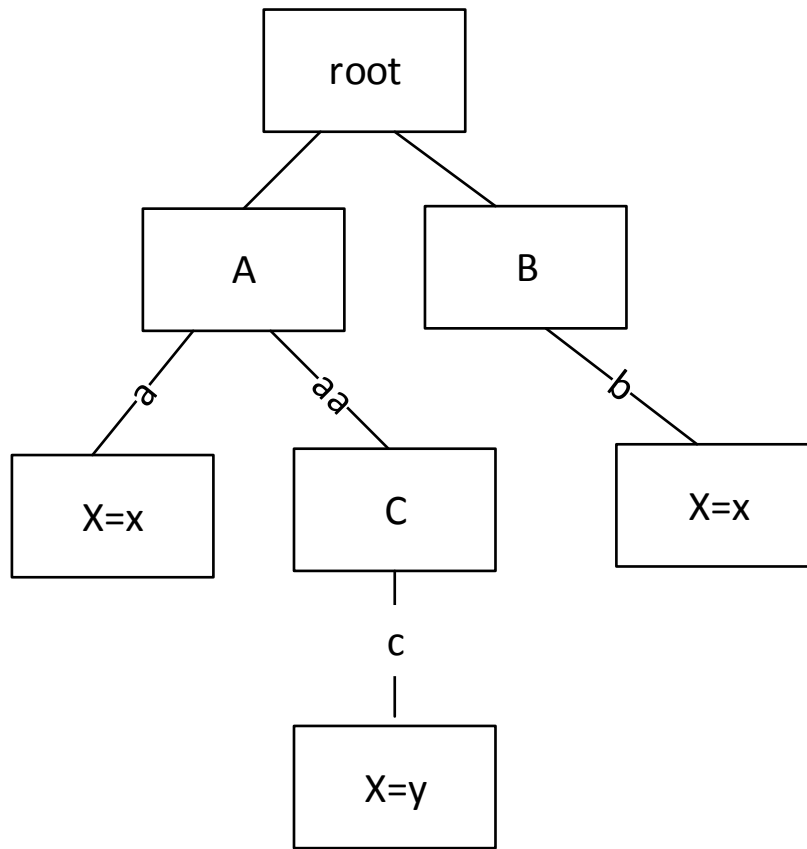


Figure 29. *Situation 4.3*

Situation 4.4: Combination of different and same types - identical values in the same condition types. In this situation, some conditions exist in both existing and new preferences and their values are the same and some conditions exist only in one of them. For example:

Assuming the existing preference states that:

IF A==a OR B==b THEN X=x FI

And the new preference states that:

IF A==a AND C==c THEN X=y FI

The merged preference that results from merging these preferences is:

IF A==a AND C==c THEN X=y

ELSE

IF A==a OR B==b THEN X=x FI

FI

The tree that depicts the merged preference is shown below

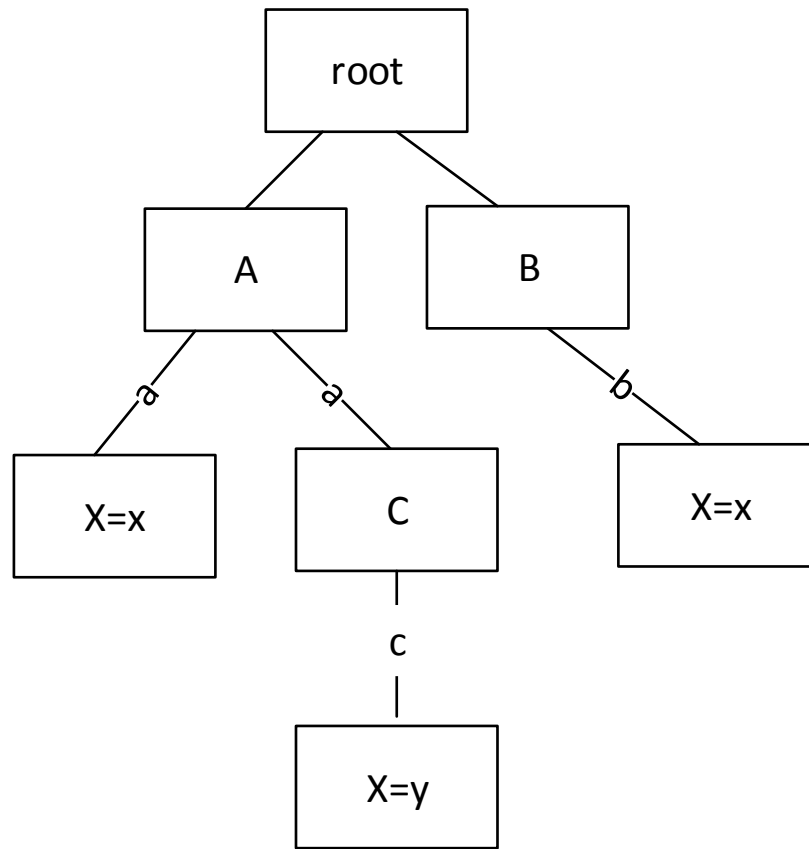


Figure 30. *Situation 4.4*

4.3.2 Preference Confidence Level

Users' preferences may change over time, particularly as newer and more sophisticated services become available to them. However, existing preferences stay in the system indefinitely unless the user removes them manually. This might have the undesired effect that the system will implement outdated or unwanted outcomes. To avoid this, it is useful to maintain a level of confidence in each action to indicate how confident the system is that the user prefers that this outcome is implemented under the stated circumstances. A confidence level can be attached to the outcome that is calculated based on how many times an outcome was successfully or unsuccessfully implemented by the system. During preference evaluation, the system can check the confidence level of an outcome to make sure that it should be implemented. The confidence level can be adjusted when the user aborts an implementation of an outcome or when they select it for implementation or when they implement that action themselves. A simple algorithm such as the one shown in the flowchart of Figure 31 can be used to adjust the confidence level of an outcome based on the user's activity.

```

private static final int MIN = 0;
private static final int MAX = 10;
public void updateConfidenceLevel(boolean positive){
    if (positive){
        switch (currentStage){
            case POSITIVE_1:
                confidenceLevel+=2;
                currentStage = Stage.POSITIVE_2;
                break;
            case POSITIVE_2:
                confidenceLevel+=3;
                currentStage = Stage.POSITIVE_3;
                break;
            default:
                confidenceLevel+=1;
                currentStage = Stage.POSITIVE_1;
                break;
        }
        if (confidenceLevel>MAX){
            confidenceLevel = 10;
        }
    }else{
        switch (currentStage){
            case NEGATIVE_1:
                confidenceLevel-=2;
                currentStage = Stage.NEGATIVE_2;
                break;
            case NEGATIVE_2:
                confidenceLevel-=3;
                currentStage = Stage.NEGATIVE_3;
                break;
            default:
                confidenceLevel-=1;
                currentStage = Stage.NEGATIVE_1;
                break;
        }
        if (confidenceLevel<MIN){
            confidenceLevel = 0;
        }
    }
}
}

```

Figure 31. *Confidence level calculation pseudocode*

The confidence level range is between 0 and 10. Each preference outcome includes a Stage field which suggests if the last two applications of this outcome were successful. When the outcome is applied, the confidence level algorithm is called with the appropriate Boolean value. A true value is passed as a parameter if the application of the outcome was successful and a false value if the user aborted the implementation of the outcome.

When the confidence level of a preference falls below 5, the outcome is not considered for implementation. When the confidence level goes above 7, the system is confident enough to implement this outcome automatically by notifying the user before implementing it and allowing them ten seconds to abort. If the confidence level is between 5 and 7 (inclusive), the system prompts the user to confirm the implementation of the outcome.

4.3.3 Learning service actions

The preference model as described in 4.1.1 can contain context conditions that define when a specific outcome should be applied. However, it is also possible to include the state of the services as conditions so that a preference can define that an Action y should be implemented when Action x is implemented. To realise this requirement, the personalisation system maintains the state of the services in the context database. A service is modelled as an Entity (as described in 3.2.2) and each of its personalisable parameters are modelled as attributes of that entity. These attributes are updated when the user performs actions that are monitored, which change the state of the service or when the personalisation system applies an outcome that changes the state of the service. By modelling the state of the services in the context database, the system avails itself from functionality provided by the context management system such as context event management. Hence, when the state of a service changes, the Preference Condition Monitor component can be informed in order to re-evaluate the preference that contains such preference conditions.

4.4 *Summary*

This chapter provided a detailed overview of personalisation for pervasive systems using a rule based approach. The concept of personalisation lies in the core of any pervasive system as the need to automate decisions and adapt the environment of the user according to their wishes is imperative.

A personalisation solution must satisfy several basic requirements to be acceptable in a pervasive system.

- It must take account of the user's environment as input to making decisions about adapting the services and resources available at any point in time by constantly monitoring the context of the user.
- It must allow the user a way to intervene if the personalisation it is applying is not aligned with the user's wishes.
- It must be able to adapt the services and resources in a proactive manner if it is confident enough that its decisions are what the user wishes by reacting to changes in the context and behaviour of the user.

- It must be able to learn from the user's behaviour and maintain appropriate up-to-date preference information by constantly monitoring the user's behaviour.

The personalisation approach presented in this chapter was employed in all three EU projects (DAIDALOS, PERSIST, SOCIETIES) with slight modifications to satisfy specific requirements of each platform. This rule based approach is an attempt to provide generic personalisation to services in a pervasive environment without the knowledge of particular workings of each service. The purpose of this decision was to maintain simplicity across all aspects of the personalisation system and using the same models and algorithms to personalise any kind of service suitable for a pervasive system without the need to understand how each individual service works.

5 Privacy in Pervasive systems

A pervasive system depends on the collection and processing of information, specifically of the personal data of its users. The quality of a pervasive computing environment is directly associated with the quality of the functionality it offers. This has been a source of controversy over the use of pervasive systems. It is an indisputable fact that there is a need for appropriate privacy protection mechanisms in any pervasive computing environment.

5.1 *Privacy policies*

A privacy policy describes how a service collects, stores, uses and disseminates the personal data of its users. More specifically, it states a) what data are requested, b) the purpose for which this data is requested, c) what type of processing will be applied to this data, d) with whom this data will be shared with and e) how long this data will be stored for. It may also include other statements about the rights of the user as well as the service provider with regard to the data. With the use of privacy policies, companies, and more specifically services, inform their users about what happens to the users' personal data after disclosure.

5.1.1 **The Request Policy**

A Request Policy document is a privacy policy expressed in machine readable format. A Request Policy comprises a Subject element and a list of Request Item elements. The Subject specifies the party to which the policy refers to, i.e. the service and service provider information. Each Request Item element refers to a specific piece of data that the service will request access to during the use of the service by the user (e.g. name, email, address, current location, etc). Each Request Item states the following (see Figure 32):

- The type of data (defined in the <Attribute> inside the <Resource> attribute).
- The purpose for which this data is requested in a user friendly manner (defined with the <Purpose> attribute).
- The type of operation it will need to perform on the data (e.g. READ, WRITE, CREATE, DELETE defined with the <Action> attribute).

- The list of conditions (each condition is defined with the <Condition> attribute) that defines restrictions on both the service provider and the user such as how these data will be shared with other parties and how long the data will be kept by the service provider (data retention period). Other conditions can also be included such as the right of the user to opt out of the service at any time and other rights or obligations of the user the service might request (a full list of conditions is provided in Appendix X).
- Whether this data type is optional for running the service (defined with the <Optional> attribute). If it is marked as not optional, then this means that the service cannot operate without this information.

```

<Request>
  <Resource>
    <Attribute AttributeId="context"
      DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>locationSymbolic</AttributeValue>
    </Attribute>
  </Resource>
  <Purpose>Your location is required to offer you services near you.
</Purpose>
  <Optional>>false</Optional>
  <Action>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
      DataType="org.societies.api.schema.privacytrust.privacy.model.privacypolicy.ActionConstants">
      <AttributeValue>READ</AttributeValue>
    </Attribute>
  </Action>
  <Action> ...
  <Condition>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:condition-id"
      DataType="org.societies.api.schema.privacytrust.privacy.model.privacypolicy.ConditionConstants">
      <AttributeValue DataType="DATA_RETENTION">12 hours</AttributeValue>
    </Attribute>
  </Condition>
  <Condition> ...
</Request>

```

Figure 32. Snippet of a Request Policy document in XACML format.

XACML as a Privacy Policy specification language

XACML was chosen as privacy policy specification language due to its simplicity and the fact that it was easily extensible. However, any language that is extensible can be adapted to specify RequestPolicy and ResponsePolicy constructs such as the P3P specification language. As shown in the examples of Figure 32 and Figure 34, a list of privacy conditions is defined for a single data type. The XACML specification allows privacy conditions (defined as <Obligation> attributes) to be defined for all the data types

included in the privacy policy. This has been altered to allow users to request different terms and conditions per data type. For example, a user can request a short data retention period for their location data and a longer one for their email address or postal address.

5.2 *Privacy Policy Negotiation*

Privacy Policy Negotiation (PPN) is the process in which a user negotiates the terms and conditions of the privacy policy with a service. It is a solution to the “take it or leave it” approach that is currently being implemented by millions of services worldwide. Negotiating for privacy means freedom for the user to adjust their privacy as they wish rather than fitting the preferences of the service provider.

A system can present to the user in a visually friendly manner what data will be requested of them and what happens to the data after disclosure. An appropriate negotiation user interface allows the user to edit the Request Items present in the Request Policy, edit the conditions and tailor the privacy policy to fit their preferences with regard to privacy. From the viewpoint of the service, a customised privacy policy means that the Quality of Service to the user will be proportionate to the restrictions imposed by the user in the privacy policy. For example, a user can demand that a map service does not disclose their location to 3rd parties. If the map service is integrated with services offering discounts in stores nearby, the user will not be informed of these.

A suitable negotiation protocol should be able to terminate in a finite number of message exchanges so as not to deadlock the system into an infinite loop of offers and counteroffers. Also, during the negotiation process, it is required that the user remains anonymous so that any information disclosed during the negotiation cannot identify the user.

The Privacy Policy Negotiation process is initiated by the user who wants to install or use a service or application in one of their devices. During the negotiation, a negotiating entity termed Negotiation Client acts on behalf of the user and another termed Negotiation Agent acts on behalf of the service provider. These two entities exchange messages and are responsible for implementing the Privacy Policy Negotiation algorithm. The Negotiation Client retrieves the service’s Request Policy from the Negotiation Agent and processes it. Using an appropriate user interface, the Negotiation Client shows the contents of the Request Policy to the user, clearly indicating with whom the user is

negotiating, the data and respective actions and conditions on that data and the purpose for which the data is requested. Figure 33 demonstrates a graphical user interface allowing the user to make adjustments to the proposed Request Policy of the service.

Privacy Policy Negotiation with Lollipop Ltd

The information below shows what data will be used during your using of Lollipop Ltd services. Configure usage per your needs and click continue.

- name**
- age**
- GPS location**
 - Purpose: Your location will be tracked to offer you services nearby.
 - Actions:
 - Read
 - Write
 - Create
 - Delete
 - Conditions:
 - Share with 3rd parties Keep data for **1 week**
 - Right to opt out
 - Decision:
 - allow
 - deny
- activity**

Cancel Continue

Figure 33. Example of a Privacy Policy Negotiation GUI

After the user reviews the items and makes the necessary adjustments to satisfy his privacy needs, the system generates a document based on the Request Policy of the service and the user's responses and adjustments. This document expressed in a machine readable format is termed the Response Policy.

5.2.1 The Response Policy

A Response Policy document is constructed during the negotiation to state the user's agreement or disagreement with the Request Policy of a service. The Response Policy comprises the Subject element which identifies the service with which the user is

negotiating and a list of Response Item elements, one for each Request Item present in the service's Request Policy document. As shown in Figure 34, the Response Item encapsulates the Request Item and the decision of the user regarding this Request Item. The Request Item is modified based on the user's input to the PPN user interface (Figure 33). If the user accepts the terms of the original Request Item and does not make any changes, the <Decision> attribute is marked with "PERMIT". If the data item is marked as optional, the user can indicate that they do not wish to disclose it at all. Then, the <Decision> attribute is marked with "DENY". If the user makes changes to the conditions for disclosing this attribute, the <Decision> attribute is marked with "INDETERMINATE".

```

<Response>
  <Decision>
    <Attribute AttributeId="Decision"
    DataType="org.societies.api.schema.privacytrust.privacy.model.privacypolicy.Decision">
      <AttributeValue>INDETERMINATE</AttributeValue>
    </Attribute>
  </Decision>
  <Target>
    <Request>
      <Attribute AttributeId="context"
      DataType="http://www.w3.org/2001/XMLSchema#string">
        <AttributeValue>locationSymbolic</AttributeValue>
      </Attribute>
    </Resource>
    <Purpose>Your name is needed so you can be identified by your contacts.
    </Purpose>
    <Action>
      <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
      DataType="org.societies.api.schema.privacytrust.privacy.model.privacypolicy.ActionConstants">
        <AttributeValue>READ</AttributeValue>
      </Attribute>
      <Optional>>false</Optional>
    </Action> ...
    <Condition>
      <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:condition-id"
      DataType="org.societies.api.schema.privacytrust.privacy.model.privacypolicy.ConditionConstants">
        <AttributeValue DataType="RIGHT_TO_ACCESS_HELD_DATA">yes</AttributeValue>
      </Attribute>
    </Condition>
    <Optional>>false</Optional>
  </Request>
</Response>

```

Figure 34. Snippet of a Response Policy document in XACML format.

5.2.2 Negotiating with the service

After generating the Response Policy based on the user's input, the Negotiation Client sends the document to the Negotiation Agent and initiates a new privacy policy negotiation session. The Negotiation Agent receiving the Response Policy will inspect all

the Response Items to see if it can satisfy the user’s requests. The Negotiation Agent acting on behalf of the service maintains an internal set of rules that define the ranges in the values of the conditions that can be satisfied by the service. For example, the service provider sets a minimum and a maximum data retention period. The maximum retention period is also set in the Request Policy, while the internal rules indicate both the minimum and the maximum value for use in the negotiation process.

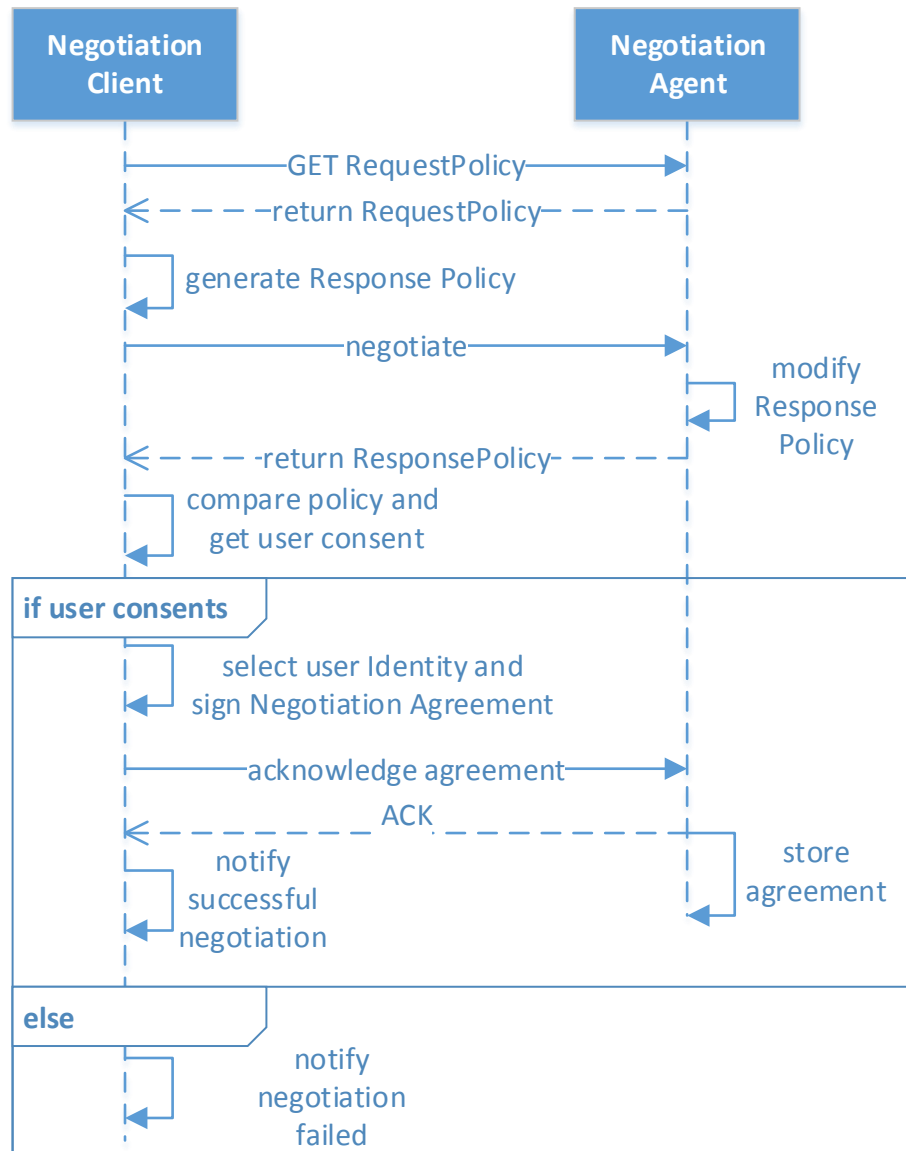


Figure 35. Privacy policy negotiation protocol.

For example, the Request Policy might set a data retention period of one month for GPS data, while the internal rules set a range of satisfiable values between twenty four hours and one month. These ranges are set by the service provider according to their capabilities and the functionality of the service. For example, a service provider may not be able to provide functionality for the user to view what data have been disclosed by them. Thus,

the rule for negotiating this condition would define that this functionality is not available. Using these rules, the Negotiation Agent modifies the Response Policy to match its own abilities and returns it to the user.

Using the privacy policy negotiation user interface, the Negotiation Client presents the service's Response Policy, highlighting the sections where the service was not able to meet the user's demands. In the XACML notation, the changed items are marked using the <Decision> attribute. The user can either abort the negotiation at this point if they are not satisfied with the response of the service or accept the Response Policy and continue to finalise the result of the Negotiation.

5.2.3 The Negotiation Agreement

A successful negotiation terminates with the signing of a Negotiation Agreement document that states what has been agreed during the negotiation process. If an Identity Management system is available to the user which makes use of multiple identities (see section 5.3), the user is prompted to select a digital identity to represent themselves to the service. The Negotiation Agreement is constructed using the agreed Response Policy document, the user's and the service provider's identities and the service identity (identifying the service uniquely). The Negotiation Agreement is then signed by the user digitally and then sent to the service for signing and acknowledgement. Upon receiving the Negotiation Agreement, the service checks to make sure it has not been altered since the previous round of negotiation and then signs it and returns it to the user.

The Negotiation agreement then defines the terms and conditions for all subsequent use of user information by the service provider which must be adhered to.

5.3 *Digital Identities*

A digital identity is an identifier that can represent a user in the system. A digital identity is also associated with a set of data which are part of a user's profile. A suitable Identity Management system for pervasive environments allows the user to use multiple identities to represent themselves to services, service providers, and other entities on the network in different contexts. Users may wish to present a different view of themselves by selectively revealing different items of information under different identities. If the user profile can be thought of as a theoretical concept that defines the set of all personal data

held in the system about the user, a digital identity can be regarded as an identifier that can give access to a subset of that data.

5.3.1 Identity Selection

Identity selection is the responsibility of the underlying Identity Management system. It is a process that is usually performed after the privacy policy negotiation stage and its primary input is the Negotiation Agreement document. As described in chapter 3, an Identity is associated with a set of data in the user's profile. Hence, the identity selection process entails looking at the set of data types defined in the Negotiation Agreement and matching them to the set of data associated with each of the Identities in an effort to select the most appropriate digital Identity. After finding one or more identities that can satisfy the requirements presented in the Negotiation Agreement, the user is asked to choose one of them or create a new identity if they wish.

Creating a new identity even though existing identities can be reused allows the user to represent themselves to each service as a different user. This means that the details of their interaction with one service cannot be linked to details of their interaction with another. In case none of the existing identities are suitable, a new digital identity should be created and associated with the data defined in the Negotiation Agreement.

As explained earlier, the user profile contains all the data that the system holds about the user. Some data attributes have multiple values. For example, a user can have many e-mail addresses, postal addresses, jobs, cars, can speak multiple languages and so forth. Supporting multiple values allows for a good level of flexibility as it allows the user to select which attribute of a specific type they prefer to associate with a specific Identity. For example, a user can create two identities, one for work related activities and the other for personal activities. The former would be associated with the attribute holding the e-mail address that they use for work correspondence and the latter with the attribute holding the e-mail address used for personal communication. Associating specific attributes with an Identity is part of the identity creation process. Selecting the appropriate attributes to associate with the new Identity depends on the service that the identity will be used for and the context in which the service will be used.

5.4 *Access Control*

Privacy policy negotiation and identity selection are two processes completed sequentially before a user makes use of a service. After these two processes complete successfully, the system must take care to continue protecting the privacy of the user by restricting access to the user's data using access control rules. Access control refers to allowing or blocking access to a data record when a request is made by a service. While the Negotiation Agreement sets out the terms and conditions for disclosing data, it does not warrant an ongoing access to the data agreed to be disclosed. Access control rules must be maintained in the system to define if access should be granted when a request for data is received.

Access control rules define an 'allow' or 'block' decision for access to perform a specific action to a specific data attribute by a specific requestor. A list of access control rules can be accrued in the system by prompting the user to confirm or block access to the data and storing that decision as an access control rule. Hence, a component providing access control maintains a list of permissions that denote if a subject (i.e. a service) is allowed to perform a specific action (read, write, create, delete) on a data item specified with its identifier.

However, in pervasive environments, the value of many data are continuously changing; specifically, data that come from sensors and services that monitor the behaviour and activity of the user. For those kinds of data, the actual value of data can be an important factor in deciding whether the data should be disclosed. There is a clear need for context-dependent access control rules. For example, the user might be willing to allow their boss to monitor their location and activity while they are at work but not when they are not working. The system must be able to provide this simple functionality. It is evident that traditional access control rules that define static rules allowing or blocking access to a data item are not sufficient in a pervasive service environment. However, defining context-aware access control rules to manage the disclosure of data is not enough; monitoring of the constantly changing data that affects the decision whether to allow or block access to a requestor is needed. Specifically, in the case of pervasive environments where smart services act in a proactive manner, listening to changes in the user's context and reacting to those changes to provide the best available information and service in the current context of the user, the system must be able to proactively make changes to the

access control rules making sure that data disclosure happens according to the wishes of the user defined in a set of context aware access control rules.

5.5 *Data Obfuscation*

Access control can be enhanced with the use of data obfuscation techniques. The purpose of using data obfuscation is to make the data more generic, thus limiting data disclosure to a level that is still acceptable for running a service while respecting the user's privacy. In some cases, services receive a lot more detailed information than needed. For example, consider a service that is only available to devices present in the UK. For a user to be able to use the service, they will have to prove the device is present in the UK by allowing the application to retrieve GPS information from the device. Even though the query is very generic, the response is a lot more detailed than needed by divulging very precise location information to the service. In such cases, data obfuscation can be very helpful. In addition to these cases, users may wish to limit the information they divulge even though that might diminish the Quality of Service they receive.

Data obfuscation can be applied to different degrees depending on the type and value of data. For example, the postal address "10 Downing St, London, SW1A 2AA, UK" can be obfuscated fully to "UK". However, for some services, full obfuscation might render the service useless, for example in the case of a restaurant finder service. Therefore, a good obfuscation algorithm should be able to offer different levels of obfuscation. In the case of a restaurant finder service it could suggest the obfuscated value of "Westminster, London, UK" or "London, UK".

5.5.1 **Data Obfuscation Limitations**

While data obfuscation is a very useful privacy enhancing technology, there are certain limitations and drawbacks to applying data obfuscation to everything.

- **Applicability.** Data obfuscation cannot be applied to any type of data because some types cannot be obfuscated. For example, it is not possible to obfuscate a binary value.
- **Availability.** In order to obfuscate data of a given type, an appropriate algorithm must exist that is able to process the data type appropriately. For example, the system would need a suitable algorithm to obfuscate location coordinates,

another algorithm to obfuscate the user's age, another algorithm to obfuscate their date of birth, their religion, their ethnicity and so forth. The number of data types is not fixed. As new sensors, services and applications are encountered, the number of data types grows and new algorithms will be needed to obfuscate them.

- Resource demanding. Data obfuscation must be performed on a request basis because different obfuscation levels may be applied to the data according to the requestor and the current value of the data. Running an obfuscation algorithm each time a request for the data is received can be computationally expensive, especially for mobile or embedded devices where most sensitive context data are likely to be collected.

5.6 *Summary*

This chapter provided an overview of a set of privacy enhancing technologies that can be combined to provide a comprehensive approach to privacy protection. These are: privacy policy negotiation, multiple identities, context-aware access control rules and data obfuscation.

Privacy Policy Negotiation allows the user to configure the terms and conditions of a service and tailor them according to their own privacy requirements and demand higher privacy than that being offered by the service, provided that the service can satisfy that demand.

The use of multiple identities allows the user to present themselves differently in different contexts to different services with appropriate user information. The “dis-integration” of the full set of data of the user's profile with the use of multiple identities impedes third parties from collecting vast amounts of user information.

Access control rules embedded with context-awareness functionality provide a differentiating decision making approach to traditional access control systems. Allowing access to certain resources according to the current context of the user is appropriate in the context of pervasive systems as different information with varying quality will be necessary in different situations.

On top of context-aware access control rules, data obfuscation strengthens the system in delivering information in a form that is both adequate for the purposes to which it is being put whilst maximising the privacy of the user.

Privacy Policy Negotiation provides an alternative solution to the “take it or leave it” concept that is currently being offered in traditional systems. However, pervasive systems are a lot more demanding in terms of information than traditional systems. The more information they have about the context and preferences of the user, the better they can perform. In spite of this, the quantity and quality of the information disclosed must be proportionate to the purpose for which the information is requested. Appropriate terms and conditions must be demanded by the user to guarantee that any disclosed information is processed, stored and eventually deleted according to the wishes of the user.

These privacy enhancing technologies are combined to deliver comprehensive privacy protection to the user. Primarily, the level of privacy that can be delivered depends on a) the demands that the user makes during the privacy policy negotiation, b) the manner in which the user creates and manages their identities, and c) the data disclosure that the user allows. Hence, the user decides the level of privacy protection that is to be applied. However, not all users have the requisite knowledge about how to achieve the level of privacy they desire using any kind of privacy protection technology. Therefore, the system must provide appropriate tools to help the user make use of these technologies in order to achieve and maintain the desired level of privacy protection. The PersoNISM system described in the next chapter describes a comprehensive privacy protection system based on such user-friendly functionality.

6 *PersoNISM* - PERSONalised Negotiation, Identity

Selection and Management

This chapter presents the PersoNISM (PERSONalised Negotiation, Identity Selection and Management) system. The PersoNISM system provides a comprehensive approach to privacy protection in pervasive environments using context dependent personalisation and learning functionalities. The PersoNISM system makes use of the technologies presented in the previous two chapters; personalisation using context dependent user preferences and user behaviour learning designed to help the user achieve the desired level of privacy protection based on privacy policy negotiation, identity selection and context-aware access control enhanced by data obfuscation.

The goals of the PersoNISM system are twofold. The first goal is to equip the user with the necessary tools to protect their privacy as they see fit and the second goal is to help the user make the best use of these tools using a set of privacy preferences designed specifically for personalising these privacy tools according to the user's wishes.

The User Privacy Preference Model is the foundation of the PersoNISM system. In essence, the User Privacy Preference Model holds all the information that drives the functionality of the PersoNISM system. The personalisation techniques described in chapter 4 are designed primarily to satisfy the requirements of adapting third party services in a specific context without knowing the specific functionality of these services. In the case of PersoNISM, personalisation is used for the purposes of a) tailoring privacy protection technologies according to the user's wishes and b) helping the user make informed decisions about their privacy. The functionality and the requirements of each privacy protection technology employed are known in advance. Therefore, the personalisation system can be designed specifically to accommodate the specific requirements of Privacy Policy Negotiation, Identity Selection, context-aware access control and data obfuscation. Hence, the User Privacy Preference Model contains five different forms of user privacy preference rules, designed to personalise these technologies where the user is involved. These are: privacy policy negotiation preferences, identity selection preferences, attribute selection preferences, access control preferences and data obfuscation preferences.

6.1 *PersoNISM architecture*

The PersoNISM system exists within a pervasive service environment that offers a multitude of services such as user context data management, identity management and trust management. Figure 36 shows the PersoNISM architecture blocks and their interactions with such services in the context of a pervasive platform similar to the platforms implemented during the three EU projects DAIDALOS, PERSIST, SOCIETIES. Figure 36 shows the set of components that are active in a pervasive platform running on behalf of a user. In the case of a service provider platform, only the Privacy Policy Management and Privacy Policy Negotiation components are necessary. In a pervasive platform where users can act as micro-operators providing services to other users (e.g. the Personal Smart Space platform), the PersoNISM system has a dual role to provide functionality that serves both service provider and service consumer.

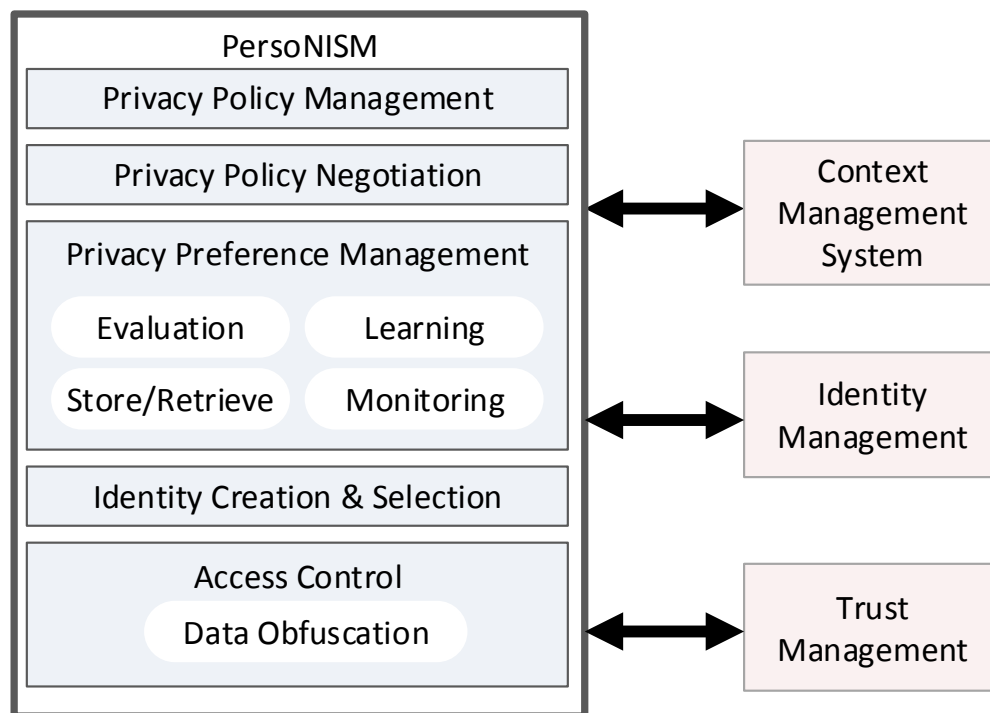


Figure 36. *PersoNISM architecture blocks.*

The Privacy Policy Negotiation process is implemented within the Privacy Policy Negotiation component that implements the Negotiation Agent and Negotiation Client functionality needed by the service provider and service consumer respectively. The Privacy Policy Management component running on behalf of the service provider has the responsibility of maintaining the set of privacy policies for the services that this service provider advertises.

From the point of view of a service provider, the privacy policy negotiation requires a privacy policy and a set of options that define the range of satisfiable privacy condition values to be used when a user sends a ResponsePolicy document to the Negotiation Agent. These options are declared when a service first becomes available and are maintained in the Privacy Policy Management component. While the developer of the service defines the data items that the service needs to operate, the service provider must make sure that the privacy policy attached to the service defines the Privacy Conditions in the policy and the range of satisfiable options for each of these data items according to the capabilities of the service provider and the implementation of the service.

From the point of view of the service consumer, the privacy policy negotiation requires input from the user either through the graphical user interface or through an automatic Response Policy generation algorithm that is based on the user's privacy preferences (described in section 6.2.3). When the privacy policy negotiation process is finalised with a successful agreement, the Identity Selection and Creation component needs to deliver an identity to be used to represent the user to the service. The process of selecting an appropriate identity takes into account the Negotiation Agreement and a set of privacy preferences designed to guide the component in selecting an identity that can satisfy the requirements of the service but more importantly in selecting the right identity depending on the current context of the user and the trustworthiness of the service.

Access Control and Data Obfuscation handle the disclosure of the data to services. Even though the Negotiation Agreement dictates the terms and conditions for data disclosure, the user retains the right to deny disclosing information to services at any moment. This is particularly important in disclosing contextual information that describe the environment of the user, their current location, activity, other people near them and other sensitive information. This information is used in deciding when and to whom to disclose what information and in what quality.

6.2 *Personalisation in Privacy Policy Negotiation*

During the privacy policy negotiation process, the user can configure several options for each data item that a service is requesting in order to tailor the use of the service with the quality of privacy that the user wishes. While this process is very flexible in terms of empowering users to request the desired level of privacy, the number of options presented

to the user can appear cumbersome. Personalisation in the form of user preferences can be used to aid the user in performing privacy policy negotiation by suggesting options that fit the user's previous privacy demands. Such user privacy preferences can be created either manually by the user or by using an appropriate learning algorithm that uses the user's previous configurations during privacy policy negotiations as input to produce a set of user privacy preferences that can be used to guide the user through future privacy policy negotiations. Such an algorithm should also be able to merge existing privacy policy negotiation preferences with newly formed ones as the user installs new services on their devices.

6.2.1 Privacy Policy Negotiation (PPN) Preferences

Traditional user preferences are designed in a way that they can be applied to a variety of services but in the case of using user preferences for personalising privacy policy negotiation, that requirement does not apply and therefore the data model can be designed to fit the needs of this task explicitly. Knowing the semantics of the conditions and the outcomes gives much greater flexibility in designing an appropriate data model for privacy policy negotiation preferences.

PPN Preference Outcome

Maintaining the basic IF-THEN-ELSE structure format for the skeleton of the privacy preferences allows the presentation of these rules to the user in a user friendly manner. In a privacy policy negotiation, a PPN preference is used to personalise privacy condition statements, where the user is able to make changes and request the level of privacy they wish. Therefore, the outcome of the PPN preference defines the value that should be set in a <Condition> element. The PersonISM privacy policy negotiation specification defines seven types of privacy Conditions:

- *Data retention.* This condition defines the time period that the service can keep the data in their servers. The values of this condition are in the range of 1 hour, 2 hours, 4 hours, 6 hours, 12 hours, 24 hours, 36 hours, 48 hours, 1 week, 2 weeks, 1 month and until the account is deactivated.
- *Data sharing.* This condition defines the right of the service to share (or sell) the data to others. The values of this condition are: no sharing, sharing with affiliated services, sharing with 3rd party services, sharing with everyone.

- *Secure storage.* This condition defines the right of the user to demand that their data is stored encrypted and in a secure database by the server. Information on how secure the storage and how strong the encryption is can also be provided in the RequestPolicy document. This condition has a binary value (yes-no).
- *Inference.* This condition defines the right of the service to manipulate the user's data, combine them with other information to infer further information about the user. This condition has a binary value (yes-no).
- *Opt-out clause.* This condition defines the right of the user to deny disclosing data to the service at any time. This condition has a binary value (yes-no).
- *Access to data.* This condition defines the right of the user to access the data that the service maintains about the user. This condition has a binary value (yes-no).
- *Edit the data.* This condition defines the right of the user to edit the data that the service maintains about the user. This condition has a binary value (yes-no).

The user preference outcomes used in service personalisation allow services to define the values of the outcomes themselves to fit the needs of their service. In the case of PPN Preference Outcomes, the key-value pairs are already defined. The key can only be one of the seven types of privacy condition, and each key has a set of predefined values.

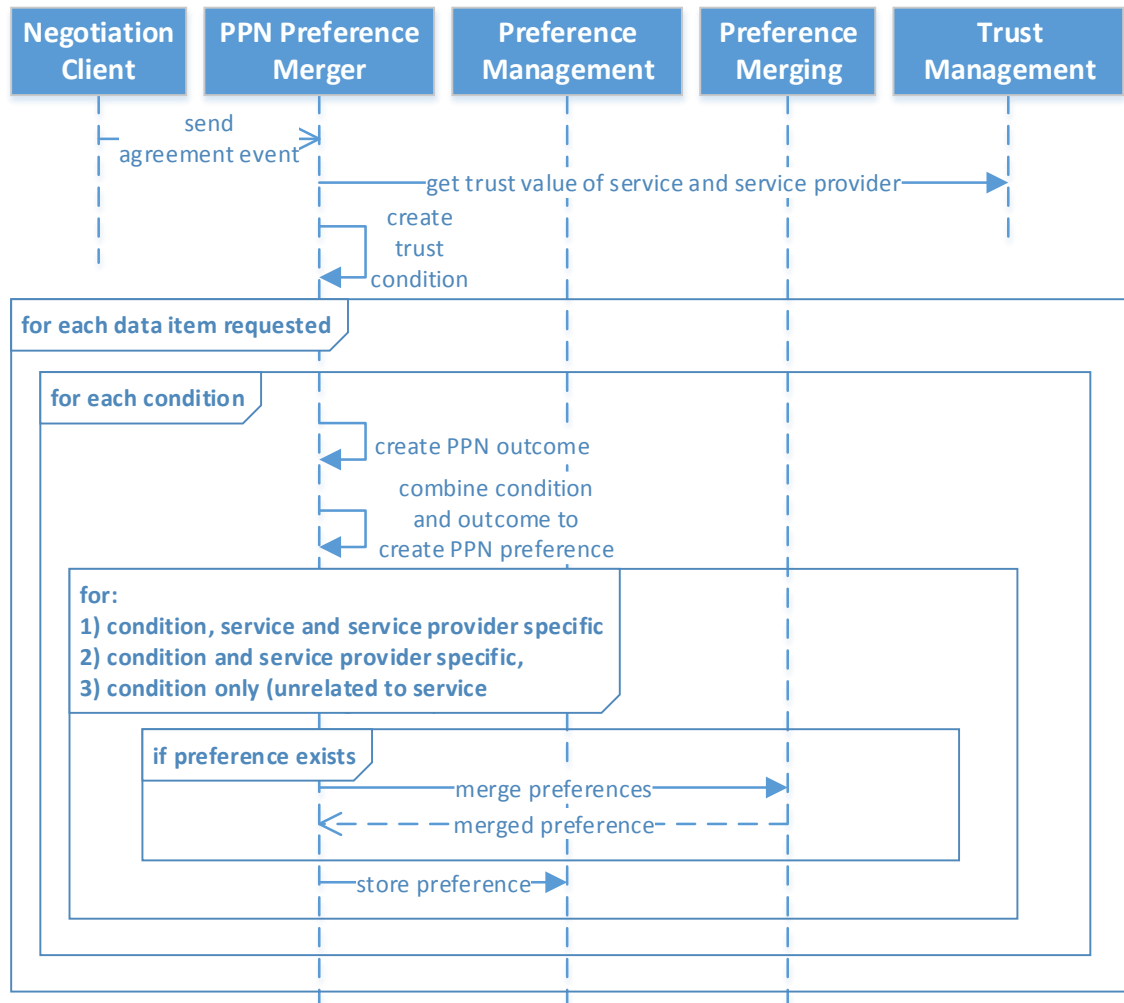
PPN Preference Condition

PPN preferences are used during a Privacy Policy Negotiation process. They define the value of the condition that should be used for a specific data item. These preferences do not need to be context-dependent as the current context of the user does not affect the decision about the terms and conditions for disclosing data. However, they need to be trust dependent. The trustworthiness of the service and the service provider can affect the decision for requesting higher levels of privacy. Like context information, trust information can also be used in a conditional statement that defines what value should be set if the trust level of a service is above some threshold. Querying the user about their perception of the service's trustworthiness with regard to the service's privacy tactics is one way to receive trust information. A more elaborate approach is to integrate a third party trust management system to deliver this information based on the experience of other users with the service. The SOCIETIES Trust Management system, which has been

used to evaluate the PersoNISM system, provides trust values that range between 0 and 100 [167][168].

6.2.2 Acquiring Privacy Policy Negotiation Preferences

The PersoNISM system acquires PPN preferences in two ways; manually through a graphical user interface or by translating the user's input to the privacy policy negotiation process into IF-THEN-ELSE statements and then merging them with existing PPN preferences if they exist. For every data item requested in the agreement document, three sets of seven PPN preferences are created and/or updated. Each set has a total of seven PPN preferences, one for each condition present in each data request. The first set is created to reflect preferences for the specific service provider and service so that in the case of the user reinstalling this service in the future, the system can provide this information to the user to help them perform the negotiation and also remind them of the options they chose previously. Even though this set of PPN preferences are set to be used only for renegotiating with the same service provider for the same service, trust conditions are still embedded in the PPN preferences as it is possible that the trustworthiness of the service might change leading to the user re-examining the level of privacy they have previously negotiated with the service. The second set of PPN preferences is created to reflect preferences for the specific service provider but unrelated to the service with which the user negotiated. This is done to create preferences to guide the user in future negotiations with other services offered by the same service provider. The third set is needed to create or update a set of generic PPN preferences independently from the service and service provider the user negotiated with. These preferences are valuable information to use when the user negotiates with service providers that they have not interacted with before. The process of creating a PPN preference is shown in Figure 37.

Figure 37. *Acquiring PPN preferences sequence diagram.*

6.2.3 Applying Privacy Policy Negotiation Preferences

When the Request Policy is received, the Negotiation Client requests the evaluation of the PPN preferences that can be used to configure the RequestPolicy. The Privacy Preference Manager then tries to find a PPN preference for each condition in each data item. As explained in the previous section, PPN preferences can be related to either a specific service provided by a specific service provider or only a specific service provider or not related to a service at all (generic preferences). The preference retrieval algorithm will first search for a PPN preference that is related to the specific service. If it doesn't exist, it will then search for a PPN preference that is related to the specific service provider and if that doesn't exist, it will retrieve the generic PPN preference. After all the available PPN preferences have been retrieved, each preference is evaluated using the preference evaluation algorithm described in section 4.2.1. The privacy preference evaluation algorithm contains an additional evaluation functionality that allows trust conditions to be evaluated. The evaluation of trust conditions is slightly different to context conditions.

A PPN preference can contain multiple trust conditions that could all evaluate to true. For example, a PPN preference might contain two trust conditions such as in the preference shown below:

```
IF trust >= 60 then data_retention = "48 hours"
```

```
IF trust >= 70 THEN data_retention = "1 week".
```

If the trust management system reports that the current trust level of the service is above 70, then both statements evaluate to true. In this case, the algorithm will try to find the closest match between the trust conditions present in the preference. To perform this, during the preference tree traversal, any outcomes whose trust conditions evaluate to true are added to a temporary list so that they can be checked at the end of the algorithm. Then, if the list contains more than one item, the algorithm compares all the trust conditions to find the one closest to the current trust level of the service.

When the privacy preference evaluation algorithm terminates, the results are combined to create ResponseItem objects, one for each RequestItem object found in the service's privacy policy. These are returned to the NegotiationClient which now needs the user's input to continue. A graphical user interface such as the one shown in Figure 35 is used to show the terms and conditions of the service and allows the user to make changes. As shown in the figure, the GUI includes a button that provides the user with personalised suggestions. When the user clicks on the button, the parameters are changed according to the list of ResponseItem objects returned from the privacy preference evaluation process. Any items that were changed are marked with arrows as shown in Figure 35, to highlight to the user what has been changed. Clicking the reset button returns all the parameters back to the values that were originally requested by the service.

Privacy Policy Negotiation Form

i You are negotiating with www.hw.ac.uk. The information provided below presents the terms and conditions that will apply when you disclose data to www.hw.ac.uk. Configure usage per your needs and then click Continue to proceed with the negotiation.

Conditions for accessing: locationSymbolic

Purpose Your location is required to offer you services near you.

Your **location** will be kept by www.hw.ac.uk for: 30 minutes

Allow sharing of your **location** with : No sharing

Allow your **location** to be used to infer further information about you no

Conditions

Demand that your **location** be stored securely by www.hw.ac.uk yes

Demand the right to opt out of disclosing your **location** at any time yes

Demand access to view your **location** from www.hw.ac.uk yes

Demand to correct your **location** held by www.hw.ac.uk if incorrect yes

Get Personalised Suggestions Restore changes

Next >

Cancel Continue

Figure 38. *PersonISM Privacy Policy Negotiation GUI.*

Automating the privacy policy negotiation

After performing the evaluation process and before presenting the graphical user interface of the privacy policy negotiation to the user, the NegotiationClient checks to see if the evaluation outcomes can be used to perform the negotiation automatically on behalf of the user. If there is at least one valid evaluation outcome for every privacy condition of each data item requested, then there is enough information to perform it automatically. The user is asked to confirm that this is what they want. If the user accepts, then the Response Policy generation algorithm produces the Response Policy on behalf of the user using the PPN preference evaluation outcomes to configure the terms and conditions for each data item. The ResponsePolicy is sent to the NegotiationAgent continuing the negotiation process. When the NegotiationAgent returns the modified ResponsePolicy, the NegotiationClient compares it with the ResponsePolicy it sent on behalf of the user. If they match, the negotiation completes successfully. If they don't match, then the privacy policy negotiation form shows to the user the new terms and conditions offered by the service. As shown in Figure 39, the parameters are marked appropriately to highlight the ones that were rejected by the Negotiation Agent.

Privacy Policy Negotiation Form

i The terms and conditions you requested for the data items: birthday, email, name from the provider were not entirely acceptable. The provider has suggested alternatives. If you accept the alternatives provided, you can continue to install the service. Otherwise, the negotiation will fail.

Conditions for accessing: email

Purpose We require your email so we can contact you.

Conditions

Your email will be kept by www.hw.ac.uk for:	until account is deactivated	✓
Allow sharing of your email with :	No sharing	✓
Allow your email to be used to infer further information about you	no	✓
Demand that your email be stored securely by www.hw.ac.uk	yes	✓
Demand the right to opt out of disclosing your email at any time	no	⚠
Demand access to view your email from www.hw.ac.uk	yes	✓
Demand to correct your email held by www.hw.ac.uk if incorrect	yes	✓

Get Personalised Suggestions Restore changes

< Back Next >

Cancel Continue

Figure 39. *Service provider's ResponsePolicy presented to the user.*

6.3 *Personalisation in Identity Selection & Creation*

The purpose of personalising the identity selection is to help the user select the appropriate identity to represent themselves to services and in the case of identity creation, to help them associate the right data with the new identity thus presenting themselves to the service in a manner that satisfies their privacy demands. Identity Selection & Creation preferences can maintain information regarding the circumstances in which a user selected a specific identity to use or in which a specific data item was associated with a specific identity. The circumstances are defined by the identity of the service provider, their trust level, the current context of the user and the privacy conditions that were agreed during the Privacy Policy Negotiation.

6.3.1 **Identity Selection Preferences**

The Identity Selection (IdS) Preferences define under what circumstances an identity should be selected for use. As in the case of the PPN preferences, IdS preferences are designed to service the specific requirements of identity selection and therefore the preference outcome defines the identity identifier. The IdS preference conditions describe the conditions in which the identity (defined in the outcome) should be used. IdS

Preference Conditions can be context conditions and trust conditions. Identity Selection Preferences are context dependent because the use of an identity can depend on the context of the user, such as their location, time and current activity. For example, a user might use an identity for work related activities and another for leisure related activities. To help the user manage their identities and select the right identity to use in a situation, the preference must be able to maintain this information and evaluate it accordingly. The trustworthiness of the service also affects the identity selection process because different identities are associated with different data. The user can maintain a set of identities to use with services with a trust level below a certain threshold. Finally, the privacy conditions that are derived by examining the Negotiation Agreement document, can also determine which identity to use in a specific situation. Similarly with trust conditions, a set of negotiated privacy conditions that denote lower settings than the user would have preferred can determine that an identity should not be used and another should. Optimally, to protect their privacy users would create two or more identities to use with certain services in order to maintain some separation between some sensitive information. For example, separating work related information from home related information. As the number of installed services grows, so will the number of identities. Managing a large number of identities and remembering which identity should be used in a specific context can be a very cumbersome exercise for the user. The Identity Selection preferences are designed to help the user with this task.

6.3.2 Attribute Selection Preferences

The Attribute Selection (AttrSel) preferences define which data item should be associated with a new identity in a specific situation when a new identity is created for a specific purpose (i.e. to use a service). Attribute Selection Preferences cannot be used for creating a new identity manually as there is no information about the purpose for creating the identity. The association of a data item with an identity can be influenced by the service's trustworthiness level as well as the privacy conditions agreed during the privacy policy negotiation. User context information such as the current location, activity or time do not affect the attribute selection process as the identity associations are final and exist as long as the identity exists. So, the only two types of preference conditions that can be used for Identity Selection Preferences are privacy conditions and trust conditions. A Privacy Condition is a conditional statement that compares the value of a Privacy Condition such as data retention, sharing with third parties, etc. (as described in section 6.2.1) with the

current value of the same privacy condition type present in the ResponsePolicy. The Attribute Selection Preference Outcome defines which data item should be used under the set of conditions. The data item is defined by its specific context identifier assigned to it by the Context Management system. For example, the context database can contain multiple data attributes of type “email” as users tend to have more than one email address that is used for different purposes. Each different attribute has its own identifier to allow it to be addressed. Therefore, the AttrSel preference outcome contains only the identifier of the data attribute.

6.3.3 Acquiring Identity Selection preferences

The PersoNISM system acquires Identity Selection preferences at the end of the privacy policy negotiation. Using the Negotiation Agreement as input, the Privacy Preference Management system extracts the identities of the user, the service provider and the service identifier. It first constructs the preference outcome using the user’s identity. Then, it creates two trust conditions, one for the service provider and one for the service itself. Then it retrieves a snapshot of the current context of the user.

By default, the current snapshot of the current context includes the following information: the user’s location, the time of day, the day of the week and the user’s activity. Through its exposed API (Application Program Interface), the PersoNISM system allows the manipulation of this set if the system is enhanced with sensors that provide additional contextual information about the user and this information can affect the selection of identities. This functionality allows the system to behave in a dynamic manner, appropriate to the nature of pervasive systems.

Each retrieved data item is translated into a context preference condition. All conditions (both trust and context) are then loaded into a preference tree structure as branches (example shown in Figure 40) with the preference outcome added as the leaf of the tree.

This preference now defines one of the circumstances under which this identity may be used. Three copies of this preference are used. One is associated with the selected identity, the specific service and service provider, one with the selected identity and the service provider only and one with only the selected identity (the generic preference). For each of these, the Preference Management component searches for existing preferences. If a

preference exists, then it is merged with the newly created preference and is stored. Again, the same merging algorithm is used as the one described in section 4.3.1.

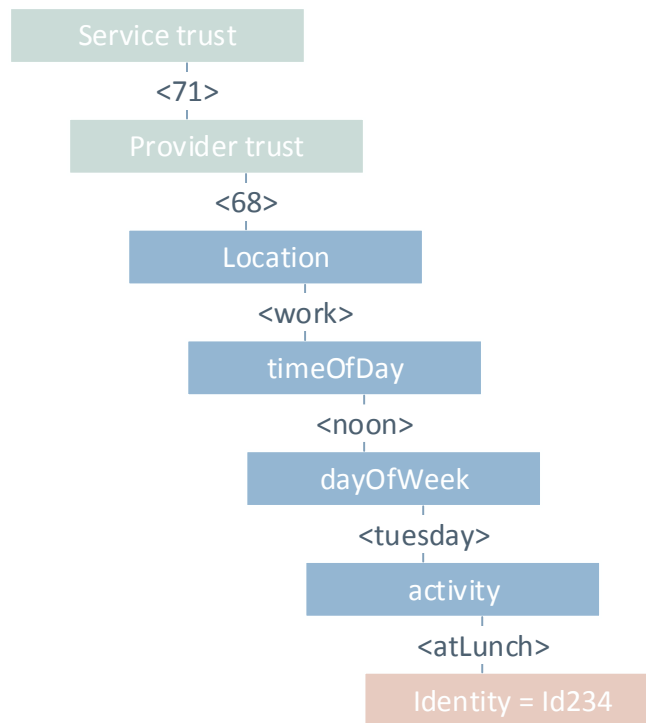


Figure 40. *Constructed IdS Preference tree.*

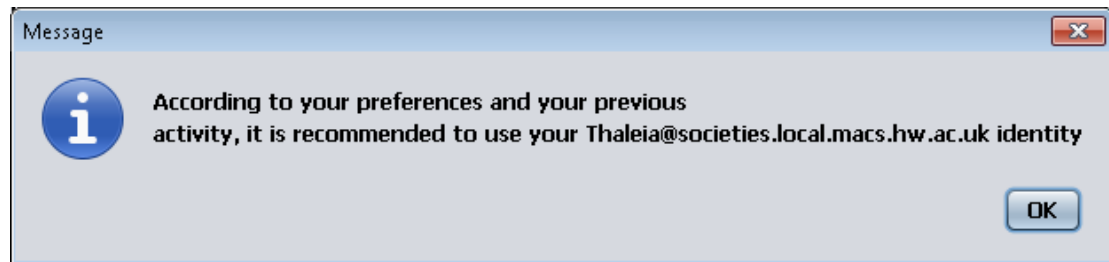
6.3.4 Acquiring Attribute Selection preferences

During the identity selection stage, the user might choose not to use any of their existing identities, or if the existing identities cannot satisfy the requirements of the ResponsePolicy, a new identity needs to be created. As explained previously, the profile of the user can contain any number of data items of the same type (such as multiple email addresses or postal addresses). The identity creation process involves picking the appropriate data item to associate with the new identity. When the user performs their selection and the new identity is finalised, the Preference Management component collects all the required information in order to create the corresponding Attribute Selection Preferences. Three sets of N number of preferences are created where N is the number of data items in the ResponsePolicy. The first set is associated with the specific service, specific service provider and data type; the second set is associated with the specific service provider and data type and the last set is associated only with the data type. The trust level of the service and service provider are retrieved and translated into trust conditions. Then, for each data type, the list of privacy conditions defined in the corresponding ResponseItem are retrieved and translated into seven privacy preference

conditions and the PreferenceOutcome is constructed from the identifier of the attribute that was selected for providing the actual information to the service. Using the same procedure as in the previous section, the preference tree is constructed with the trust and privacy conditions placed as branches of the tree and the preference outcome as the leaf. Preferences that already exist in the system are retrieved and merged with the newly created preference tree using the merging algorithm described in section 4.3.1.

6.3.5 Applying Identity Selection Preferences

Identity Selection preferences are evaluated to produce an identity that should be selected for use with a specific service in a specific situation. If the user has requested that the privacy policy negotiation be performed on their behalf automatically (as described in section 6.2.3), Identity Selection preferences are evaluated in order to perform automatic Identity Selection. If the user has selected manual configuration, the Identity Selection preferences are only evaluated upon request. When the user is prompted to select an identity, the graphical user interface provides a button (shown in Figure 42) that allows the user to request the aid of the system in selecting the identity. When that happens, the preference evaluation algorithm is called to evaluate all the relevant IdS preferences and suggest one or more identities to the user. The current trust level of the service and service provider and the current context of the user are retrieved to be used in the preference evaluation process. For each existing identity that can satisfy the requirements of the ResponsePolicy, the corresponding preferences are retrieved. The algorithm begins to evaluate the most specific preference first; the one associated with the identity, the service provider and the service. If that preference doesn't produce a result, it then evaluates the second most specific preference; that is the one associated with the identity and service provider and if that doesn't produce a result, the most generic preference is evaluated. If the preference evaluation algorithm produces one identity to be used, it is selected and the service is started with that identity. If there is more than one identity that can be suggested, the user is prompted to confirm which identity they wish to use. If the evaluation algorithm does not produce a result, the user has to be prompted to create a new identity.

Figure 41. *Identity Selection*Figure 42. *Personalised Identity Selection.*

Proactive context-aware Identity Selection

By monitoring the user's active sessions, the PersonISM system is constantly aware of the identities that are being used at any time. Using the algorithm described in section 4.2.2, a change in the values of the context and trust conditions that affect the evaluation of the Identity Selection preferences of the active identities triggers their re-evaluation. If the evaluation algorithm results in a suggestion to use a different identity, the user is alerted that their privacy may be compromised and is advised to take appropriate course of action such as switching to the identity recommended in the evaluation result.

6.3.6 Applying Attribute Selection Preferences

The Identity Creation process involves the association of a list of data items with a new identity. The user decides what information to associate with an identity so that this information is then disclosed to the service. Attribute selection preferences can be used to aid the user in performing this task faster and more efficiently. As shown in Figure 43,

the user can request help from the PersonISM system by clicking the button “Get recommended attributes”.

Identity name:

Requested data (Click to see stored values):

- birthday
- email
- name
- locationSymbolic

Actual data stored in your profile

email : et90@hw.ac.uk
 email : JohnSmith323@gmail.com
 email : bittersweetsymphony@hotmail.com

Add selected attribute

Add new attribute of this type

Get recommended attributes

Linked data to this identity

Data Type	Current value	ID
email	bittersweetsymphony@hotmail...	context://university.societies.loc...

Remove selected

Reset data links

Cancel

OK >>

Figure 43. Identity Creation form.

The evaluation algorithm will be triggered to evaluate all the preferences that are related to the data types that need to be associated with this identity, the service provider and service. Again, the order of evaluating the retrieved preferences begins from the most specific preference; the one that is associated with the data type, the service provider and the service and if that fails to produce a result, it moves to the preference associated with the data type and service provider only and so on. If all three preferences are evaluated without producing a result, then the system cannot recommend a specific attribute. The

results of the evaluation are collected and presented to the user for approval as shown in Figure 44.

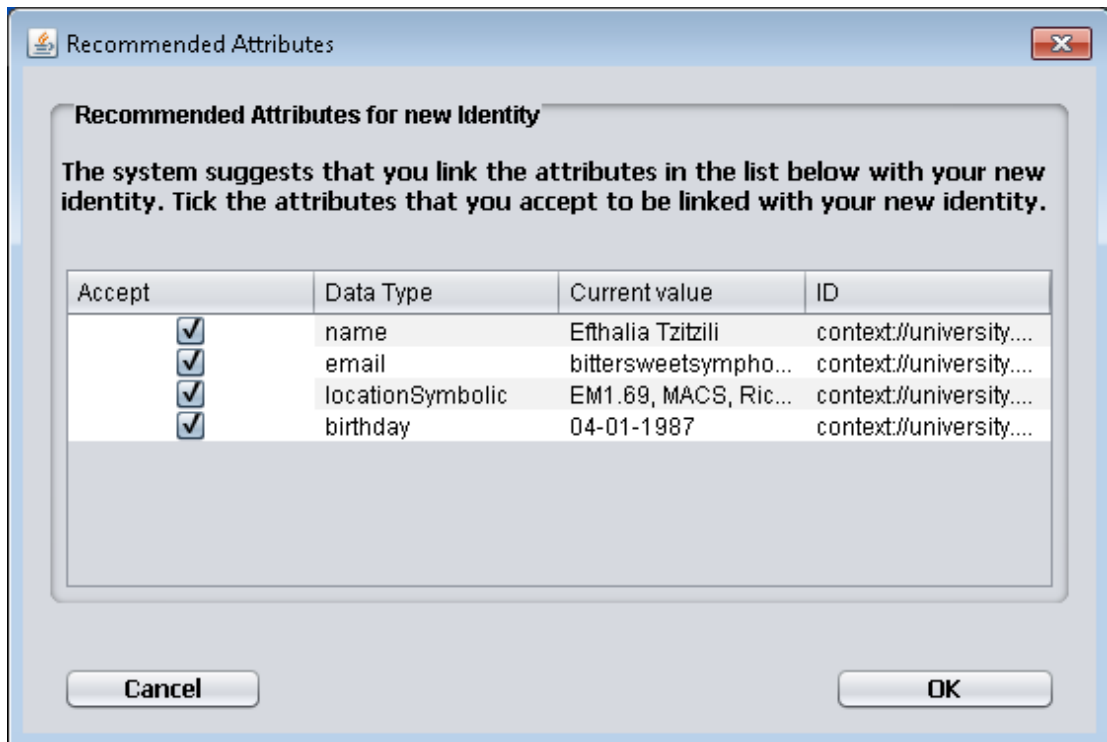


Figure 44. *Recommended attributes list for new identity.*

6.4 *Personalisation in Access Control*

The Privacy Policy Negotiation and Identity Selection stages happen before any data is disclosed to a service. At this stage, the system has prepared the environment in which the data will be disclosed, defining the rules that the service will adhere to after it receives the user data. After this stage finishes, the PersonISM system facilitates the actual data disclosure according to the wishes of the user. The PersonISM system maintains a set of access control rules that declare who has access to perform a specific action on a resource. These rules are managed in a dynamic manner. The access control rules change by constantly monitoring the current context of the user that affects the decision of granting permission to access a data resource. The PersonISM system utilises a set of access control preferences that are evaluated to perform this dynamic management of access control rules. In addition to access control functionality, data obfuscation is performed in some cases to alter the quality of the information that is disclosed to services. To determine the level of obfuscation that should be applied to each data item, another type of privacy preference is used; the data obfuscation preference.

6.4.1 Access Control and Data Obfuscation Preferences

Access control preferences state if a requestor (i.e. a service) is allowed to perform an action (read, write, create, delete) on a data item (specified by its context identifier) in a specific situation. The trust level of the service and service provider combined with current context information affect the decision whether to allow a service access to a data item. Therefore both trust and context preference conditions are used in Access Control preferences. The preference outcome of an access control preference defines an “allow” or “block” permission. Access control preferences have the following preference hierarchy:

- a) The specific AccCtrl preference is associated with the data identifier, the action to be performed (read, write, create, delete), the service and service provider information.
- b) The generic AccCtrl preference is associated with the data identifier and the action to be performed.

Data obfuscation (DObf) preferences state the level of obfuscation that should be applied to a specific data item in a specific situation. The data obfuscation algorithm of a data type defines the number of obfuscation levels that can be applied. The DObf preference outcome states the obfuscation level that should be applied in numeric form, the bounds of which are defined by the algorithm.

For example, the DObf preference

```
IF symbolic_location== “home” AND trust>=51 THEN obfLevel = 4.
```

defines data obfuscation level four (4) should be applied to the data attribute of type <symbolic_location> when the current value of that data attribute is <home>.

6.4.2 Acquiring Access Control and Data Obfuscation Preferences

All preferences are created by collecting information about the user’s interactions with the system and then translating that behaviour into a preference tree. Likewise, access control and data obfuscation preferences are created by monitoring the user’s decisions about data disclosure when they are prompted with a graphical user interface as shown later in Figure 45.

Static profile information such as the user's name, postal address changes rarely or not at all and when it is disclosed at least once, it has little consequence if it is disclosed again to the same service in a different situation. For static context information, only trust conditions are used (not context) in constructing the corresponding access control or data obfuscation preferences, as the disclosure and obfuscation of those data values does not depend on the current context of the user. For these data types, the trustworthiness of the service can affect the decision to block access or obfuscate the value of the data.



Figure 45. *Access Control and Data Obfuscation requests.*

As shown in Figure 45, a service requests access to the user's symbolic location and the user is prompted to allow or block the service from accessing the data and to indicate how they want to obfuscate their symbolic location before disclosing it to the service. When the value of the slider bar changes, the example value provided below it (showing "EH14 4AS, Edinburgh, Scotland, UK" in Figure 45) changes to reflect the obfuscated value of the chosen level. After receiving the user's input, the Privacy Preference Management component translates it into access control and data obfuscation preferences respectively.

In the case of the access control notification, depending on the user's input, the Access Control Preference Outcome is created to indicate an "Allow" or "Block" decision (e.g. effect = "Allow"). Similarly, the Data Obfuscation Preference Outcome is constructed using the selected level of obfuscation (i.e. obfLevel = 3). Trust conditions are created by retrieving the trust value of the service and service provider from the Trust Management component. The only context condition that is created refers to the current value of the data item that was requested. In the example of Figure 45, the data item is of type symbolic location and the value that is used to construct the context condition is the non-obfuscated value of the data item. The context condition is created only if the data item is contextual information and not static profile information (i.e. location information as opposed to an email or postal address).

As the user interacts with their environment, their location and other information changes and the service listening to these changes tries to retrieve the updated information. This triggers the access control and data obfuscation notifications to appear and request the user's input. Every time the user is involved in a decision, the PersonISM system collects the information and updates the access control and DObf preferences trees appropriately. Existing preferences associated to the same data are retrieved and are combined with the newly created preference trees using the preference merging algorithm.

6.4.3 Applying Access Control and Data Obfuscation Preferences

Access Control and Data Obfuscation preferences are applied when a service requests access to a data item. Using the preference hierarchy described in section 6.4.1, the appropriate preferences are retrieved and evaluated against the current trust level of the service requesting the data, and if applicable, the current context of the user (if a context condition exists in the preference tree). When the preference evaluation algorithm produces a preference outcome, its confidence level defines how this outcome will be implemented. If the confidence level is high (by default 7 but these thresholds can be changed by the user), then an access control notification is presented to the user with a countdown timer of fifteen seconds (this can also be modified by the user). The notification informs the user that their preferences recommend this permission and offer the user the option to change it. If the countdown timer reaches zero without being interrupted by the user, the outcome will be implemented. If the user intervenes and changes the outcome, this information is fed back to the preference merging algorithm to

adjust the confidence level of the preference outcome appropriately using the confidence level calculation algorithm described in section 4.3.2. The countdown timer is a parameter which can be modified if the user needs more time to react to these notifications. If the confidence level is between 5 and 7, the notification requests the explicit input of the user while informing them what the preference evaluation algorithm suggested. Again, the user's input is delivered to the preference merging algorithm to update the existing preferences and their confidence level. Finally, if the confidence level is below 50, the notification appears without a suggestion of action as the system needs the user to make the decision without being offered a wrong suggestion. The same approach is implemented in the case of Data Obfuscation preferences.

Condition Monitored Access Control

In a pervasive service environment where context information changes frequently, access control must be performed in a proactive manner. The trust and context conditions that affect the access control preferences must be monitored constantly and, when a change occurs, a re-evaluation of those preferences should be triggered. In an event driven platform, services can listen for changes to the contextual information of the user in order to continuously update the information they are using. In the PersoNISM system, the Privacy Preference Management component performs the re-evaluation of the preferences that are affected by these preferences before any updated data is delivered to the services. This is the same process as the one used for personalising services in a proactive manner (described in section 4.2.2).

6.5 *Preference Learning for PersoNISM*

The PersoNISM system uses the Preference Merging algorithm as described in section 4.3.1 with slight differences. The major difference between learning user preferences for personalisation, and learning privacy preferences for PersoNISM is that PersoNISM does not make use of the C45 learning algorithm. Instead, information that is captured during the user's interaction with a) the PPN GUI, b) the identity selection or creation GUIs and c) the access control and data obfuscation GUIs is translated directly into the corresponding privacy preference format and merged with the existing preference of the same type using the Preference Merging algorithm directly as described in sections 6.2.2, 6.3.3, 6.3.4, and 6.4.2.

The algorithm as described in section 4.3.1 differs in the following situations:

Situation 2 uses option d) which states that both actions are kept in the system. Option d) was preferred in the case of the PersoNISM system as it does not maintain a user behaviour history that it mines to produce privacy preferences. Instead, the input from the user is fed directly into the merging algorithm that merges the input with the existing preference. Partly, this solution allows that a form of behaviour history is actually stored as a preference tree. It also gives the user more options when they edit the preference manually and they can see the different decisions they have made in the past and the corresponding confidence level shows them how the system treats each of the actions.

Situations 4.1 and 4.3 use option a) as defined under situation 4.1. The same reasoning is used as in situation 2.

6.6 *Summary*

The PersoNISM system is an approach to privacy protection that utilises a set of personalisation and behaviour learning techniques. It is designed to help users obtain better privacy and handling of their data. The privacy policy negotiation process offers users an alternative to the “take it or leave it” approach where users agree to a set of terms and conditions that are dictated by the service provider. Demanding the right privacy that fits the needs of the individual user is the first goal of the PersoNISM system. The second goal is to use sophisticated user behaviour learning and personalisation tools in order to help the user maintain the privacy they desire in a user friendly manner. By providing such a system, we can emphasize the importance of privacy protection to the user. However, the privacy protection that the PersoNISM system provides is directly affected by the user’s decisions.

The PersoNISM system maintains a set of Privacy Policy Negotiation preferences that are constantly updated by collecting information from the user’s input. PPN preferences can be used to recommend to the user terms and conditions similar to the ones they have selected in previous negotiations. PPN preferences can also be used to automate the process of privacy policy negotiation if the user wishes to do so.

Identity Selection and Creation preferences are maintained to guide the user through choosing the right identity to use in a specific situation to represent themselves to services,

and configure these identities by allowing the user to keep certain information separate from other information.

Access Control preferences dictate if a service has permission to access a resource in some context and data obfuscation preferences define the quality of the information that will be accessed.

The PersonISM system provides different levels of automation depending on the degree of trust that the user has in the system. Users may choose how often they prefer to be prompted. However, in order to reassure users they are always in control of their data, the system notifies them when a data is about to be disclosed.

7 PersoNISM Evaluation

The evaluation of the PersoNISM system was conducted in two phases. In both cases, users were involved in the evaluation. In the first case, a questionnaire was published online, advertised on social media to attract as many respondents as possible. Users were asked to answer questions about their online data disclosure habits. Through a series of questions and scenarios, users were asked how much they know about what happens to their data after they are disclosed, what tools they use to protect their privacy and what they think about the consequences of data disclosure. The aim of the questionnaire was to assess the level of awareness of privacy of the average user and to assess the necessity for a system such as PersoNISM to be available.

It was predicted from early on that it would be difficult to attract a large number of people to participate in a “hands on” approach to evaluating the PersoNISM system, therefore, the online questionnaire was partially done to collect answers from a wider audience.

The second phase of the evaluation involved a smaller user group as the experiment was conducted face to face in one hour sessions for each user. Each user followed a set of instructions on how to use the PersoNISM system in order to experience its main functionalities and answered a set of questions in each step of the evaluation process.

The evaluation of the PersoNISM system aimed at, first, establishing that there is a real need for changes in the way that privacy is handled currently by pinpointing the current privacy practices of digital services and further, evaluating the concept and usefulness of the PersoNISM system by exposing users to its functionalities and subtle personalisation techniques.

7.1 *Online Questionnaire on Privacy*

Subject Group

This section presents the results of the questionnaire conducted in March 2014. The questionnaire was advertised on LinkedIn and Facebook and attracted 185 respondents of various ages as shown in Figure 46 most of whom have received a university degree.

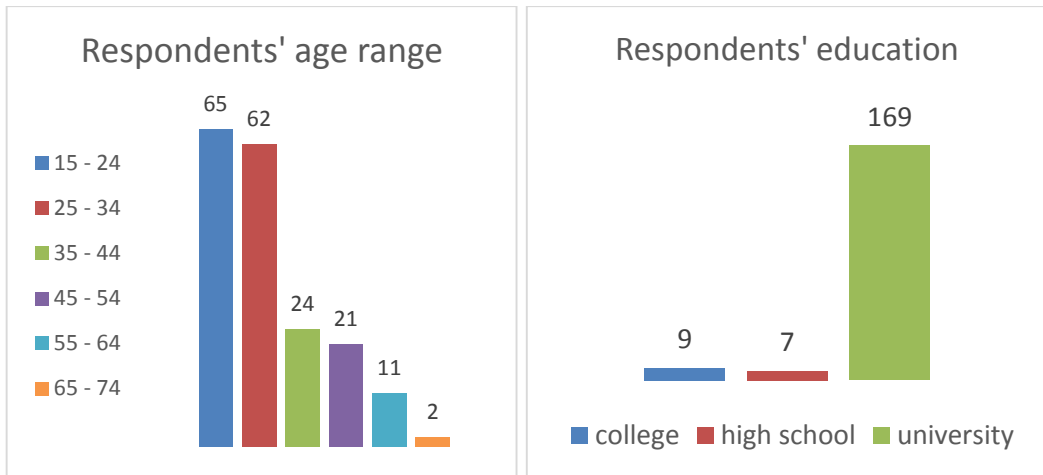


Figure 46. Questionnaire respondents' ages and education.

The respondents' were assessed in terms of how privacy aware they perceive themselves. As shown in Figure 44, 44.3% of respondents declare that they try to protect their privacy as much as they can while 34.6% only do so when the software provides explicit tools to do so. Notably, 20.5% declare that they are not worried about their privacy while a very small percentage (0.5%) declare that they do not care about their personal data.

Finding 1: 78.9% (44.3% + 34.6%) state that they want to protect their personal data. Hence there is a clear need to design a tool to help them protect personal information.

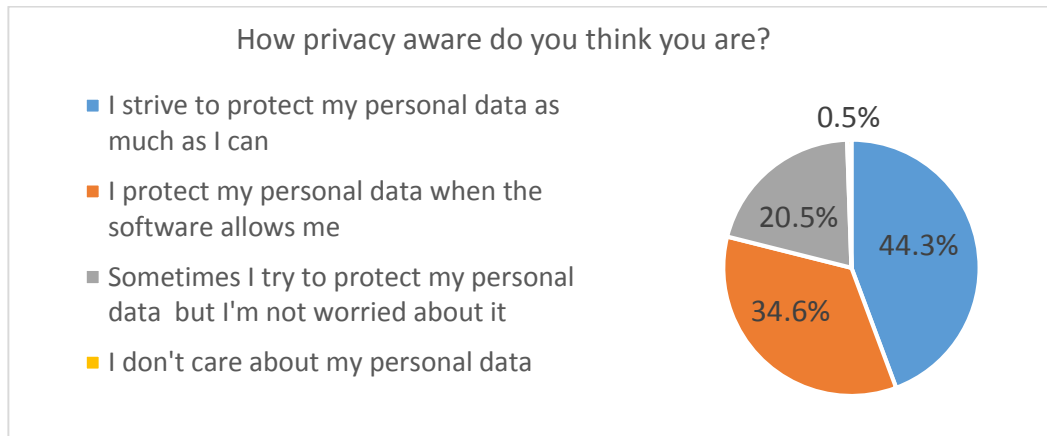


Figure 47. Privacy awareness perception of respondents.

Respondents' Social media use

Social networking sites (SNSs) promote the disclosure of personal information. Figure 48, Figure 49 and Figure 50 show the subject group's use of SNSs, the frequency with which they post information on them and what type of information they typically disclose on them. The questions were asked in order to determine the degree to which the

respondents use SNSs and disclose personal information on them. Since pervasive systems are not common-place at the moment, most users would be unable to comprehend the enormity of data disclosure in such an environment. SNSs provide a platform on which users are used to disclosing data.

As shown in Figure 45, at most only 2.7% of respondents do not use social networking sites of any kind, which shows that the subject group is largely familiar with disclosure of personal information.

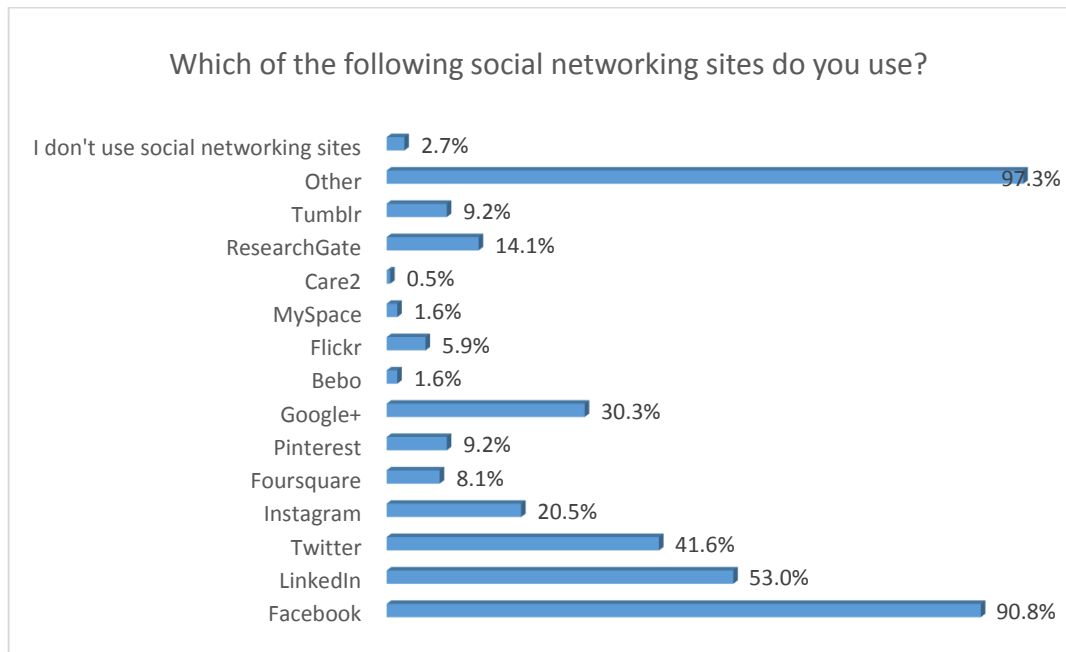


Figure 48. Respondents' use of social media.

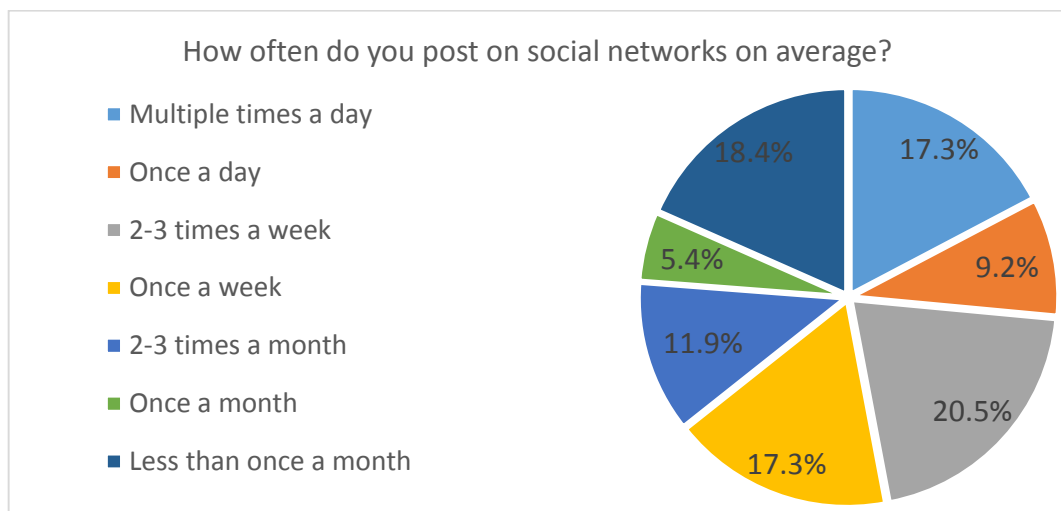


Figure 49. Respondents' SNS frequency of usage

Finding 2: As shown in Figure 50, users disclose a lot of personal information on SNSs and they constantly update that information.

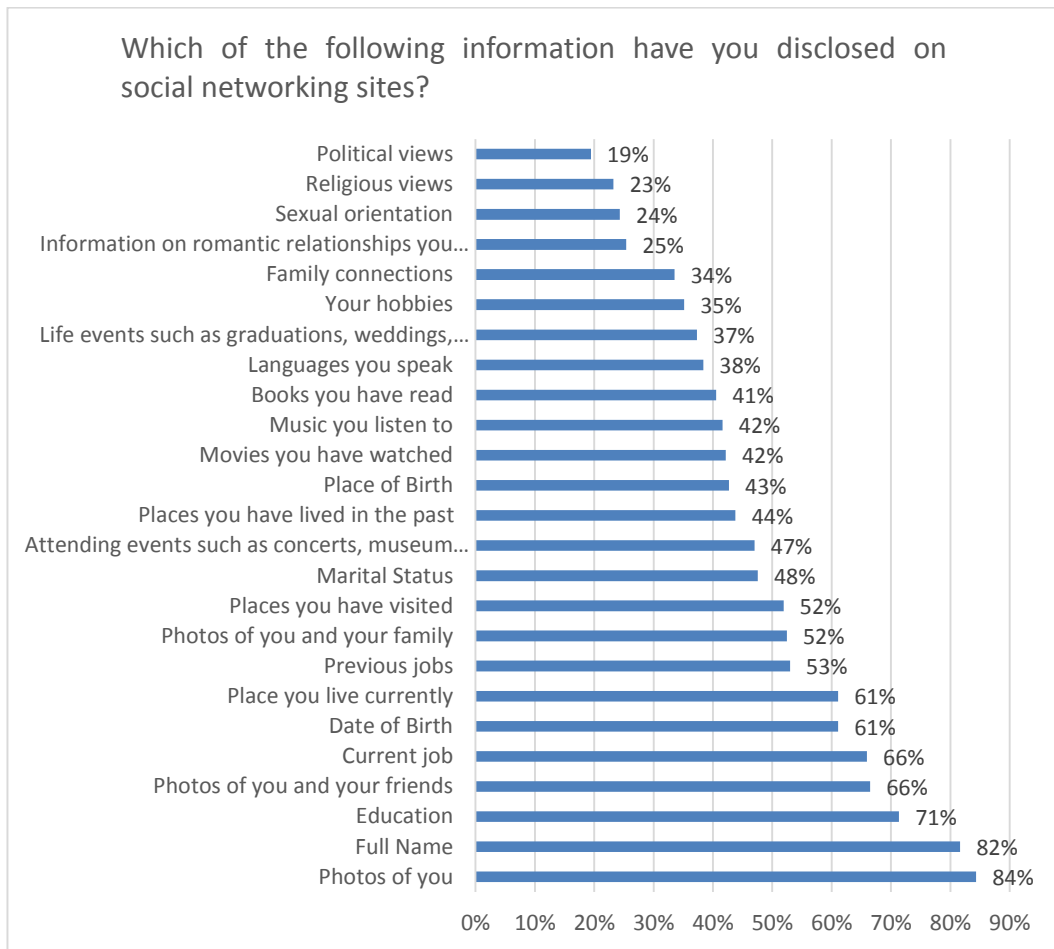


Figure 50. *Range of personal information disclosed on SNS.*

Respondents’ Awareness of Privacy in Social media

There has been a lot of attention on the issue of privacy in Social Networking Sites due to the vast amounts of personal information being disclosed voluntarily by users, many of whom are not aware of potential risks in disclosing so much information. Figure 51 shows the level of awareness of users in what SNSs do with the information they accumulate. It is noteworthy that 9.1% (3.2% + 5.9%) of subjects state that they do not care about what happens to their data. The majority of subjects (49.2%) state that they have limited knowledge of what SNSs do with their data and that worries them but they feel they have no way to protect their information except to not use SNSs.

Finding 3: 82.8% (19.5% + 49.2% + 14.1)) of respondents state their only option is to not use social networking sites.

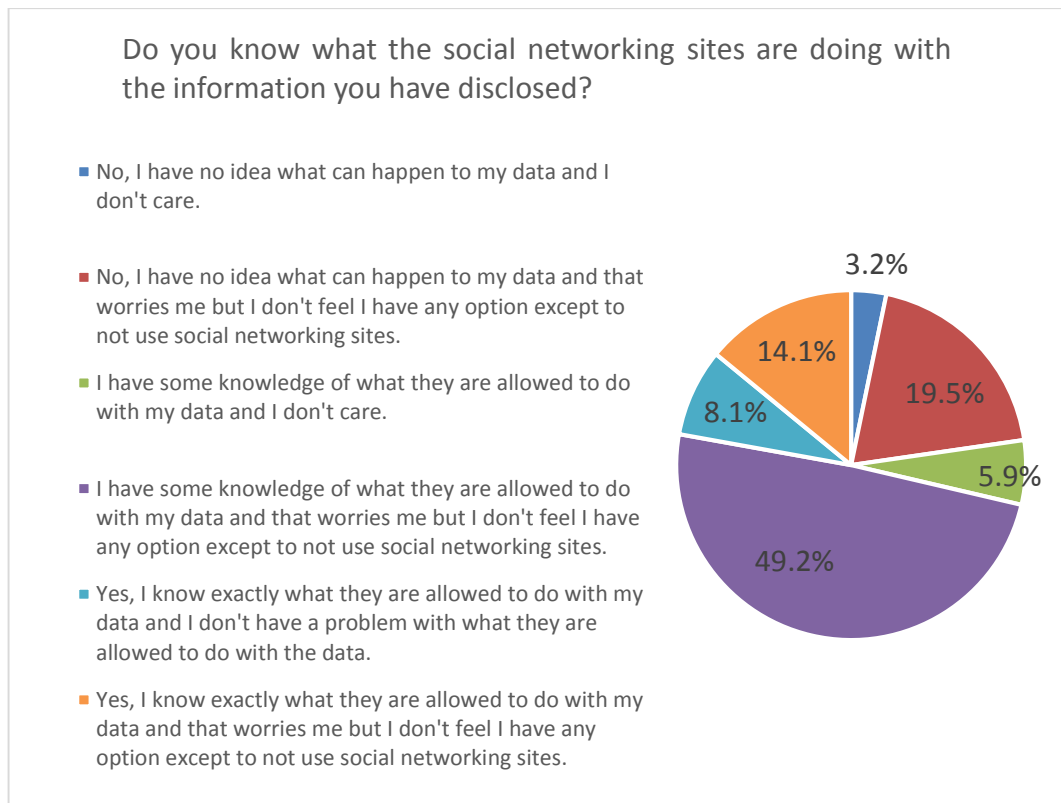


Figure 51. *Privacy Awareness in Social Media.*

Privacy practices are described in privacy policies and terms and conditions (T&C) documents and presented to users before they use a service. It is important to find out how many users actually read these documents and how well they read them and understand them and discover the reasons for not reading them. As shown in Figure 52, 21.1% of respondents state that they always read the privacy policy and T&C documents but only 15.1% state they understand them. 33% state they don't understand them regardless of whether they read them or not. 23.8% of respondents believe that service providers don't abide by their privacy policies. There is a need for service providers to provide proper mechanisms to show that data disclosure and processing is performed according to the privacy policies they publish. 12.4% of respondents don't read privacy policies and T&C documents because they are too long. A worryingly high percentage of 64.3% of respondents state that they never read privacy policies and T&C documents. There should be a mechanism to present the information in a concise form that users can understand easily and quickly to be able to provide informed consent about disclosing their personal data.

Finding 4: The majority of users don't read the privacy policies of the services they use.

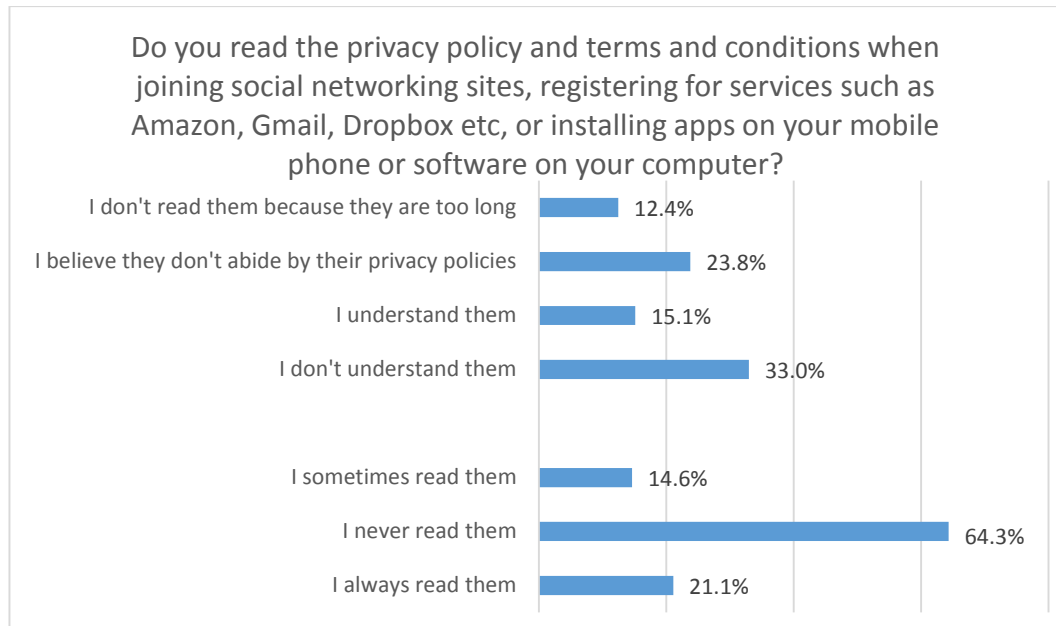


Figure 52. *Reading and Understanding Privacy Policies.*

The majority of SNSs provide privacy settings where users can define which SNS members can see what they have disclosed in the SNS. It should be noted that the privacy settings only give users the ability to hide information from other users but do not give them any way to stop SNSs from sharing information with other companies for advertising purposes.

As shown in Figure 53, 90.7% of the subject group have configured their privacy using privacy settings tools provided by SNSs but 29.7% of those state that they are not satisfied with it. The most common reasons for the users' dissatisfaction are:

- a) 18.9% of respondents commented that the privacy settings do not provide comprehensive privacy protection. The functionality is limited and there are data they would like to hide but they cannot.
- b) 12.4% of respondents commented that the settings are not user friendly and that some settings they applied did not work as expected.
- c) 7.8% of respondents commented that the settings affect the visibility of the data inside the social networking site but not about what the SNS does with that data.

61% of respondents state that they are satisfied with the privacy tools provided by the SNSs they use.

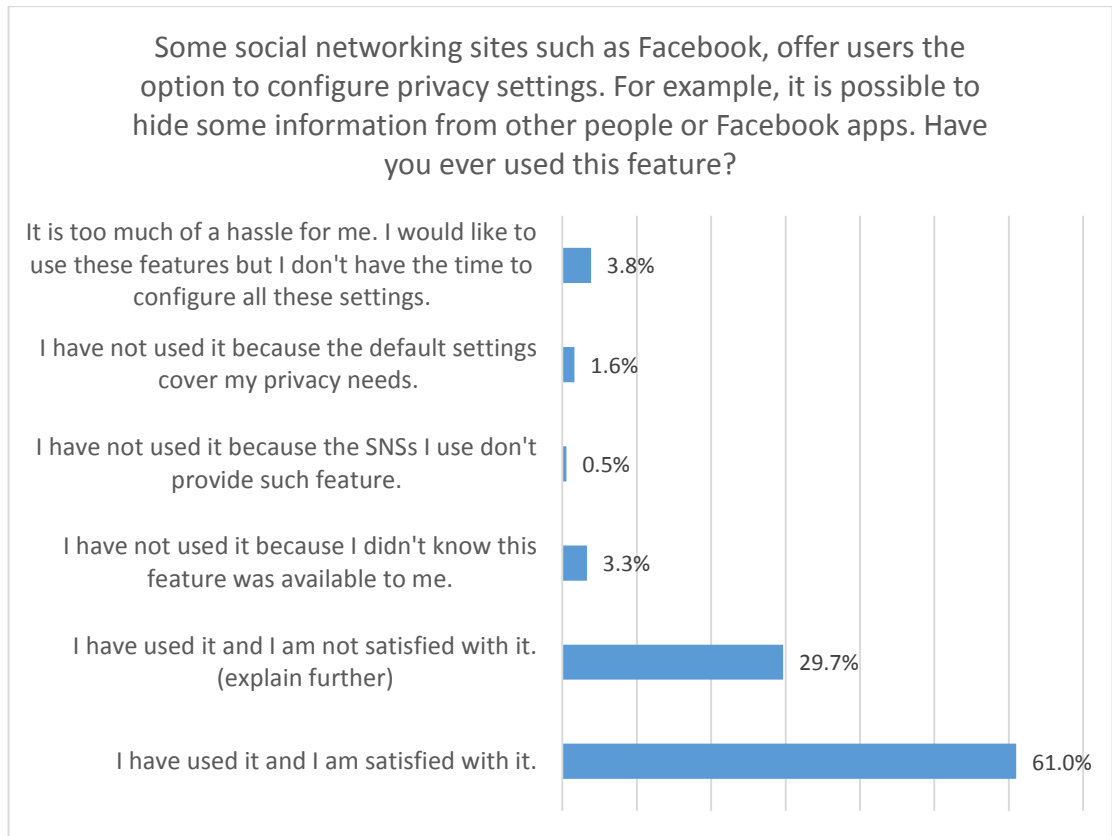


Figure 53. *Privacy tools in SNSs.*

68.1% of respondents state that they have aborted the installation or signing up for a service because they disagreed with the service’s privacy policy. Allowing users to configure the privacy policy to match their preferred privacy settings as much as the functionality of the service allows, could attract more users to that service.

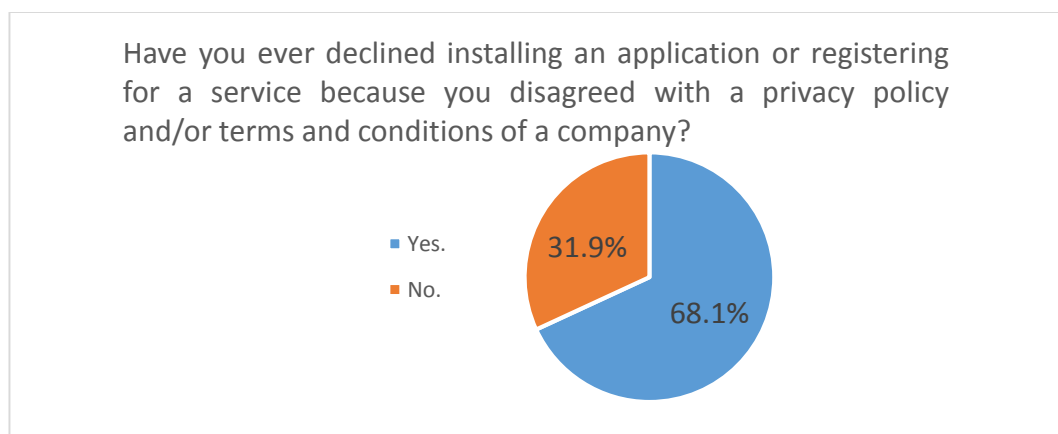


Figure 54. *Acceptance of privacy policy.*

Respondents were asked if they would use a privacy policy negotiation tool such as the user interface shown in Figure 55. 87% of respondents stated that they would use such a

tool. 5.9% of respondents did not understand what the tool was asking of them. It is necessary to provide enough information and guidance to use such a feature properly.

A small percentage (7%) stated they would not use such a tool listing the following reasons:

- Not seeing the difference between current privacy terms and what the tool provides.
- Not trusting the service providers to abide by the rules hence they don't see any value in using it.
- The tool looks too complex.
- Some of the settings are not clear (specifically the actions).

Privacy Policy Negotiation with Edinburgh Council

The list of data required by Edinburgh council for using parking services are outlined below. Configure the terms and conditions as you wish and click Continue.

- ▶ [name](#)
- ▶ [age](#)
- ▼ [GPS location](#)
 - Purpose: Your location will be tracked to offer you services nearby.
 - Actions:
 - Read
 - Write
 - Create
 - Delete
 - Conditions:
 - Share with 3rd parties Keep data for 1 week
 - Right to opt out
 - Decision: allow deny
- ▶ [activity](#)

Cancel Continue

Figure 55. Privacy Policy Negotiation Tool User Interface

Finding 5: The vast majority of users (87%) stated they would use a privacy policy negotiation tool.

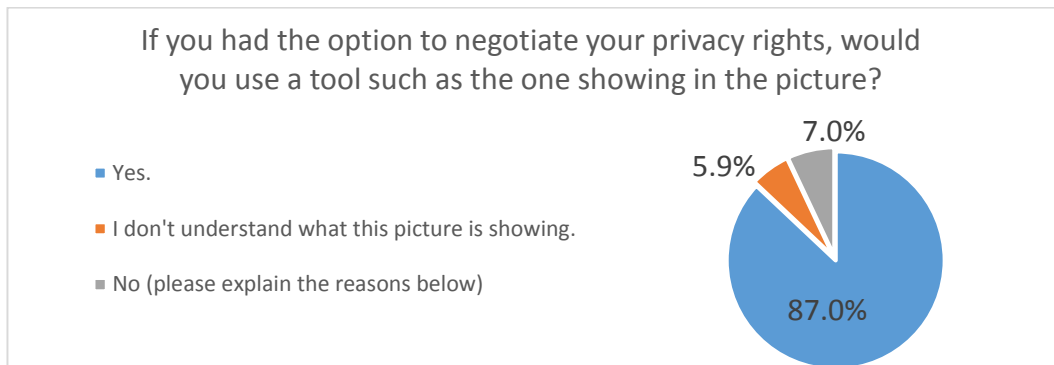


Figure 56. *User Customisation of privacy policies.*

25.4% of respondents were not aware that contextual information as described in the following paragraph can exist in a digital form and be used by web services. The following definition of user context was given to users:

Context information is information that describes the environment and yourself. This includes current location (GPS coordinates), symbolic locations (home, work, gym, pub etc), activity (sleeping, walking, working, exercising, watching television etc), age, room temperature, light conditions, current weather. Most of this information is currently available through sensors and personal mobile devices.

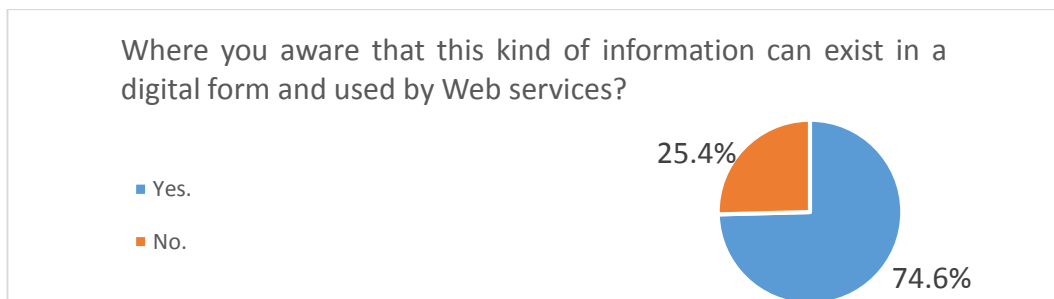


Figure 57. *Awareness of information that can be represented in digital form.*

As shown in Figure 58, 8.6% of users do not care if mobile phone or SNS apps have access to this information while 75.1% of them state that they do. However, as shown in Figure 59, only 3.2% would not want to restrict access to this information. The last 16.2% of respondents stated that it depends on what the apps would use the information for. Specifically, the following were the most common statements:

- There has to be a clear benefit from the app for disclosing information.

- The disclosed information has to be clearly relevant (or its relevance explained) to the application that requests it.
- The information is not to be used for advertising.
- The information must only be disclosed when the user wants it and not always.

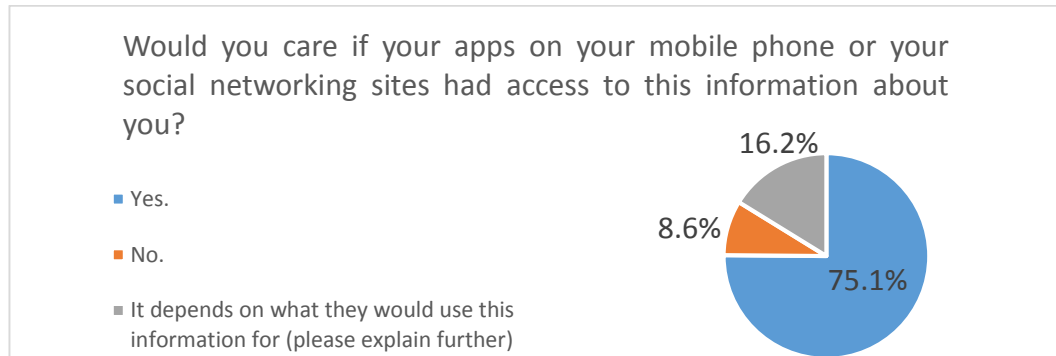


Figure 58. *Context information usage.*

The following reasons were given by those who answered no to restricting access to contextual information (Figure 59):

- Contextual information make navigation through webpages and apps easier.
- Not seeing the harm in disclosing contextual information.
- Having nothing to hide and feeling that such information disclosure works to the user's advantage.
- Making the apps more functional (and hence more useful).
- Not considering contextual information as personal information.

Finding 6: Almost all respondents (96.8%) state they would want to restrict access to contextual information made available through their smart phones.

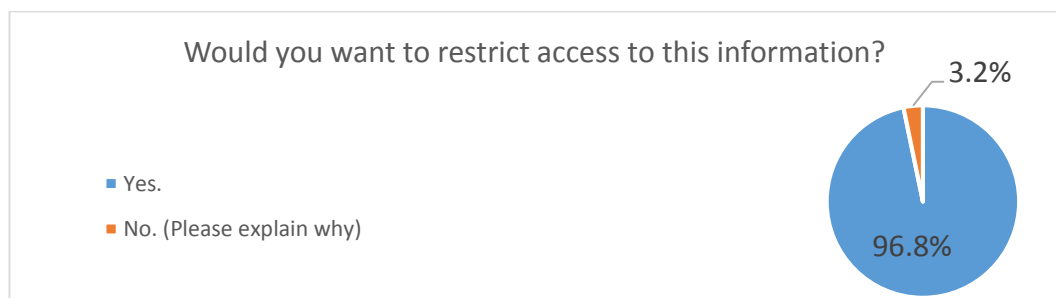


Figure 59. *Restricting access to information.*

As shown in Figure 60, the desirable frequency for prompting users with questions about disclosing data varies a lot between users. 39.5% - the highest percentage – state that they want to be prompted every time. However, 32.4% state they want to be asked depending on their context and 26.5% of respondents state they only want to be asked once for each type of information. Finally, a very small percentage (1.6%) stated they don't want to be asked at all. It can safely be deduced that the system should provide appropriate mechanisms to customise the frequency of prompting users for privacy permissions to the user's preference.

Finding 7: Users want to be prompted to permit or deny access to contextual information but they should be able to customise when they are prompted.

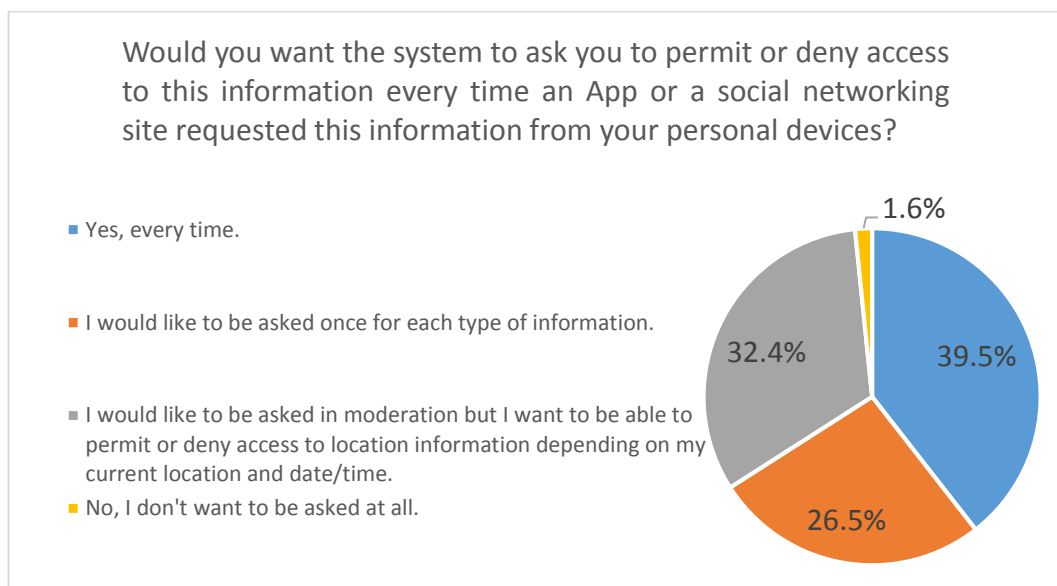


Figure 60. Access control for contextual information.

The following text was used to briefly describe to users what privacy preferences are:

Privacy Preferences (or settings) could be used to block access to this information (context information) under certain circumstances. For example, you may block specific services, apps or people to access your location information (both GPS and symbolic) when you are at specific locations or performing some activity.

As shown in Figure 61, users were asked whether they would create such privacy preferences to block access to their context information. Only 5.9% of respondents stated they would not use them due to time restrictions. 44.9% stated they would find it tedious to configure them manually. It is safe to deduce that there is a need for a mechanism that

learns from the users' previous privacy decisions to facilitate the creation of privacy preferences. Figure 62 shows how useful respondents find a privacy preference learning mechanism. Only 7.6% don't find it at all useful, while the rest of the respondents find it useful in varying degrees.

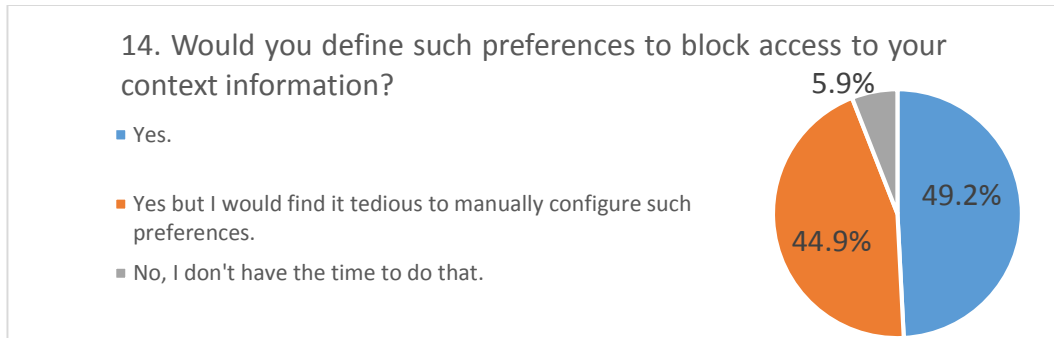


Figure 61. *Creating privacy preferences.*

Finding 8: Almost all users (94.1%) would define privacy preferences but half of them would prefer not to do it manually. Almost the same number (92.4%) feel that preference learning would be useful.

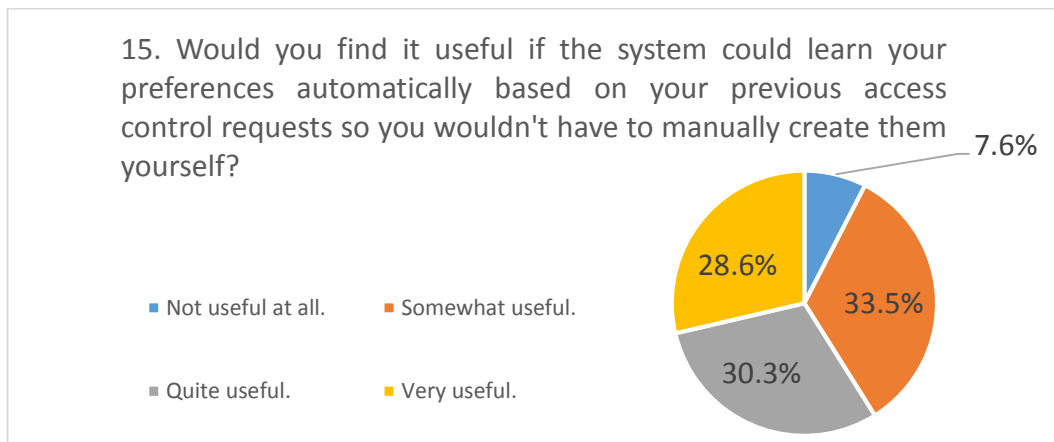


Figure 62. *Assistance using Privacy Preference Learning*

Using Extreme Dark Scenarios

Four very short dark scenarios were presented to the respondents to demonstrate extreme cases of disclosing information with consequences of varying degree. After reading these scenarios, the respondents were asked to comment freely on them and answer questions three questions (Figure 63, Figure 64 and Figure 65 respectively)

Scenario 1: *Social network disclosures.*

While you are booking a flight, you are given the option to use a tool that can offer to seat you next to any friends in your social networks who happen to be on the same flight. You use the tool but you don't find any friends on the flight. On the day of the flight, you realise that one of your "social network friends" who you don't really want to talk to is sitting right next to you because they spotted you using the seat finder tool.

Scenario 2: Location based services in combination with poor inference.

Your friend is having an abortion so you go with her for support to the clinic. One of the apps on your phone which monitors your location records this information on your profile. A few years later, you are applying for a job but the person reviewing your application rejects it because of the apparent pro-abortion stance revealed by your profile. Needless to say, a different reason is given to you.

Scenario 3: Location based services in combination with poor profiling.

Arriving on an international flight, you are pulled out of the line by immigration officials, detained and interrogated for 24 hours because your physical characteristics match someone on their terrorist watch list and places you have recently visited, recorded by an application on your phone, match locations in which the terrorist has been spotted.

Scenario 4: Health sensor-based service.

On the advice of your doctor, you make use of a wrist band that monitors your heart rate along with a mobile app that transmits this data to a computer at your local hospital where it is stored along with the data of many other similar patients. Meanwhile the local police are investigating a series of serious crimes but getting nowhere. One of the victims works at the hospital and, suspecting one of the patients as being the perpetrator, makes all the data available to the police. Coincidentally, your data happens to indicate an increase in your heart rate at the time of every crime. With nothing else to go on, the police make you their prime suspect and start searching for evidence that can be used against you in court.

Respondents were asked to comment on these scenarios. The most common comments collected from respondents were:

- These scenarios, albeit exaggerated, could happen but under very specific circumstances.
- Most respondents described the scenarios as scary and worrying. Some also mentioned George Orwell's '1984' novel [169] famous for its depiction of an authoritative state where all citizens are constantly monitored and controlled by the state.
- It had not occurred to many respondents that personal information can be used in this way to their disadvantage.
- Some respondents stated that the scenarios made them rethink their current privacy settings.
- Some respondents commented that these scenarios present the worst consequences of disclosing information. However, technology should be developed to aid users to perform tasks whilst protecting their privacy appropriately.
- Finally, some respondents raised the concern that existing privacy laws may not be adequate to protect the privacy of individuals and hence appropriate tools must be provided to help users protect their privacy themselves.

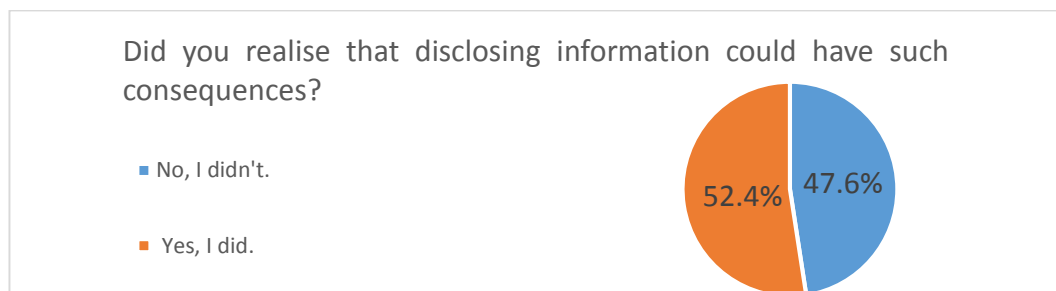


Figure 63. *Consequences of disclosing personal data.*

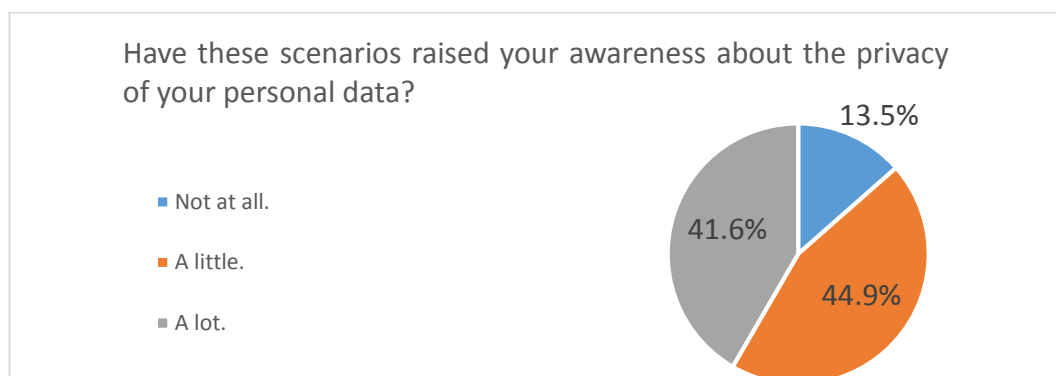


Figure 64. *Awareness of privacy changed.*

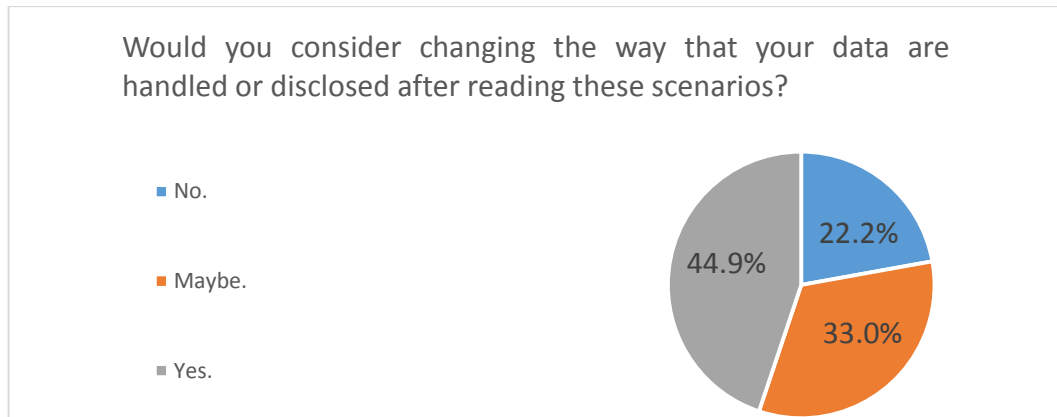


Figure 65. *Users consider changing data disclosure practices.*

7.2 *Evaluation of PersonISM system*

The PersonISM system was evaluated in a live trial using students and staff from Heriot-Watt University in Edinburgh. The experiment took place in December 2014 in the Pervasive, Ubiquitous and Mobile Applications Lab. 21 people from the School of Mathematical and Computer Sciences (both staff and students) participated in the experiment. Each user was assigned a one hour slot to evaluate the PersonISM system. The experiment was split into a series of tasks with a list of questions at the end of each task.

7.2.1 **Pre-Evaluation Step**

Before evaluating the PersonISM system, each user was asked to read three privacy policies in the same way as if they were sitting at home and started to use the respective services and answer a list of questions about what they read. The selected privacy policies were Google's (3659 words) [170], the BBC website's privacy policy (3282 words) [171] and Heriot-Watt University's privacy policy (5219 words) [172]. The privacy policies were selected for the following reasons:

- a) The length of the privacy policy.
- b) The familiarity of the users with the companies or organisations. All three can be considered widespread and well known to the experiment participants.
- c) The differences in the content and the presentation of the content in each privacy policy. Google and BBC present a well-structured privacy policy document in HTML format containing links that help explain some terminology which might be unknown to the average user. Heriot-Watt University provides a legal document in PDF format.

The current industry practice for informing users about what happens to their data is to present a lengthy privacy policy document to the user and request their informed consent by way of ticking a checkbox in a page. The reading speed for an average reader is 200 to 400 words per minute for comprehension [173]. However, the reading speed is lower when the reader is not familiar with the subject and the text is not in the user's native language. McDonald and Cranor conducted a survey which shows that the average time to read a privacy policy is 10 minutes using an average document length of 2500 words. That only shows that the user has read the document in 10 minutes but does not show how well they understood the content. If an individual were to read the privacy policy at every website they visited even once per year, they would spend, on average, an estimated 244 hours per year reading privacy policies (calculated by visiting an average number of 1462 websites per year) [5].

In the first step of the PersonISM evaluation, the following timings were observed for reading each privacy policy:

	Google	BBC	HWU
average	04:52	04:19	05:53
min	00:02	00:02	00:33
max	15:32	10:27	16:45
median	03:49	03:29	04:42

Table 1. Statistics of time spent reading privacy policies.

4 out of 21 users spent a minute or less reading each privacy policy as they only skimmed through the text.

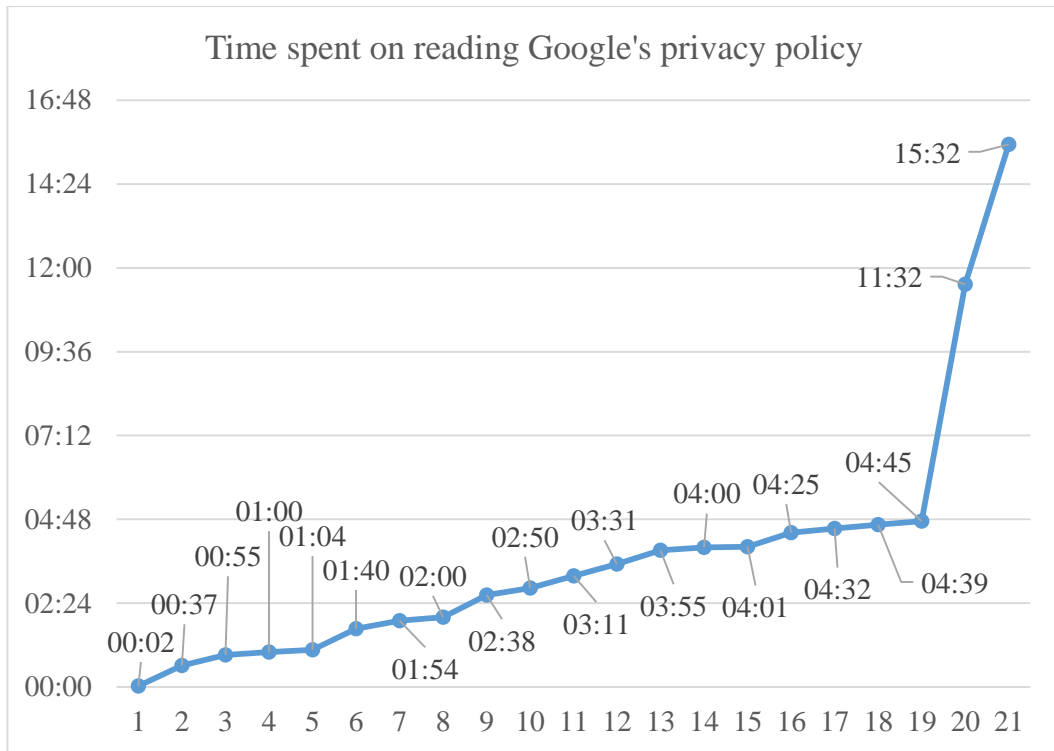


Figure 66. Google Privacy Policy read times.

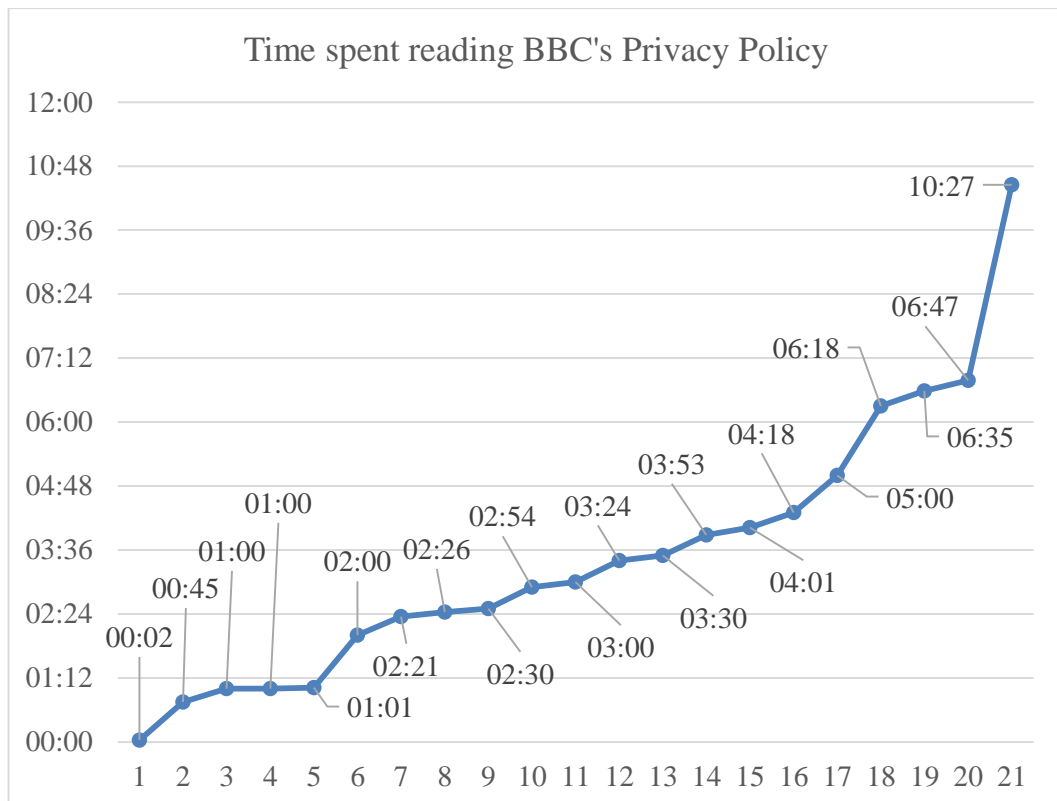


Figure 67. BBC Privacy Policy Read Times.

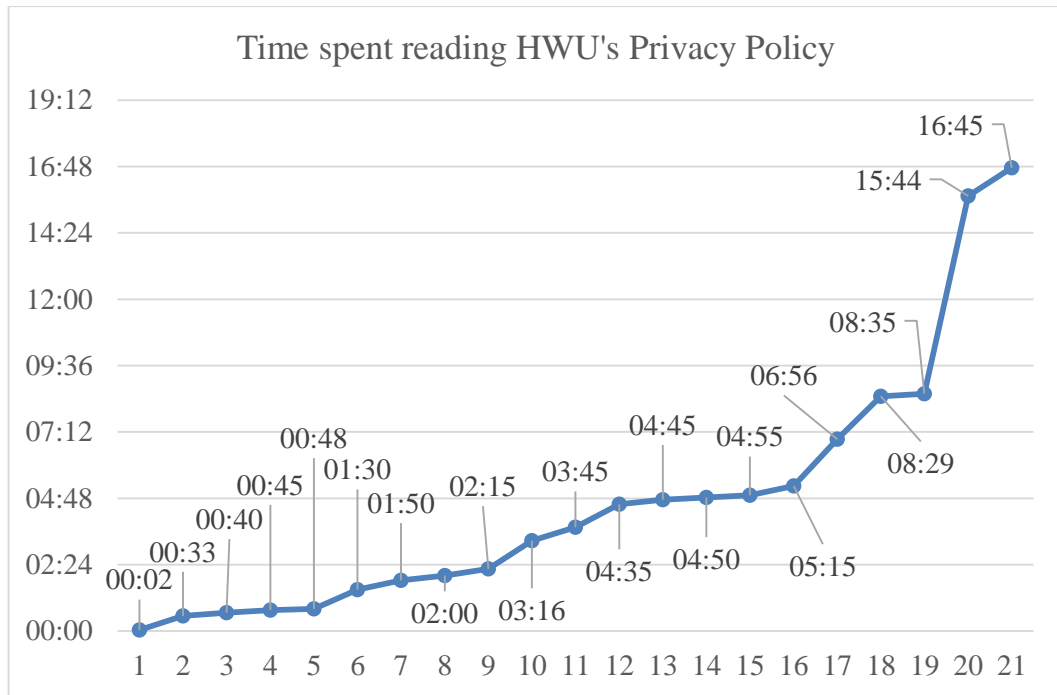


Figure 68. *Heriot-Watt University Privacy Policy Read Times.*

As users were asked to read the privacy policies in the same way they would read them at home, the majority of them did not try to assimilate and comprehend the text. This finding is obvious in the answers they gave when they were asked to report some of the information they had read on the privacy policies.

Pre-Evaluation Questions

After reading the privacy policies, users were asked to answer a list of questions about what they had read.

4. *Do you use the BBC and Google websites and HWU services regularly?*

Google: 100% (21 out of 21 users use Google regularly)

BBC: 71.4 (15 out of 21 users use BBC regularly)

HWU: 100% (21 out of 21 users use HWU services regularly)

The numbers indicate that the users are regular users of the three websites.

5. *Have you ever read their respective privacy policies before?*

Google: 23.8% (4 of 21 users) have skimmed through Google’s privacy policy and 19% (5 of 21 users) have read it.

BBC: 14.3% (3 of 21 users) have read BBC's privacy policy.

HWU: 14.3% (3 of 21 users) have read HWU's privacy policy.

The statistics indicate that a small minority have read privacy policies of these websites that they regularly visit.

6. *Before you read their privacy policies did you know what information BBC, Google and HWU collect through your interactions with their services?*

Google: 33.3% (7 out of 21) said they didn't know what information Google was collecting about them. 52.4% (11 out of 21) answered they assumed or had a rough idea and 14.3% (3 out of 21) answered that they knew.

BBC: 38.1% (8 out of 21) said they didn't know what information the BBC was collecting about them. 52.4% (11 out of 21) answered they assumed or had a rough idea and 9.5% (2 out of 21) answered that they knew.

HWU: 38.1% (8 out of 21) said they didn't know what information HWU was collecting about them. 47.6% (10 out of 21) answered they assumed or had a rough idea and 14.3% (3 out of 21) answered that they knew.

7. *Before you read their privacy policies did you know what they do with the information they collect from you?*

Google: 38.1% (8 out of 21) said they didn't know what Google did with their information. 47.6% (10 out of 21) answered they assumed or had a rough idea and 14.3% (3 out of 21) answered that they knew.

BBC: 47.6% (10 out of 21) said they didn't know what BBC did with their information. 33.3% (7 out of 21) answered they assumed or had a rough idea and 19% (4 out of 21) answered that they knew.

HWU: 47.6% (10 out of 21) said they didn't know what HWU did with their information. 42.9% (9 out of 21) answered they assumed or had a rough idea and 9.5% (2 out of 21) answered that they knew.

8. *After reading their privacy policies do you know exactly what information BBC, Google and HWU collect through your interactions with their services? (list the information they collect)*

38.1% (8 out of 21) couldn't list the information Google, BBC and HWU collect even after reading their privacy policies. 61.9% (13 out of 21) listed some information (personal details - name, age, DOB, GPS information, browsing history, clicks, registration information, credit card information, linked accounts, email addresses).

9. *After reading their privacy policies do you know what they do with the information they collect from you? (write how they use your information)*

19% (4 out of 21) couldn't answer what Google, BBC and HWU do with the collected information even after reading their privacy policies. 47.6% (10 out of 21) listed some information (targeting ads, website improvements, service customisation, selling information to third parties, recommendations).

10. *After reading the privacy policies, can you tell how you can make changes (remove, edit) to the information that was collected from you from their systems?*

Google: 76.2% (16 out of 21) could not tell how they can remove or edit their information on Google. 23.8% (5 out of 21) claimed to know how.

BBC: 76.2% (16 out of 21) could not tell how they can remove or edit their information on the BBC website. 23.8% (5 out of 21) claimed to know how.

HWU: 81% (17 out of 21) could not tell how they can remove or edit their information held by HWU. 19% (4 out of 21) claimed to know how.

11. *How long will the information that was collected be kept in their systems?*

Google: 28.6% (6 out of 21) answered that their information will be kept in their systems for as long as necessary or for as long they use Google services. 23.8% (5 out of 21) answered that their information will be kept forever. 47.6% (10 out of 21) answered they do not know.

Note: Google's privacy policy does not state clearly how long the information will be kept by Google or companies that use Google user data. Information from search queries are anonymised after 9 months. Anonymisation is performed by removing the last octet from the IP address (contrary to EU data supervisors that call for IP anonymisation after 6 months). However this is not stated anywhere in Google's privacy policy. According to the Commission Nationale de l'Informatique et des Libertés [174] – France's regulatory

body for the enforcement of the EU Data Protection Directive, Google has refused to provide a data retention period [175].

BBC: 19% (4 out of 21) answered that their information will be kept in their systems for as long as necessary or for as long they use BBC services. 28.6% (6 out of 21) answered that their information will be kept forever. 52.4% (11 out of 21) answered they do not know.

Note: The following excerpt is from the BBC Privacy Policy section on “How long will the BBC keep my personal information?”

“We will hold your personal information on our systems for as long as is necessary for the relevant activity, or as long as is set out in any relevant contract you hold with the BBC or the BBC's corporate retention schedule (a database that defines which documents should be kept and for how long). If you cancel your registration as a BBC website member and your account is deleted a red flag goes on the database and, while the BBC cannot use the personal information, it stays on the system for a period of one year for administration purposes before being deleted automatically.” [171]

HWU: 19% (4 out of 21) answered that their information will be kept in their systems for as long as necessary or for as long they use HWU services. 23.8% (5 out of 21) answered that their information will be kept forever. 57.1% (12 out of 21) answered they do not know.

Note: Section 3.7 of the Heriot-Watt University Data Protection Policy [172] states that personal data will be retained only for as long as required according to the University records retention policies. In some cases, some information may be held for historical or statistical reasons but personal data will be redacted.

12. Do you know whether information you have disclosed to these companies has been shared with other companies? (yes/no)

61.9% (13 out of 21) answered they did not know whether their information has been shared with other companies.

13. Do you know how to stop these companies from sharing your information with other companies (yes/no)?

85.7% (18 out of 21) answered they did not know how to stop companies from sharing their information with other companies.

14. Do you know if your information has been processed further and combined with other information to infer more information about yourself (yes/no)?

66.7% (14 out of 21) answered they did not know if their information has been used to infer further information about them.

15. Do you know how to tell the company to stop doing that (yes/no)?

95.2% (20 out of 21) answered they did not know how to tell companies to stop combining their information to infer more information about them.

Pre-Evaluation answers discussion.

Only a small minority of users stated that they know what information the services are collecting about them and what they do with the information they collect. However, it is more noteworthy that even after reading the privacy policies, the majority of the users could not accurately describe what information the services were collecting about them and what they did with the information they collected.

Specifically in the case of Google, which does not state in its privacy policy how long they keep personal information, 28.6% of users stated that their information would be kept by Google for as long as necessary and 23.8% of users answered that their information would be kept forever. It can be concluded that users can often perceive the information erroneously or assume statements with no actual basis.

Hence, there is a clear need to present the terms and conditions for disclosing data and interacting with services in a manner that allows users to easily comprehend what information is collected, how long it is kept, how it is used, processed and shared with others.

7.2.2 Experiment specification

One of the problems with evaluating the PersoNISM system with real users is the amount of time available to conduct the experiment. The benefits of the PersoNISM system can only be demonstrated properly after the user has used the system a few times so that enough information has been accumulated to create preferences that can be used to automate the processes involved. However, due to the fact that users have not experienced a privacy policy negotiation process, the aim of the 1st step of the PersoNISM evaluation is to compare the practices of reading a privacy policy and accepting it with participating in a privacy policy negotiation with a service provider. Hence, the 1st step of the experiment is performed without any privacy preferences present in the system. The privacy policy negotiation that is performed during the 1st step is performed manually where the user must configure all the settings to continue. In later steps, as the user starts to provide information and take decisions during the privacy policy negotiations, identity creations and selections, and the system has started to accumulate enough information to apply them, the user is subjected to personalisation in all of the steps. The experiment was designed this way to show to the user a gradual personalisation being applied while they use all the privacy functionality.

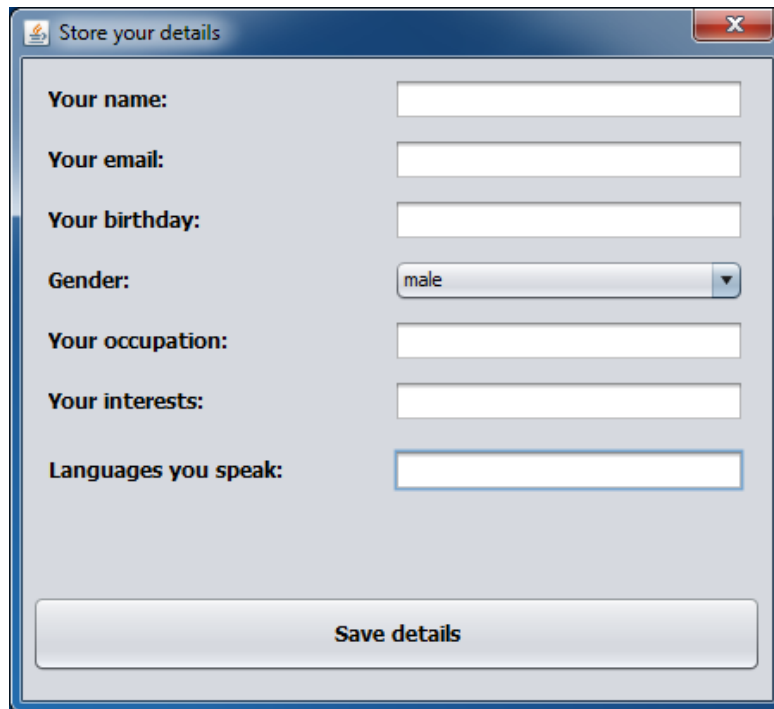
Four mock applications were implemented to evaluate the PersoNISM system with real users. The applications are a fictional application called “Google Venue Finder”, a fictional application “HWU Campus Guide App” and mock versions of a BBC News app and a BBC Weather app. Each application has its own privacy policy. Even though all privacy policies request the same data – name, age, date of birth and location, the conditions for disclosing the data and the corresponding condition ranges differ for each service.

PersoNISM Evaluation Step 1 of 3

Users are given a handout to follow so that all users complete a pre-defined set of steps with minimal intervention from the investigator. The handout document can be seen in appendix B.

In step 1, users are requested to fill in some of their personal information as shown in Figure 69. Personal information is requested in an effort to instil a sense of ownership of

the data that is going to be used in the experiment despite the fact that users are aware they are in a protected environment.



The screenshot shows a window titled "Store your details" with a close button in the top right corner. The window contains several input fields and a dropdown menu. The fields are labeled: "Your name:", "Your email:", "Your birthday:", "Your occupation:", "Your interests:", and "Languages you speak:". The "Gender:" field is a dropdown menu currently showing "male". At the bottom of the window is a large button labeled "Save details".

Figure 69. Users enter their personal details.

After inputting their personal information, the PersonISM evaluation tool GUI starts showing the four apps that are available for installation (Figure 70).



Figure 70. PersonISM evaluation tool – available services.

Next, the user is asked to rate the four service providers in terms of how much they trust it using a trust settings GUI that allows them to enter a trust level value using a slider (shown in Figure 71). Users are told that the actual trust level values are not important but rather how they rate one service provider relative to another. The fact that one service provider is rated higher or lower than another is what will play an important role later in the privacy preference learning and privacy preference evaluation.



Figure 71. *Adjusting trust level values.*

The next step involves installing the first service. According to the trust level values that the user has entered, the user is told to install the service with the lowest trust level. The user is unaware of the reason for the selection of the service. The service with the lowest trust level is selected first so that the preferences that will be learnt first can be reused in later steps to demonstrate how the privacy preferences can aid the user in configuring their privacy settings.

The Privacy Policy Negotiation form as shown in Figure 72 presents the terms and conditions for using Google Maps. The current page as shown in Figure 72 presents the conditions for disclosing the data type “birthday”. After the user has configured the conditions to match their privacy needs, they are told to continue by clicking the “Next” button. As mentioned before, each privacy policy used in the experiment defines four data

types to be accessed: name, birthday, email and location. Hence, the next page will show the conditions for one of the remaining data types. After the user has configured the settings for all four data types, they are asked to confirm their selections.

Figure 72. *Privacy Policy Negotiation with Google.*

Next, the privacy policy negotiation process begins and the user is presented with the response from the “mock” Google service provider. The terms that Google could not accept are highlighted with a warning sign as shown in Figure 73.

Privacy Policy Negotiation Form

i The terms and conditions you requested for the data items: name, locationSymbolic, email, birthday from the provider were not entirely acceptable. The provider has suggested alternatives. If you accept the alternatives provided, you can continue to install the service. Otherwise, the negotiation will fail.

Conditions for accessing: name

Purpose Your name is needed so you can be identified by your contacts.

Your name will be kept by www.google.com for: until account is deactivated ⚠

Allow sharing of your name with : No sharing ✓

Allow your name to be used to infer further information about you yes ⚠

Conditions

Demand that your name be stored securely by www.google.com yes ✓

Demand the right to opt out of disclosing your name at any time no ⚠

Demand access to view your name from www.google.com yes ✓

Demand to correct your name held by www.google.com if incorrect yes ✓

Get Personalised Suggestions Restore changes

Next >

Cancel Continue

Figure 73. *Google's Privacy Policy Negotiation Response.*

The user is asked to review the refined terms and conditions and to click Continue if they agree or cancel if they do not accept them. If they click cancel, the privacy policy negotiation process aborts and the service cannot be installed. If the user accepts the new terms, the next step requires them to select an identity to use with Google Maps. Since this is the first installation for the user, there are no such identities yet. The window to create a new identity is shown in Figure 74. The handout explains how to use the GUI to create the identity with the data that the user wants to disclose to Google. The GUI allows the user to enter new data to associate with their identity if they don't wish to associate the data that already exists in the system.

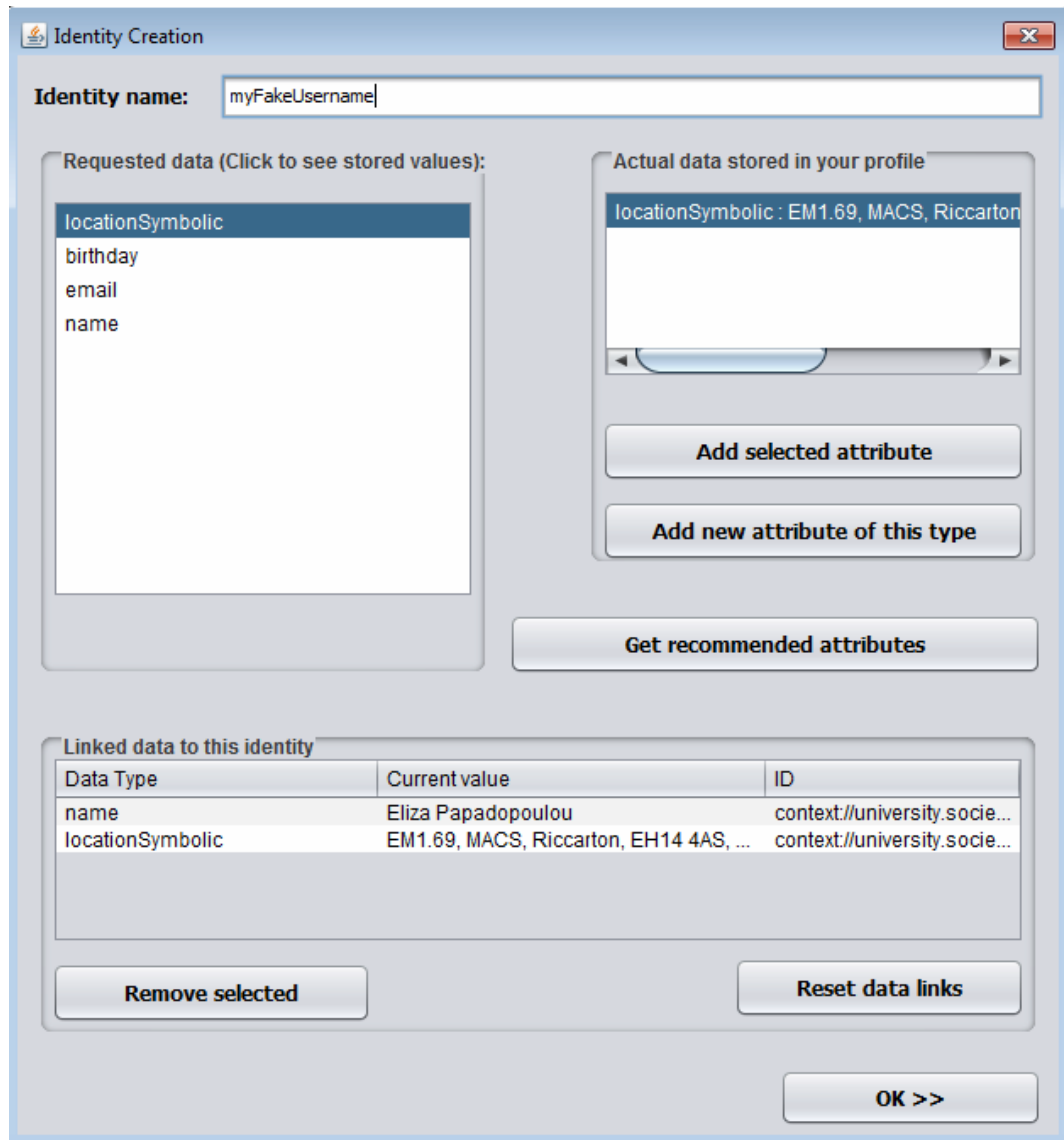


Figure 74. Identity Creation GUI.

The identity must be associated with all the data requested in the privacy policy. After creating the new identity, the privacy policy negotiation process finalises and the service is installed. At this point, the user is asked the following questions:

Q1. On a scale of 1 to 10, indicate the level of understanding of the terms and conditions in each form presented to you: (1 is you don't understand at all, 10 being you understood fully)

Text:	1	2	3	4	5	6	7	8	9	10
PersonISM:	1	2	3	4	5	6	7	8	9	10

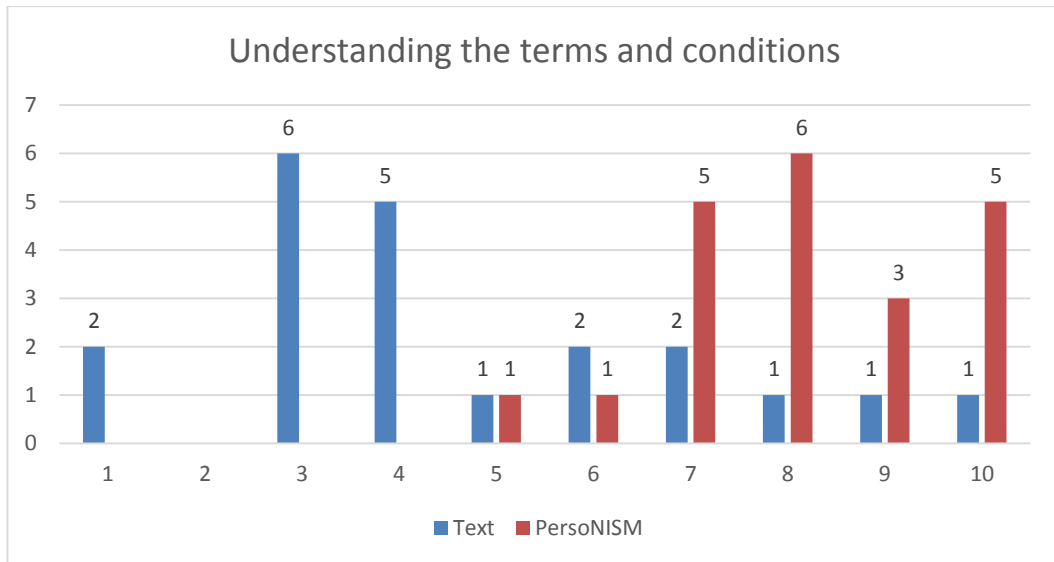


Figure 75. *Understanding the terms and conditions answers.*

As shown in Table 2, the PersoNISM system presented the terms and conditions in a much more understandable manner than a privacy policy document. The average level of understanding a privacy policy document on a scale of 1 to 10 is 4.67 with a median of 4, while understanding terms and conditions on the Privacy Policy Negotiation GUI is 8.14 with a median of 8. A graphical representation of the results is also shown in Figure 75.

	Average	Min	Max	Median
Text	4.67	1	10	4
PersoNISM	8.14	5	10	8

Table 2. *Comparing the level of understanding the terms and conditions.*

In the next two questions, users are asked to evaluate a) the value of specifying the terms and conditions with such a fine granularity on a per data and per service basis and b) how they would feel about configuring the terms and conditions with such fine granularity. The questions were phrased as follows:

Q2. On a scale of 1 to 10, how valuable is it to specify the terms and conditions on a per data and per service basis? (where 1 is not valuable at all and 10 being extremely valuable)

Q3. On a scale of 1 to 10, how happy would you be to configure the terms and conditions for every data type and every service? (1 being not happy at all and 10 being very happy)

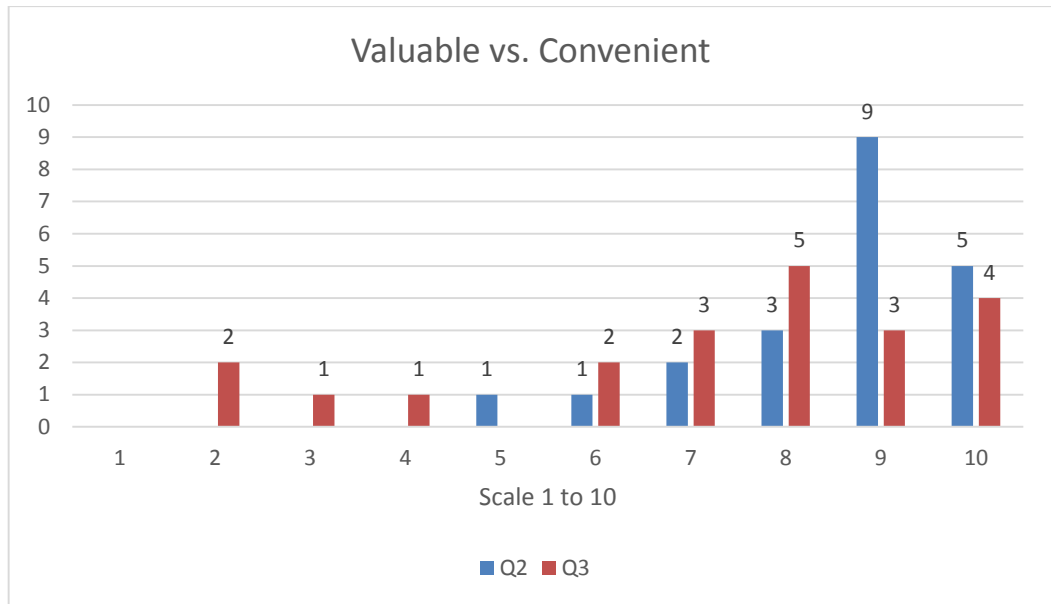


Figure 76. Valuable vs. Convenient presentation of T&Cs.


	Average	Min	Max	Median
Q2	8.57	5	10	9
Q3	7.19	2	10	8

Table 3. Q2 – Q3 answers.

As shown in Figure 76, users consider it valuable to be able to modify the terms and conditions for every data type, each condition and service. However, they would not be too happy to have to configure all these items manually. This was expected as the number of data types would not normally be as small as four but rather a much longer list.

PersoNISM Evaluation Step 2 of 3

In the 2nd step of the PersoNISM evaluation, the user experiences the use of personalised suggestions in all the steps of the installation process as well as an automatic negotiation based on their previous decisions. Finally the user is asked to evaluate their experience.

Following the instructions on the handout, the user attempts to install another application. At the negotiation GUI that pops up, they can now use the “Personalised Suggestions” button (as shown in Figure 72). The system has accumulated some privacy preferences based on the previous PPN with Google and can now suggest parameters to the user. The suggested parameters are highlighted with a  sign. Continuing on in a similar fashion to the previous step, when the user reaches the Identity Selection process, they are asked to select an identity to interact with the new application (shown in Figure 77). The user

can make use of the “Get recommended identity” button which can suggest the most appropriate identity to use with the service they are installing. The user can opt to create a new identity as well.



Figure 77. *Selecting an Identity Dialog.*

In this case, the user is encouraged to create a new identity in order to experience the personalisation in the identity creation process by clicking on “Get Recommended Attributes” in the Identity Creation GUI (Figure 74). The suggested attributes are listed as shown in Figure 78. Users have the option of selecting the attributes they prefer using the tick boxes.

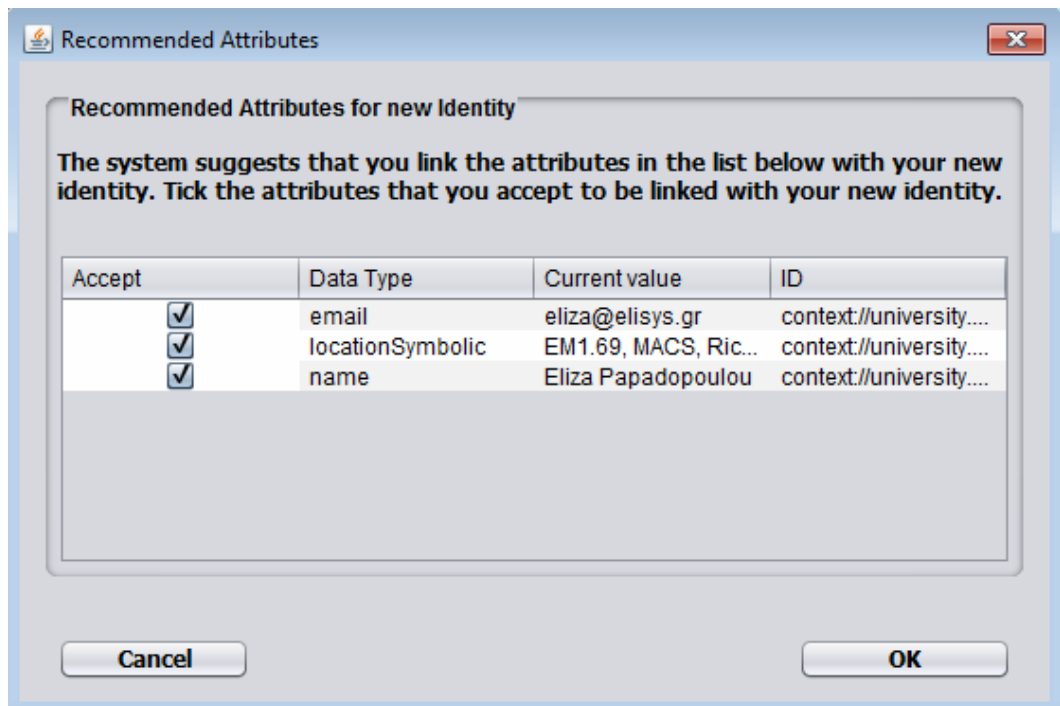


Figure 78. *Suggested attributes for new identity.*

This task served to familiarise the user with the semi-automatic nature of personalisation of the privacy functionality offered by the PersonISM system. The fully automated mode of the PersonISM system is demonstrated to the user when they are asked to install a third application, BBC Weather, where all the privacy functionality is performed without manual intervention after the user has consented (see Figure 79).



Figure 79. *Automatic Negotiation Dialog.*

The following questions were asked at the end of the 2nd step.

Q1. On a scale of 1 to 10, did you feel that the personalised suggestions were useful?
(1 being not useful at all and 10 being very useful)

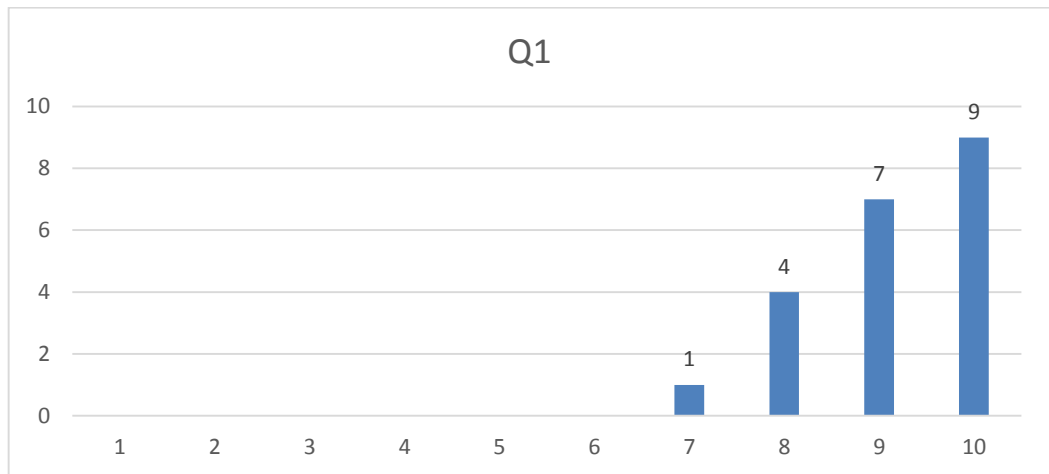


Figure 80. *Are personalised suggestions useful?*

Q2. On a scale of 1 to 10, how much would you trust the system to perform this process on your behalf without your involvement in later negotiations? (1 indicating no trust in the system at all and 10 indicating complete trust in the system)

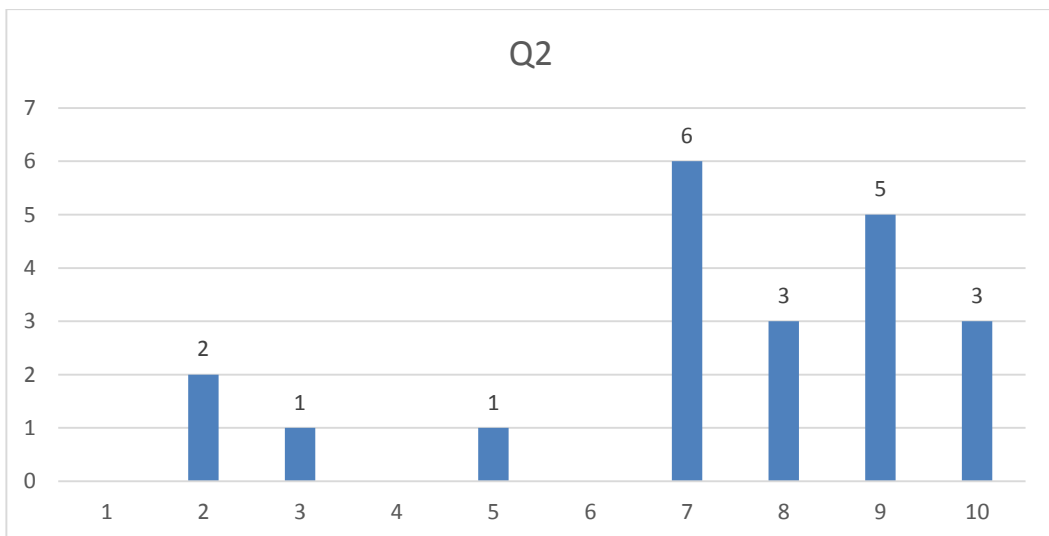


Figure 81. *Trusting the system to automate processes on behalf of user.*

The purpose of question 2 was used to measure if users would welcome the level of automation made available to them. It is important that the users have the option to choose the automation rather than it being imposed on them without their explicit consent.

Q3. On a scale of 1 to 10, how likely would you be to configure the terms and conditions for every data type and every service with the help of personalised suggestions? (1 being very unlikely and 10 being highly likely)

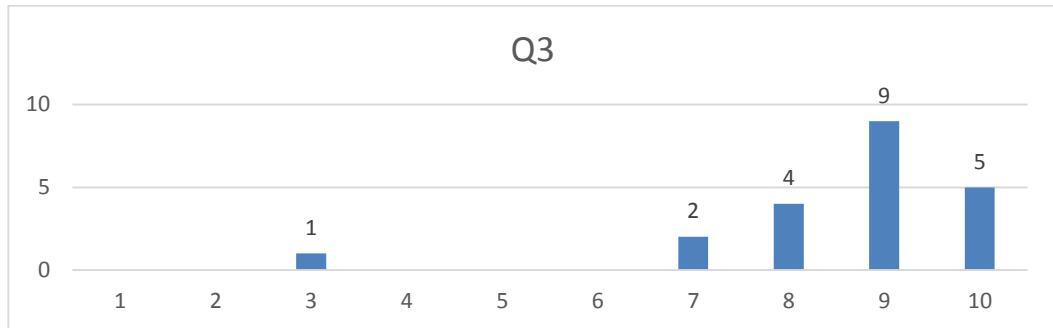


Figure 82. *Configuring T&Cs with personalised suggestions.*

	average	min	max	median
Q1	9.14	7	10	9
Q2	7.55	2	10	8
Q3	8.57	3	10	9

Table 4. *Personalised Suggestions Statistics of Step 2 Questions.*

PersoNISM Evaluation Step 3 of 3

In this final step, the users are subjected to personalised access control and data obfuscation. The data obfuscation was not a feature that users had experienced before.

Users were asked to start one of the installed applications. Upon starting up, the application requested a login username and then attempted to retrieve the user’s data from the system. The system asks the user to allow or block access to the data using notification boxes (such as the one shown in Figure 83).

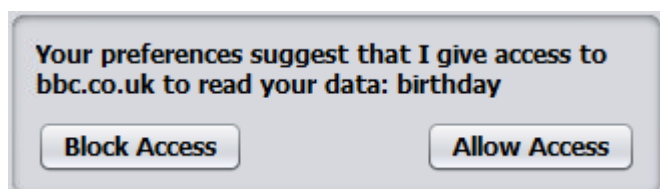


Figure 83. *Access control notification for “birthday”.*

After each request, and if the user allows the disclosure, it also asks for a data obfuscation level by displaying examples of applying data obfuscation to each data type as shown in Figure 84.

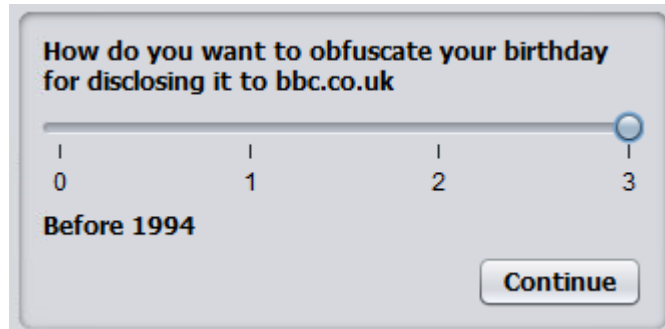


Figure 84. *Data Obfuscation notification for “birthday”.*

The system should have accumulated privacy preferences with high confidence level. To demonstrate this to the user, they are asked to trigger a second request for data by the service by clicking on the “Update profile” button (as shown in Figure 85). As the confidence level of the privacy preferences is high, the user sees a timed notification instead of a standard notification.



Figure 85. *Mock Venue Finder App after having retrieved data.*

At the end of step 3, the user is asked to answer the following questions:

Q1. Taking into account all the steps you did, on a scale of 1 to 10 indicate if using this tool would make you take your privacy more seriously. (1 indicating that it would make you take it less seriously, 5 indicating no change and 10 indicating it would make you take your privacy very seriously)

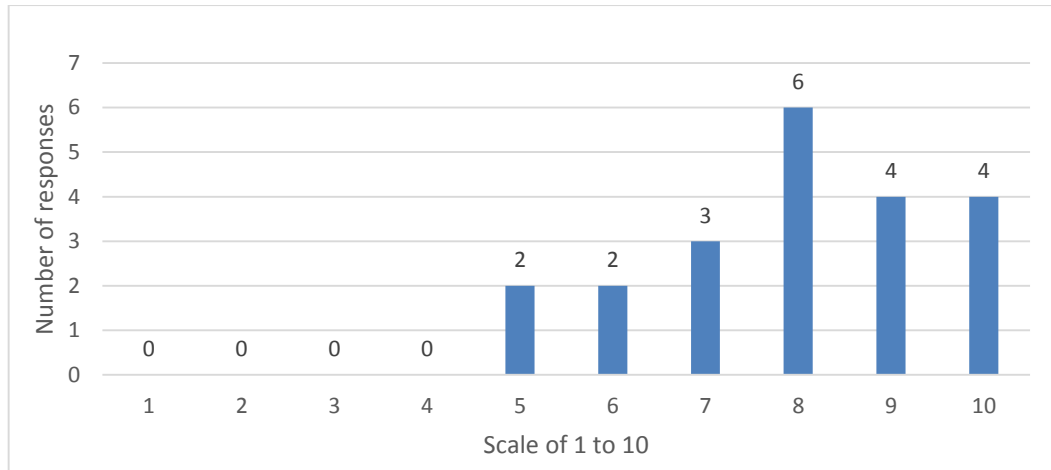


Figure 86. *Would using PersonISM make users take privacy more seriously?*

Q2. On a scale of 1 to 10, indicate how useful you would find having such control of your personal data. (1 being not useful at all and 10 being very useful)

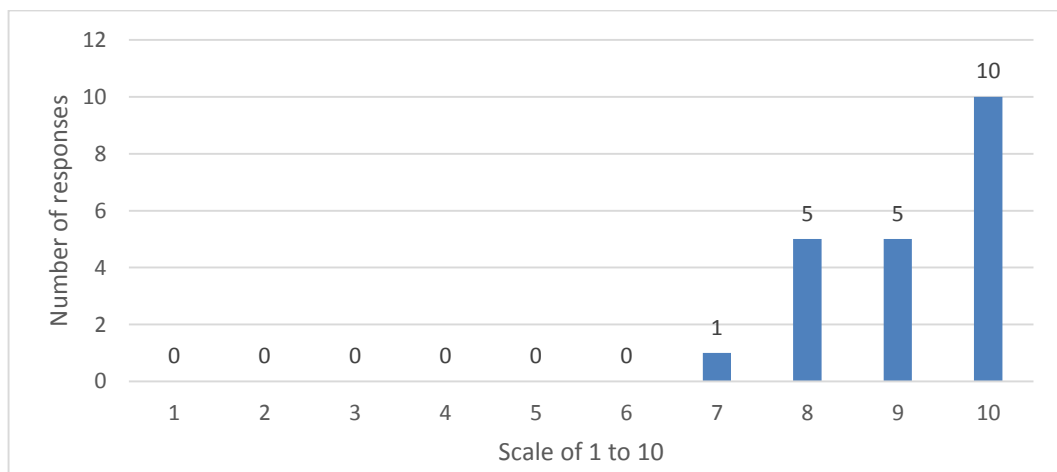


Figure 87. *How useful it is to have such control of personal data.*

Q3. On a scale of 1 to 10, indicate how necessary you think it is to have tools like this to control your privacy. (1 not necessary at all, 10 being essential to your privacy)

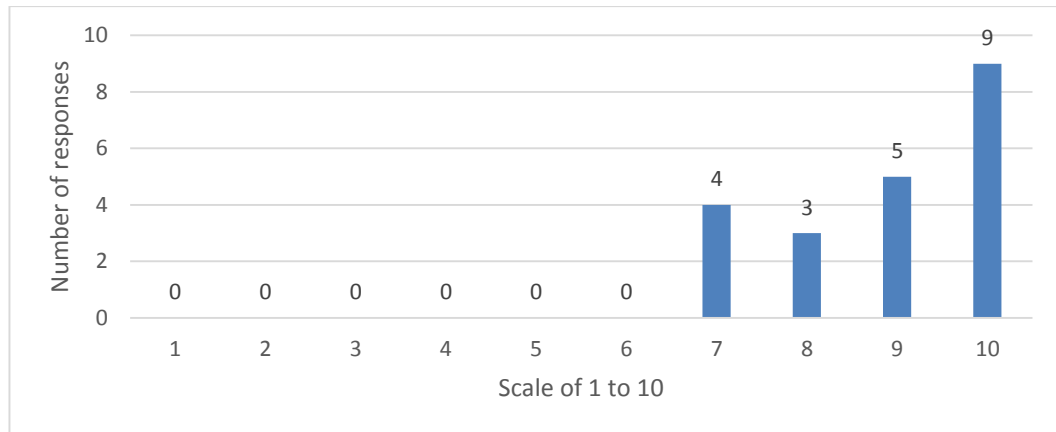


Figure 88. *How necessary it is to have a tool like PersonISM.*

Q4. Do you think service providers (such as Google, Facebook, websites) should be forced by law to give YOU control of your data? (yes/no)

100% of users answered yes, they believe service providers should be forced by law to give them control of their data.

Q5. Do you believe that personal devices such as mobile phones and tablets should have built in privacy control mechanisms such as the one you experimented with? (yes/no)

100% of users answered that they want personal devices to have built-in privacy control mechanisms such as PersonISM. One user commented that more training should be given to use the tool more efficiently and another user commented that it would be hard to get service providers to take such a tool seriously.

Q6. As our world becomes more and more pervasive do you believe that privacy protection is becoming a bigger problem? (1 indicating you don't believe it's becoming a bigger problem, 10 indicates you think it's a very big issue).

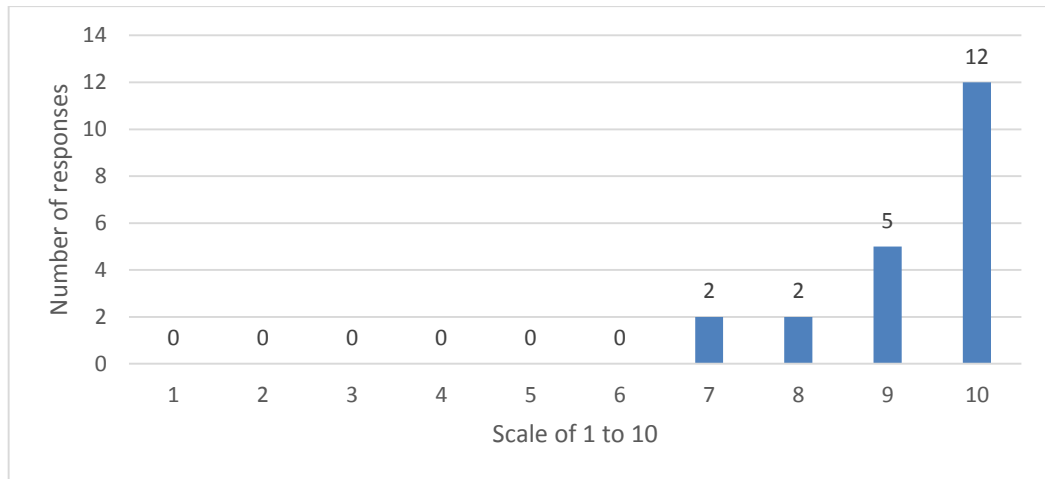


Figure 89. *Is privacy becoming a bigger problem?*

Q7. If tools such as the ones you have just experimented with were provided at a small cost, would you be prepared to pay for them to protect your privacy? (1 indicates you would never pay, 10 being you would definitely pay)

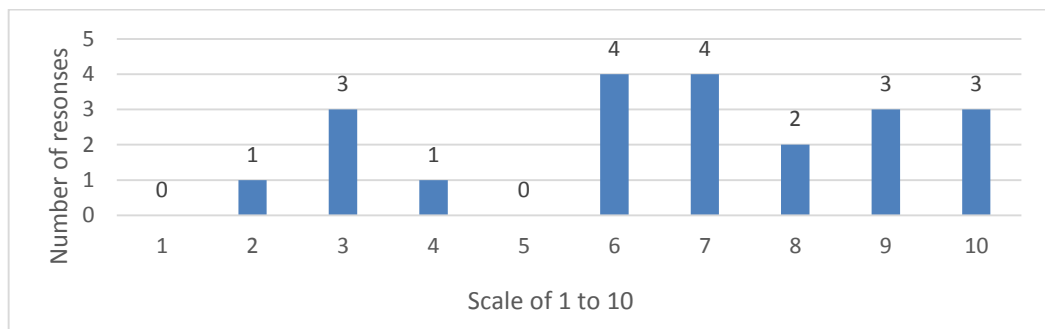


Figure 90. *Paying for privacy tools such as PersoNISM.*

Q8. Finally, can you express in a few words what would be your biggest worry regarding your privacy in a pervasive environment?

The most common concern expressed by the users is the misuse of personal information and the possibility of someone using the data to cause them harm such as stealing their identity. Some users expressed their fear that the world is moving into a constantly monitored state where nobody can remain anonymous (the “Big Brother” effect). For some, lack of control of personal data means loss of personal freedom and the freedom to think individually without bias.

7.2.3 Evaluation Results Analysis

The evaluation results show that the Privacy Policy Negotiation form presents the contents of a privacy policy in a much more user-friendly manner than a textual privacy policy document. It is preferable to present the terms and conditions succinctly, in a manner that invites the user to pay attention to the privacy policy as its purpose is to get the user's informed consent. When users just accept the terms and conditions without reading them, then the consent cannot be considered informed. As the evaluation shows, the PersoNISM system can improve the user's comprehension of terms and conditions.

Even though the PPN form provides a medium to present all the terms and conditions in great detail, the results show that the users would not be very happy having to configure all the terms and conditions for each data type every time they install a new service despite the fact that they find it very valuable. We can deduce that the value of the privacy policy negotiation is to give users the power to negotiate the terms and conditions of the data that matters to them. Step 1 of the evaluation process was designed specifically to highlight this issue in order to demonstrate the benefits of the personalisation in steps 2 and 3. Any process that involves spending a lot of time configuring settings can frustrate users and eventually discourage users from using the system.

Even though the evaluation only used four data types, in reality services could request a very long list of data types. The fact that half of the users reported that they would not be very happy to configure the terms and conditions for each data type and service in this example with only four data types, suggests that a long list of data types would be extremely tedious for them to configure. However, not all data types are important to all users. Hence, one solution to this problem would be to learn which data types are important for each user and draw their attention to those. This feature would also require a few negotiations to take place to be appropriately useful.

The personalised suggestions in the privacy policy negotiation process were rated quite highly; demonstrating the benefits of personalisation in a process that can become quite tedious to perform manually. The use of personalised suggestions is very obvious to the user as the application of the accumulated privacy preferences is highlighted appropriately to draw the attention of the user to them. Contrary to that, the automated privacy policy negotiation using privacy preferences performs the entire process without

the user’s involvement. The evaluation results show that this kind of automation is not highly regarded as users don’t trust the system enough to perform the negotiation on their behalf. However, some users appreciated the option to automate this process. Therefore, the flexibility feature to allow users to use it if they wish but not to force it on them is sound.

The responses to Question 3 of Steps 1 and 2 show that the personalised suggestions provide an important improvement in the privacy policy negotiation process. Users rate the use of the PersonISM system with the use of personalised suggestions much higher than without them.

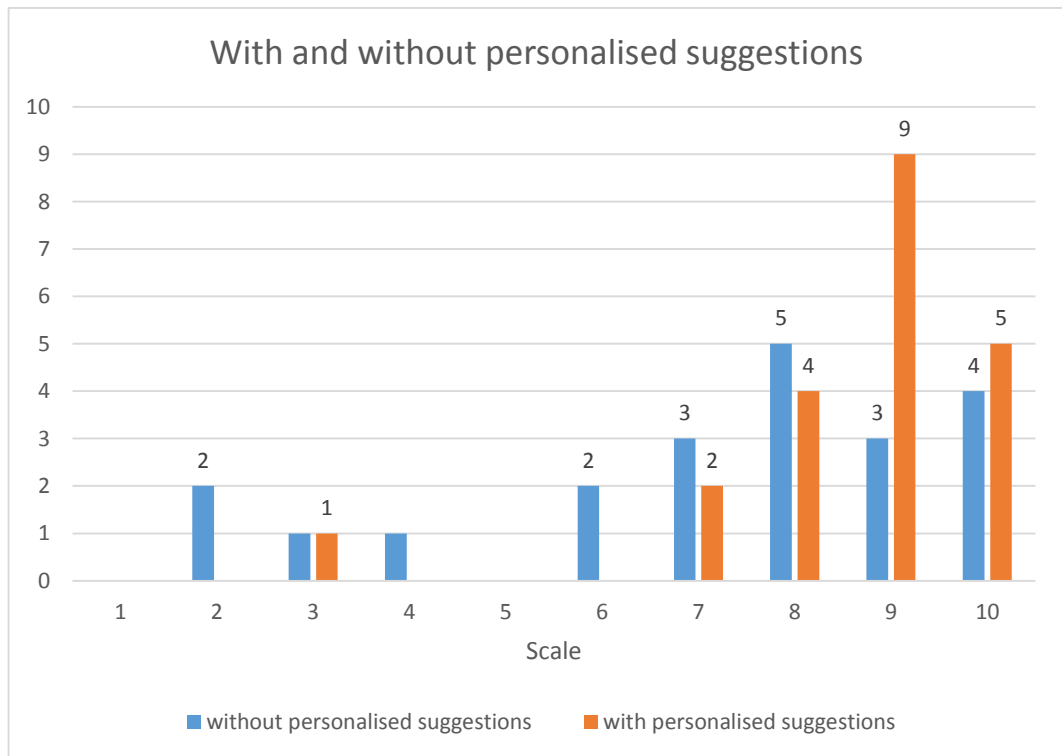


Figure 91. Rating PersonISM with and without personalised suggestions.

For the users that never read privacy policies and do not care about data disclosure, the PersonISM system does not change anything for them, as they can simply accept the terms and conditions offered to them by the service and continue to install the service as they would normally do.

Based on the answers from Step 3, the PersonISM system appears to be a very valuable and necessary tool for protecting one’s privacy. Given the abundance of information that can be accumulated digitally, it is important to give total control to the user to dictate

what information is disclosed to whom and when and under which specific terms. According to the users' responses, a mechanism such as the PersoNISM tool should be available on all personal devices, especially smart phones, which can be a source for a lot of personal information including contextual information about the user such as location, activity, heart rate, speed.

The necessity of a tool such as PersoNISM is also shown in the answers to question 7 that shows the willingness of users to pay a small amount in order to have such a tool at their disposal with which to protect their privacy.

One of the issues that were not raised during the evaluation of the PersoNISM system but which is often raised is scalability. The question is whether the system could cope with a large number of users. The PersoNISM system is designed so that each user will have their own instance of the system running on their behalf. Each instance will only have access to its user's data and will be providing its functionality only to that user. This is contrary to other architectures that are deployed in a centralised fashion that serve all the users of a pervasive platform.

7.3 *Summary*

The evaluation of the PersoNISM system demonstrated that many users are not reading the privacy policies of the services they use and many who do read them do not properly understand them but merely agree blindly to the terms and conditions imposed by the services. Users highlighted the "take it or leave it" issue; users have no option but to agree to the terms and conditions if they want to use the services.

The evaluation of the PersoNISM system showed that users welcome a mechanism that allows them to configure the way their data are handled by service providers. As expected, the manual configuration of all the privacy functionalities of the PersoNISM system tired users even though only four types of data were requested by the example services. This highlights the need for personalisation in the privacy policy negotiation process. After experiencing the personalised suggestions, users gave very positive feedback on their usefulness.

The full automation of the PersoNISM system was not received in the same positive light as the personalised suggestions. It is evident that users do not wish to relinquish total

control of the privacy protection to a system they do not fully trust to perform these tasks on their behalf as they want. There is no “one size fits all” solution to the privacy problem. That is why the personalisation functionality is an important feature to include in order to help the user protect their privacy on their own terms.

8 Conclusion

The issue of privacy in the digital world is complicated. It involves combining many different approaches that must work in harmony to have the desired effect. This thesis attempts to address the issue of using personalisation to automate a set of privacy protection techniques including privacy policy negotiation, identity management, identity selection and access control in a pervasive environment with the purpose of aiding users to make better decisions regarding the use and disclosure of their data. These privacy enhancing technologies cannot be performed in isolation. The result of a privacy policy negotiation drives the identity creation and selection processes. Without personalisation, the user has to perform all of these steps by themselves every time they install new services and privacy protection is inadequate for context information that changes constantly and where users want flexibility in disclosing data in a context-aware fashion. This thesis also demonstrates that there is only a superficial conflict between personalisation and privacy and that this conflict can be resolved by using personalisation to enhance privacy and using privacy enhancing technologies to enhance personalisation.

8.1 *Future work*

The evaluation results of the PersoNISM system showed that users welcomed the system and gave very good feedback. However, the following issues need to be further researched.

8.1.1 Need for more information

New types of information become available almost daily and are made available through new types of sensors that are embedded into mobile devices and systems in our environment. Services are created to take advantage of this information and provide certain functionalities to the end-user. Controlling access to that information as it keeps growing can become very frustrating to users. The process of privacy policy negotiation can become very tedious to perform when no applicable privacy preferences are present to guide the user. One solution would be to provide a number of templates of privacy preferences that can be downloaded from trusted third parties or assist communities and friends in sharing their privacy preferences to aid one another. Another solution would be to organise data items into groups according to their semantics in order to perform the privacy policy negotiation and the access control in a more efficient manner.

8.1.2 Including semantics to control identifiability

The PersoNISM system does not use any semantic information to handle the data it protects. By attaching semantic information to data, more sophisticated algorithms can be used to relate data to each other in the same manner as inference algorithms do. This information could be used to detect the level of identifiability of a user and alert them if their privacy is threatened. When a combination of data is about to be disclosed to a service that identifies a user as an individual, they can be alerted to take appropriate steps. This functionality could also enhance the identity creation process to ensure that identities are not linked back to the single user.

8.1.3 Monitoring service providers' activities

The PersoNISM system does not provide monitoring functionality that checks to ensure that service providers adhere to the agreed terms and conditions. Further research is required in this field to protect data after they have been disclosed. There should be transparency in the way that service providers acquire information, process and share it. This could be accomplished using a system of receipts, similar to a “chain of evidence” to show users how data was acquired. Such a mechanism would have a great impact on the willingness of users to disclose data to services.

8.1.4 Graphical User Interfaces

A usability study on the aesthetics of the graphical user interfaces used in the PersoNISM system could be conducted to improve their appearance. The GUIs shown in chapters 6 and 7 were designed to perform the evaluation of the PersoNISM system and demonstrate the functionalities such as the application of the privacy preferences and the privacy policy negotiation process. The aesthetics of a graphical user interface play an important role in attracting users and retaining them.

8.1.5 Storing user data

The PersoNISM system is designed to store the privacy preferences of the user on devices owned by the user. However, there are some issues such as data availability at all times for the mobile user and whether the user owns or has access to a device that is always available and can satisfy the needs of the PersoNISM software. A solution would be to engage the services of trusted third party security providers such as i-brokers in combination with a certification authority.

8.1.6 Federated Privacy Policy Negotiation

In an environment in which multiple services are combined to offer a composed service, the user would have to negotiate with all of the services separately which would be confusing to the user and would very likely drive users to avoid using such services because of the additional hassle. To avoid this, the component responsible for selecting which services to include in the service composition (such as a Service Discovery component) could use the privacy policies and the predefined set of options of each service as another criterion for service selection. It can collect the privacy policies and the predefined set of options from the available services and compare all privacy statements. For each privacy statement, it can find the option that can be satisfied by all services. Using this information it can produce a) a single privacy policy that can be used as a starting point to the negotiation with the user and b) a modified set of options that can be satisfied by the services that will be needed during the privacy policy negotiation with the user. The Negotiation Agent of one of the services could be elected to act as a delegate between the Negotiation Agents of the services included in the service composition and the Negotiation Client running on behalf of the user. This means that the delegated Negotiation Agent would be authorised to perform a negotiation process without input from the Negotiation Agents of the other services. During the negotiation process, the elected Negotiation Agent has at its disposal the modified set of options that it can use to negotiate with the Negotiation Client. In a successful negotiation, the delegated Negotiation Agent would need to inform the services in the composition of the result of the negotiation sending them the agreed Response Policy which they will have to adhere to when they receive the user's data.

8.2 *Key Contributions*

The most important requirements for designing a privacy protection system are giving the user control over the manner in which their data are disclosed to others and over what happens to the data after disclosure. An effective system must take its orders from the user and act accordingly. While it is very important for the user to have control over every decision about how their privacy is handled, it is not practical to prompt the user constantly for every decision. Hence, a successful system must also provide mechanisms to maintain a set of rules that reflect the user's decisions which can be re-used in similar circumstances. Setting preferences in the system is one of the ways that a user can instruct the system to act in a certain way. Context-aware user preferences give the user a higher

level of flexibility about how their data should be handled and disclosed under different circumstances. User preferences for privacy must be able to handle all the different types of personalisation that can be applied to the privacy enhancing technologies employed by the system. For every decision that the system asks the user to make, it has to have an appropriate user preference that can represent that decision as a rule for future use.

While giving the user the ultimate control, the system needs to provide a mechanism to learn from the user's decisions regarding their privacy protection and automatically create user preferences based upon those decisions. Therefore, there is a clear requirement for privacy preference learning.

The key contributions of this thesis can be summarised as follows:

- The key issues in the protection of privacy in a pervasive environment were identified through research into current industry practices, state-of-the-art in pervasive and ubiquitous systems, context-aware personalisation and privacy enhancing technologies and finally surveying of users.
 - The “take it or leave it” issue. The predominant notice-choice model disadvantages not only users who either reluctantly agree to fixed terms and conditions or are prevented from using a service. This also disadvantages service providers when users a) don't agree with the presented terms and conditions and elect not to use a service or b) are willing to disclose more information than requested.
 - Growing amount of information. As devices, sensors and services grow, so does the amount and quality of data that is accumulated in the system about the user, making it possible to monitor the user's activities in greater detail.
 - Users' inability to maintain knowledge about previous data disclosures to maintain the same level of data disclosure.
 - Users' reluctance to use privacy preserving technologies they do not understand or are too busy to concern themselves with.

- The deciding factors in disclosing information are the user's context, the type of information, the recipient of the information what happens to the information after it is released.
- Design of the PERSONalised Negotiation, Identity Selection and Management (PersonISM) system including the:
 - Design of a privacy policy negotiation protocol allowing both parties to the negotiation to demand or compromise on the disclosure processing and sharing of specific data.
 - Design of a mechanism to control the creation and selection of digital identities based on the intended use of the identity in a specific context.
 - Design of a context-dependent access control system aided by data obfuscation capabilities.
 - Design of a privacy preference model specifically tailored to aiding users in performing all the privacy enhancing technologies used in the PersonISM lifecycle using IF-THEN-ELSE rules.
 - Design of a rapid behaviour learning approach based on an IF-THEN-ELSE preference rule merging algorithm for acquiring privacy preferences, taking advantage of instant user feedback.
- Implementation of the personalised privacy policy negotiation functionality, the identity selection and context-dependent access control mechanisms in the PERSIST Personal Smart Space platform. Implementation of the PersonISM system as a privacy protecting framework for the SOCIETIES Cooperative Smart Space platform and subsequent use in live user trials. Both prototypes were demonstrated to EU project reviewers and received “excellent” status.
- Major survey of privacy, data disclosure practices and online user behaviour to assess the level of awareness of privacy of the average user and to assess the value of a system such as PersonISM.
- An evaluation of the PersonISM system by 21 users using real data as input.

8.2.1 Negotiating Privacy

Protecting Privacy is not just about controlling the disclosure of data but it is also about controlling what happens to the data after they have been disclosed. This is one of the hardest challenges when designing a privacy protection system. In a nutshell, a privacy policy is the tool used by service providers to state what type of data they need access to and what type of processing they will apply to the data after acquiring it. Currently, users that want to use a service must comply with the service privacy policy and terms and conditions. The obvious problem is that users have no option but to agree to all of the statements in the privacy policy if they want to use it. By allowing users to negotiate the terms and conditions in the privacy policy of the service, they are given better control over their privacy. Privacy policy negotiation can be an arduous task to perform manually. The system should employ appropriate mechanisms to automate this process as much as possible but at the same time it should acquire informed consent from the user about the decisions that need to be made during the privacy policy negotiation. First, the system should be able to analyse the terms of the privacy policy and present them to the user in a user friendly manner. User preferences can be very useful in automating privacy policy negotiation. A graphical user interface should be provided for the user to create preferences so that the system is able to perform the negotiation on their behalf. Moreover, the system should also employ preference learning techniques to learn preferences for automating the privacy policy negotiation process based on previous decisions of the user thus relieving the user of the burden of manually creating these preferences or constantly entering the same decision into the privacy policy negotiation forms.

There are different requirements for designing privacy policy negotiation preferences to other types of privacy preferences such as identity selection preferences or access control preferences. The privacy policy negotiation preference model does not need to be context dependent. The purpose of the privacy policy negotiation is for the user to be able to negotiate the manner in which the data are going to be processed, stored, shared and eventually deleted by the service that retrieves them. Therefore, privacy policy negotiation preferences must be flexible in order to allow the user to define different courses of action depending on the statements in the privacy policy.

Privacy policy negotiation has received some criticism over the years and fears have been expressed that it might lead to services offering incentives for users to give up their privacy. For example, a service could offer better quality of service to users that divulge more personal information. While this is the argument against privacy policy negotiation, the argument for enabling privacy policy negotiation is the fact that protecting privacy will become something that services will compete over to provide better quality of service than others.

8.2.2 Personalising the use of Digital Identities

The primary purpose of an Identity Management system operating in a pervasive environment is to provide users with multiple identities to represent themselves to services and entities in a network. An identity can be used to retrieve information from a context or a content management database. Identities can be associated with specific data records so that when a service uses the identifier of a user's identity to retrieve information, it can only see the data records associated with that identity, hence they will get access to only a partial view of the user's profile. Any system should ensure that the decision to associate a data attribute with an identity can only be made by the user that owns the identity. In some cases other input to this process will be necessary. For example, when the creation of an identity is triggered because none of the existing identities of the user are applicable for use with a service, the negotiation agreement with the service has to be used as input to selecting what types of data need to be associated with that identity to be applicable for use with that service. As the number of identities and the information stored in the context and content management databases grows, the user can become overwhelmed with the complexities of maintaining the identities and their data associations. By monitoring the use of the identities and their associations, user preferences can be created to indicate how sensitive a data attribute is and define parameters to be considered when associating this data attribute with an identity. Two different types of preferences are needed; one used to guide the identity creation process and the other used to select an identity to interact with a specific service.

As the user starts to use more and more services, the system can analyse the user's decisions about the use of each of their identities and learn preferences that define where each identity should be used or not used. Moreover, during the identity creation process, the user configures the new identity and associates the data they want to share using that

identity. The system should be able to monitor this process and create preferences that define under what circumstances a data attribute could be linked to an identity.

8.2.3 Controlling Data Disclosure

Any system with an access control mechanism employs some sort of rule model and rule engine to control data disclosure. Even though a user has agreed to allow access to some of his data during the privacy policy negotiation process, they should be allowed to deny access to any data attribute they want at any time they want. This requirement is important for data attributes that describe the context of the user and therefore the values of these data attributes change as the user goes about their daily routine. It is not very important for data attributes that do not change such as the user's name as once this information is disclosed, it does not matter how many times it is disclosed.

8.2.4 Proactive Privacy Protection

Context-dependent preferences define actions that should be implemented in different situations. A personalisation system that employs context-dependent preferences must be proactive in applying the appropriate outcome in each situation. Changes in the context of the user must be monitored, the affected preferences must be evaluated, and the corresponding outcomes must be applied. In pervasive systems, context and preference management systems use event management mechanisms to be notified of changes in the context and preferences of the user and adapt their behaviour accordingly. The same can be applied to privacy protection components. The goal of proactive personalisation of privacy is to proactively change the access permissions or change the identity with which a user uses a service at any particular moment according to the current situation of the user and the evaluated outcome of their privacy preferences. Consequently, one more requirement is that the privacy framework as well as the underlying system has to support changes to the access control permissions as well as reconfiguring a service with a different identity if it becomes necessary.

Appendix A – Online Questionnaire

Age

15-24 25-34 35-44 45-54 55-64 65-74

Education: High School, Higher Education,

Tick if you own the following devices:

Laptop, Mobile phone with GPS, Tablet, SatNav

1. How privacy aware do you think you are?

Range 1 to 10 where 1 is no privacy aware at all and 10 being very privacy aware.

2. Which of the following social networking sites do you use? (tick as appropriate)

- a. I don't use social networking sites
- b. Facebook
- c. Twitter
- d. LinkedIn
- e. Instagram
- f. Foursquare
- g. Pinterest
- h. Google+
- i. Bebo
- j. Flickr
- k. MySpace
- l. Hi5
- m. Care2
- n. ResearchGate
- o. Tumblr
- p. Other: Please specify

3. How often do you post on social networks on average?

- a. Multiple times a day
- b. Once a day
- c. 2-3 times a week
- d. Once a week
- e. 2-3 times a month

- f. Once a month
 - g. Less than once a month
4. Which of the following information have you disclosed on social networking sites?
- a. Full name
 - b. Date of Birth
 - c. Place of Birth
 - d. Place you live currently
 - e. Places you've lived in the past
 - f. Places you have visited
 - g. Education
 - h. Sexual orientation
 - i. Marital Status
 - j. Information on romantic relationships you've been involved in
 - k. Religious views
 - l. Political views
 - m. Current job
 - n. Previous jobs
 - o. Languages you speak
 - p. Family connections
 - q. Photos of you
 - r. Photos of you and your family
 - s. Photos of you and your friends
 - t. Your hobbies
 - u. Books you have read
 - v. Movies you have watched
 - w. Music you listen
 - x. Life events such as graduations, your wedding, buying a car or a house
 - y. Attending events such as concerts, museum exhibits, movies etc.
5. Do you know what the social networking sites are doing with the information you have disclosed?
- a. No, I have no idea what can happen to my data and I don't care.

- b. No, I have no idea what can happen to my data and that worries me but I don't feel I have any option except to not use social networking sites.
 - c. I have some knowledge of what they are allowed to do with my data and I don't care.
 - d. I have some knowledge of what they are allowed to do with my data and that worries me but I don't feel I have any option except to not use social networking sites.
 - e. Yes, I know exactly what they are allowed to do with my data and I don't have a problem with what they are allowed to do with the data.
 - f. Yes, I know exactly what they are allowed to do with my data and that worries me but I don't feel I have any option except to not use social networking sites.
6. Do you read the privacy policy and terms and conditions when joining social networking sites, registering for services such as Amazon, Gmail, Dropbox etc, or installing apps on your mobile phone or software on your computer?
- a. I always read them and I understand them.
 - b. I want to read them but I don't understand them so I don't bother.
 - c. I never read them because I don't care about my privacy.
 - d. I think it's a waste of time because I'm sure they don't abide by their privacy policies.
 - e. If your answer is not covered above, please answer in your own words.
7. Some social networking sites such as Facebook, offer users the option to configure privacy settings. For example, it's possible to hide some information from other people or Facebook apps. Have you ever used this feature?
- a. Yes.
 - i. Are you satisfied with this feature?
 - 1. Yes, it allows me to protect my privacy as I want to.
 - 2. No, I'm not satisfied (Please specify)
 - b. No.
 - i. Is that because:
 - 1. I didn't know this feature was available to me.

2. The social networking sites I am a member of, do not support this feature.
 3. I don't need to. The default settings cover my privacy needs.
 4. It's too much a hassle for me. I would like to use these features but I don't have the time to configure all these settings.
 5. Other (please explain)
8. Have you ever declined installing an application or registering for a service because you disagreed with a privacy policy and/or terms and conditions of a company?
- a. Yes
 - b. No

When installing an application or signing up for a service you are required to read the privacy policy and terms and conditions of the service and tick a box to indicate that you agree. Instead of a document, a privacy policy can be graphically shown using a tool such as the one showing in the picture below. A privacy policy negotiation is the process that allows you to negotiate with the service what data are going to be accessed by the service and the terms and conditions under which the data are going to be disclosed and further processed. Examples of conditions include sharing the data with other companies, how long the data will be kept by the service, your right to opt out of the service any time you wish, your right to delete any data the service holds about you etc.

Privacy Policy Negotiation with Lollipop Ltd

The information below shows what data will be used during your using of Lollipop Ltd services. Configure usage per your needs and click continue.

- [▶ name](#)
- [▶ age](#)
- [▼ GPS location](#)
 - Purpose: Your location will be tracked to offer you services nearby.
 - Actions:
 - Read
 - Write
 - Create
 - Delete
 - Conditions:
 - Share with 3rd parties Keep data for 1 week
 - Right to opt out
 - Decision: allow deny
- [▶ activity](#)

9. If you had the option to negotiate your privacy rights, would you use a tool such as the one showing in the picture?
- Yes
 - I don't understand what this picture is showing.
 - No (please explain the reasons below)

Context information is information that describes the environment and yourself. This includes current location (GPS coordinates), symbolic locations (home, work, gym, pub etc), activity (sleeping, walking, working, exercising, watching television etc), age, room temperature, light conditions, current weather. Most of this information is currently available through sensors and personal mobile devices.

10. Where you aware that this kind of information can exist in a digital form and used by Web services?
- Yes
 - No
11. Would you care if your apps on your mobile phone or your social networking sites had access to this information about you?
- Yes

- b. No
- c. It depends on what they would use this information for (please explain further)

12. Would you want to restrict access to this information?

- a. Yes
- b. No. (Please explain why)

13. Would you want the system to ask you to permit or deny access to this information every time an App or a social networking site requested this information from your personal devices?

- a. Yes, every time.
- b. No, that would become too much of a hassle. I already receive too many notifications.

Privacy Preferences (or settings) could be used to block access to this information under certain circumstances. For example, you may block specific services or apps to access your location information (both GPS and symbolic) when you are at specific locations or performing some activity.

14. Would you define such preferences to block access to your context information?

- a. Yes
- b. Yes but I would find it tedious to manually configure such preferences.
- c. No, I don't have the time to do that.

15. Would you find it useful if the system could learn your preferences automatically based on your previous access control requests so you wouldn't have to manually create them yourself?

- a. No, I wouldn't find it useful at all.
- b. I would find it useful but I would like to be able to see what preferences the system learnt on my behalf.
- c. I would find it very useful because I wouldn't have to bother responding to privacy notifications all the time.
- d. Other (please explain).

Read the following scenarios and answer the questions that follow.

Social network disclosures

While you are booking a flight, you are given the option to use a tool that can offer to seat you next to any friends in your social networks who happen to be on the same flight. You use the tool but you don't find any friends on the flight. On the day of the flight, you realise that one of your "social network friends" who you don't really want to talk to is sitting right next to you because they spotted you using the seat finder tool.

Location based services in combination with poor inference

Your friend is having an abortion so you go with her for support to the clinic. One of the apps on your phone which monitors your location records this information on your profile. A few years later, you are applying for a job but the person reviewing your application rejects it because of the apparent pro-abortion stance revealed by your profile. Needless to say, a different reason is given to you.

Location based services in combination with poor profiling

Arriving on an international flight, you are pulled out of the line by immigration officials, detained and interrogated for 24 hours because your physical characteristics match someone on their terrorist watch list and places you have recently visited, recorded by an application on your phone, match locations in which the terrorist has been spotted.

Health sensor-based service

On the advice of your doctor, you make use of a wrist band that monitors your heart rate along with a mobile app that transmits this data to a computer at your local hospital where it is stored along with the data of many other similar patients. Meanwhile the local police are investigating a series of serious crimes but getting nowhere. One of the victims works at the hospital and, suspecting one of the patients as being the perpetrator, makes all the data available to the police. Coincidentally, your data happens to indicate an increase in your heart rate at the time of every crime. With nothing else to go on, the police make you their prime suspect and start searching for evidence that can be used against you in court.

16. What do you think of the above scenarios?

17. Did you realise that disclosing information could have such consequences?

- a. No.
- b. Yes.

18. Have these scenarios raised your awareness about the privacy of your personal data?

- a. Not at all.
- b. A little.
- c. A lot.

19. Would you consider changing the way that your data are handled or disclosed after reading these scenarios?

- a. No.
- b. Maybe
- c. Yes.

Appendix B – Evaluation handout document

PersoNISM Trial Evaluation

This experiment evaluates the PersoNISM (Personalised Negotiation, Identity Selection and Management) tool in the context of pervasive systems. A pervasive system is a computing environment embedded with sensors and digital devices such as smart phones, tablets and laptops. These devices provide information (often sensitive information) about the user and their environment such as the user's current location, activity, people in the vicinity, current room temperature, light intensity etc. This and other information is collected, processed and meshed together to form the user's profile. Applications running on the user's devices can access this information to provide their services to their users. Pervasive systems are slowly starting to appear in our daily lives as sensors such as GPS and accelerometers are becoming standard features in smart phones and cars providing location based services to users. As more sensors are embedded in our environment, the more information about ourselves will be captured in a digital form which raises the issue of privacy.

Currently, applications provide the terms and conditions in a privacy policy to users and ask them to accept them. The PersoNISM tool allows the user to negotiate these terms and conditions to fit their privacy requirements and configure their digital identity to represent themselves to applications and services.

If you have any questions, please don't hesitate to ask them before continuing with the experiment.

Note that any information about you collected during this experiment will be used for analysis and will be deleted afterwards.

Step 1. You will be timed for this step.

Use an Internet browser and find and read Google's privacy policy.

Use an Internet browser and find and read BBC's privacy policy.

Use an Internet browser and find and read Heriot Watt University's privacy policy.

Answer the following questions:

1. Do you use the BBC and Google websites and HWU services regularly?
2. Have you ever read their respective privacy policies before?
3. If the answer to the above question is no, explain why

4. Before you read their privacy policies
 - a. did you know what information BBC, Google and HWU collect through your interactions with their services?
 - b. did you know what they do with the information they collect from you?
5. After reading their privacy policies
 - c. do you know exactly what information BBC, Google and HWU collect through your interactions with their services? (list the information they collect)
 - d. do you know what they do with the information they collect from you? (write how they use your information)
6. After reading the privacy policies, can you tell how you can make changes (remove, edit) to the information that was collected from you from their systems?

Google:

BBC:

HWU:

7. How long will the information that was collected be kept in their systems?

Google:

BBC:

HWU:

8. Do you know whether information you have disclosed to these companies has been shared with other companies? (yes/no)
9. Do you know how to stop these companies from sharing your information with other companies (yes/no)?
10. Do you know if your information has been processed further and combined with other information to infer more information about yourself (yes/no)?
11. Do you know how to tell the company to stop doing that (yes/no)?

Go to step 2.

Step 2.

- a) Enter your details in the form and click “Save Details”.
- b) In the Window titled “PersonISM Evaluation Tool”, go to the “My Data” menu and click on “Trust Settings”. In the window that opens, adjust the current trust values to indicate your personal trust level for each of the three companies with regard to how much you trust them to safeguard your data and respect your privacy. When you are done, click “Save” and close the window.
- c) There should be 4 services showing in the “Available Apps” section. Click on “Google Venue Finder” and then click “install”.

The next window shows you Google’s privacy policy in a digital format allowing you to make changes to the terms and conditions (T&Cs) to fit your privacy requirements. Make your changes to each of the data items requested by Google by clicking the “Next” and “Back” buttons available at the bottom of the page. You can reset your changes in each of the data items by clicking the “Reset changes” button. This will only reset the changes you made to the data item you are currently viewing. When you are finished, click “Continue”.

Your negotiation with Google begins. Google responds with an updated privacy policy indicating if they can satisfy your requests. Requests that cannot be satisfied fully are marked with a warning. If you are happy with Google’s response, click “Continue”, otherwise click “Cancel”.

Assuming you have clicked “Continue”, the next window that pops up allows you to create an identity to interact with Google. Your new identity must be linked with the data that is requested by Google. In this case, the data are limited to “name”, “email”, “birthday” and “locationSymbolic” (Symbolic locations indicate locations with semantic meaning such as a postal address as opposed to location using GPS coordinates). Clicking on the items on the left hand side will update the current values on the right side that are currently available in your profile. Select a value you want to link with your new identity and click “Add selected attribute” or add a new attribute of the selected data type and type a new value. The new value will appear in the left hand. At the bottom of this page, you will see the list of data you have linked with your new identity. Don’t forget to type a name (username) for your new identity at the top. When you are finished, click “OK >>”. Google Venue Finder should now appear in the “Installed Apps” section.

Answer the following questions:

1. On a scale of 1 to 10, indicate the level of understanding of the terms and conditions in each form presented to you: (1 is you don't understand at all, 10 being you understood fully)

Text: 1 2 3 4 5 6 7 8 9 10

PersoNISM: 1 2 3 4 5 6 7 8 9 10

2. On a scale of 1 to 10, how valuable is it to specify the terms and conditions on a per data and per service basis? (where 1 is not valuable at all and 10 being extremely valuable)

1 2 3 4 5 6 7 8 9 10

3. On a scale of 1 to 10, how happy would you be to configure the terms and conditions for every data type and every service? (1 being not happy at all and 10 being very happy)

1 2 3 4 5 6 7 8 9 10

Go to Step 3.

Step 3.

- a) In the Applications window, click on the BBC News app and click “install”. In the Privacy Policy Negotiation form, you can now make use of the “Get Personalised Suggestions” functionality. When you click it, the terms and conditions are configured based on your previous negotiation. You can reset the changes by clicking “Restore changes”. Configure the terms and conditions suggested for every data type requested as you wish and click “Continue”.
 - b) In the next form, review the BBC’s changes and if you are happy with the changes, click “Continue”. Otherwise, click “cancel”. By clicking cancel, you abort the negotiation and installation of the service.
 - c) Assuming you clicked “Continue”, the next window asks you to select an identity to interact with BBC. You can create a new identity if you wish or use the same identity you used to interact with Google. If you use the same identity, the data that you linked with this identity and you made available to Google, will also become available to BBC. If you want different data (for example, a different email address) to be disclosed to the BBC, create a new identity.
 - d) If you decided to create a new identity, you can make use of the “Get recommended attributes” functionality which will show you what attributes can be linked with that identity according to your previous activity and the trust you assigned to the services in the previous step. When you are happy with your new identity, click “OK >>”.
 - e) In the main window, go to the menu “Identities” and click on “Identities Viewer”. You can see the identities you created and the corresponding values of each data type linked to each identity. After you are done, close the Identities viewer window
 - f) In the main window, click on “BBC Weather” and click install. The system should now have enough information to negotiate on your behalf and select the appropriate identity. Click yes.
- Answer the following questions.
1. On a scale of 1 to 10, did you feel that the personalised suggestions were useful?
(1 being not useful at all and 10 being very useful)

1 2 3 4 5 6 7 8 9 10

2. On a scale of 1 to 10, how much would you trust the system to perform this process on your behalf without your involvement in later negotiations? (1 indicating no trust in the system at all and 10 indicating complete trust in the system)

1 2 3 4 5 6 7 8 9 10

3. On a scale of 1 to 10, how likely would you be to configure the terms and conditions for every data type and every service with the help of personalised suggestions? (1 being very unlikely and 10 being highly likely)

1 2 3 4 5 6 7 8 9 10

Go to step 4.

Step 4.

- a) Click on any of the installed applications. On the application window, click on “Login in” and login with the identity available.
- b) As soon as you login, the application retrieves your data from the system. On the right hand side of the main window of the evaluation tool, you will find a notification section. Access control notifications will appear there to ask you if you want to disclose your data and data obfuscation notifications to ask you how you want to obfuscate your data. Use the slider to see examples of different data obfuscation levels and select the one you want to use. The service will try to retrieve all data types (name, email, birthday and locationSymbolic) from your profile. Configure the obfuscation levels for each one. Go back to the application window to see the retrieved data.
- c) Go back to the application window and click on “Update profile”. Notifications in the main window will appear in a personalised format with a countdown timer. If the countdown timer reaches 0, the system will apply the recommended access permission and data obfuscation level to that item.
- d) In the main window, go to menu “My Data” and click on “profile information”. In the search box enter “locationSymbolic”. Change the value from EM1.69 to EM1.24 to simulate a location change (you walking from the puma lab to the student office). The application will try to retrieve your new location and a non-timed notification will appear in the notifications area. Click as you wish.

Answer the following questions:

1. Taking into account all the steps you did, on a scale of 1 to 10 indicate if using this tool would make you take your privacy more seriously. (1 indicating that it would make you take it less seriously, 5 indicating no change and 10 indicating it would make you take your privacy very seriously)

1 2 3 4 5 6 7 8 9 10

2. On a scale of 1 to 10, indicate how useful you would find having such control of your personal data. (1 being not useful at all and 10 being very useful)

1 2 3 4 5 6 7 8 9 10

3. On a scale of 1 to 10, indicate how necessary you think it is to have tools like this to control your privacy. (1 not necessary at all, 10 being essential to your privacy)

1 2 3 4 5 6 7 8 9 10

4. Do you think service providers (such as google, facebook, websites) should be forced by law to give YOU control of your data? (yes/no)

5. Do you believe that personal devices such as mobile phones and tablets should have built in privacy control mechanisms such as the one you experimented with? (yes/no)

6. As our world becomes more and more pervasive do you believe that privacy protection is becoming a bigger problem? (1 indicating you don't believe it's becoming a bigger problem, 10 indicates you think it's a very big issue).

1 2 3 4 5 6 7 8 9 10

7. If tools such as the ones you have just experimented with were provided at a small cost, would you be prepared to pay for them to protect your privacy? (1 indicates you would never pay, 10 being you would definitely pay)

1 2 3 4 5 6 7 8 9 10

8. Finally, can you express in a few words what would be your biggest worry regarding your privacy in a pervasive environment?

End of experiment.

Please write any comments you wish to add.

Appendix C – PersoNISM code repository

The source code for the PersoNISM system can be retrieved from the following Github repository:

https://github.com/EPapadopoulou/SOCIETIES-Platform/tree/Student_Trial_Eliza

Publications

Journal Publications

- [1] Papadopoulou, E., McBurney, S., Taylor, N. K., Williams, H., & Abu Shaaban, Y. (2009). User preferences to support privacy policy handling in pervasive/ubiquitous systems. *International Journal on Advances in Security*, 2 (1), 62-71.
- [2] Papadopoulou, E., Gallacher, S., Taylor, N. K., & Williams, M. H. (2012). A Personal Smart Space approach to realising Ambient Ecologies. *Pervasive and Mobile Computing*, 8(4), 485-499. 10.1016/j.pmcj.2011.10.008.
- [3] Gallacher, S., Papadopoulou, E., Taylor, N. K., & Williams, M. H. (2013). Learning User Preferences for Adaptive Pervasive Environments. *ACM Transactions on Autonomous and Adaptive Systems*, 8(1), [5]. 10.1145/2451248.2451253.
- [4] Gallacher, S., Papadopoulou, E., Abu-Shaaban, Y., Taylor, N. K., & Williams, M. H. (2014). Dynamic context-aware personalisation in a pervasive environment. *Pervasive and Mobile Computing*, 10(PART B), 120-137. 10.1016/j.pmcj.2012.11.002
- [5] Liampotis, N., Roussaki, I., Kalatzis, N., Papadopoulou, E., Gonçalves, J., Papaioannou, I. and Sykas, E. (2014). Context-Sensitive Trust Evaluation in Cooperating Smart Spaces. Springer New York, [online] pp.187-201. Available at: http://dx.doi.org/10.1007/978-1-4939-1887-4_13.
- [6] Bental, D.S., Papadopoulou, E., Taylor, N.K., Williams, M.H., Blackmun, F., Ibrahim, I., Lim, M., Mimitsoudis, I., Whyte, S. & Jennings, E. (2015). Smartening up the Student Learning Experience with Ubiquitous Media, *ACM Transactions on Multimedia Computing, Communications and Applications*, to appear.
- [7] Liampotis, N., Papadopoulou, E., Kalatzis, N., Roussaki, I. G., Kosmides, P., Sykas, E. D., Bental, D., & Taylor, N. K. (2016). Tailoring Privacy-Aware Trustworthy Cooperating Smart Spaces for University Environments. In A. Panagopoulos (Ed.), *Handbook of Research on Next Generation Mobile Communication Systems* (pp. 410-439). Hershey, PA: Information Science Reference. doi:10.4018/978-1-4666-8732-5.ch016.

Conference Publications

- [8] Papadopoulou, E. Stobart, A., Taylor, N.K. & Williams, M.H., (2015). Enabling Data Subjects to Remain Data Owners. KES-AMSTA 2015 session on Business Model Innovation and Disruptive Technologies, Sorrento ITALY. In Agent and Multi-Agent Systems - Technology and Applications: 2015, Smart Innovation, Systems and Technologies Volume 38, 2015, pp 239-248. Eds. G Jezic, R J Howlett and L C Jain. Springer, Switzerland. ISBN 978-3-319-19727-2. Springer doi:10.1007/978-3-319-19728-9_20.
- [9] Papadopoulou, E., Williams, H., Gallacher, S., & Taylor, N. K. (2006). Redirecting Communication in a Pervasive System. In Exploiting the knowledge economy: issues, applications and case studies. (pp. 1688-1694).
- [10] Yang, Y., Williams, H., Taylor, N., McBurney, S., & Papadopoulou, E. (2006). Handling personalized redirection in a wireless pervasive computing system with different approaches to identity. In 2006 1st International Symposium on Wireless Pervasive Computing. (Vol. 2006, pp. 441-446). 10.1109/ISWPC.2006.1613665.
- [11] Williams, H., Yang, Y., Taylor, N., McBurney, S., Papadopoulou, E., Mahon, F., & Crotty, M. (2006). Personalized dynamic composition of services and resources in a wireless pervasive computing environment. In 2006 1st International Symposium on Wireless Pervasive Computing. (Vol. 2006, pp. 377-382).
- [12] Williams, H., Papadopoulou, E., Taylor, N., McBurney, S., & Dolinar, K. (2007). Conflict between privacy and personalisation in a pervasive service environment. In Proceedings of the 3rd IASTED International Conference on Advances in Computer Science and Technology, ACST 2007. (pp. 176-181).
- [13] McBurney, S., Williams, H., Taylor, N. K., & Papadopoulou, E. (2007). Managing user preferences for personalization in a pervasive service environment. In Proceedings of Third Advanced International Conference on Telecommunications, AICT 2007. (pp. 33:1-6). 10.1109/AICT.2007.27.
- [14] Papadopoulou, E., McBurney, S., Taylor, N., Williams, H., & Bello, G. L. (2008). Adapting stereotypes to handle dynamic user profiles in a pervasive system. In Proceedings of the 4th IASTED International Conference on Advances in Computer Science and Technology, ACST 2008. (pp. 7-12).
- [15] Papadopoulou, E., McBurney, S., Taylor, N. K., Williams, H., Dolinar, K., & Neubauer, M. (2008). Using user preferences to enhance privacy in pervasive

- systems. In 3rd International Conference on Systems, ICONS 2008. (pp. 271-276). 10.1109/ICONS.2008.46. Best Paper Award.
- [16] Papadopoulou, E., McBurney, S., Taylor, N., Williams, M., (2008). "Using Personalization to Support Privacy in Ubiquitous Systems", Poster supplement for The 10th International Conference on Ubiquitous Computing (Ubicomp), South Korea, September 2008.
- [17] Papadopoulou, E., McBurney, S., Taylor, N., & Williams, H. (2008). Linking privacy and user preferences in the identity management for a pervasive system. In Proceedings - 2008 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2008. (pp. 192-195). 10.1109/WIIAT.2008.331.
- [18] Papadopoulou, E., McBurney, S., Taylor, N., & Williams, H. (2008). A dynamic approach to dealing with user preferences in a pervasive system. In Proceedings of the 2008 International Symposium on Parallel and Distributed Processing with Applications, ISPA 2008. (pp. 409-416). 10.1109/ISPA.2008.32.
- [19] McBurney, S., Papadopoulou, E., Taylor, N., & Williams, H. (2008). Adapting pervasive environments through machine learning and dynamic personalization. In Proceedings of the 2008 International Symposium on Parallel and Distributed Processing with Applications, ISPA 2008. (pp. 395-402). 10.1109/ISPA.2008.63.
- [20] Dolinar, K., Papadopoulou, E., Liampotis, N., & Abu-shaaban, Y. (2009). Protecting the privacy of personal smart spaces. Proceedings of the PERSIST Workshop on Intelligent Pervasive Environments, AISB 2009 Convention (AISB'09), Edinburgh, Scotland, pp 33-40.
- [21] Liampotis, N., Roussaki, I., Papadopoulou, E., Abu-Shaaban, Y., Williams, H., Taylor, N.K., McBurney, S., & Dolinar, K. (2009). A privacy framework for personal self-improving smart spaces. In Proceedings - 12th IEEE International Conference on Computational Science and Engineering, CSE 2009 - 2009 IEEE International Conference on Privacy, Security, Risk, and Trust, PASSAT 2009. (Vol. 3, pp. 444-449). 10.1109/CSE.2009.148
- [22] Papadopoulou, E., McBurney, S., & Williams, M. H. (2009). A Model for Personalised Communications Control in Pervasive Systems. In Proceedings of the IASTED International Conference (Vol. 1, No. 9, p. 200).
- [23] McBurney, S., Papadopoulou, E., Taylor, N., & Williams, H. (2009). Implicit adaptation of user preferences in pervasive systems. In Proceedings of the 4th

- International Conference on Systems, ICONS 2009. (pp. 56-62). 10.1109/ICONS.2009.19.
- [24] McBurney, S., Papadopoulou, E., Taylor, N., & Williams, H. (2009). User preference management in a pervasive system should be a trusted function. In Proceedings of the IASTED International Conference on Advances in Computer Science and Engineering, ACSE 2009. (pp. 77-82).
- [25] McBurney, S., Taylor, N., Williams, H., & Papadopoulou, E. (2009). Giving the user explicit control over implicit personalisation. In Procs. of Workshop on Intelligent Pervasive Environments (under AISB'09), Edinburgh, Scotland.
- [26] McBurney, S., Papadopoulou, E., Taylor, N., Williams, H., Abu-Shaaban, Y. (2009). Comparing Two Different Architectures for Pervasive Systems from the Viewpoint of Personalisation. In Proc. eChallenges (e2009), IOS Press, ISBN 978-1-905824-13-7.
- [27] Papadopoulou, E., Abu-Shaaban, Y., Gallacher, S., Taylor, N., & Williams, H. (2010). Two approaches to handling proactivity in pervasive systems. Communications in Computer and Information Science, 54, 64-75. 10.1007/978-3-642-12035-0_8.
- [28] Papadopoulou, E., Gallacher, S., Taylor, N. K., & Williams, H. (2010). Personal smart spaces as a basis for identifying users in pervasive systems. In Proceedings - Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing in Conjunction with the UIC 2010 and ATC 2010 Conferences, UIC-ATC 2010. (pp. 88-93). 10.1109/UIC-ATC.2010.73
- [29] Gallacher, S. M., Papadopoulou, E., Taylor, N. K., & Williams, M. H. (2010). Putting the 'Personal' into Personal Smart Spaces. In Proc. of Pervasive Personalisation Workshop, Pervasive (Vol. 2010, pp. 10-17).
- [30] Gallacher, S., Papadopoulou, E., Taylor, N. K., Williams, M. H., & Blackmun, F. R. (2012). Linking between personal smart spaces. In Mobile and Ubiquitous Systems: Computing, Networking, and Services (pp. 401-408). Springer Berlin Heidelberg.
- [31] Gallacher, S., Papadopoulou, E., Taylor, N.K., Blackmun, F.R., Williams, M.H., Roussaki, I., Kalatzis, N., Liampotis, N., & Zhang, D. (2011). Personalisation in a system combining pervasiveness and social networking. In 20th International Conference on Computer Communications and Networks (ICCCN 2011). Maui.
- [32] Gallacher, S., Papadopoulou, E., Taylor, N. K., & Williams, H. (2011). User recognition and memory support in a pervasive system. 146-152. Paper presented at

- 6th International Conference on Pervasive Computing and Applications, Port Elizabeth, South Africa.10.1109/ICPCA.2011.6106494.
- [33] Papadopoulou, E., Gallacher, S., Taylor, N. K., & Williams, H. (2012). Personalizing the User's Physical Environment in a Pervasive System. In W. Assawinchaichote, & M. H. Hamza (Eds.), Proceedings 2nd IASTED Asian Conference on Modelling, Identification and Control (AsiaMIC 2012). ACTA Press. 10.2316/P.2012.770-013.
- [34] Papadopoulou, E., Gallacher, S., Blackmun, F., Taylor, N. K., & Williams, M. H., (2012). Intelligent Systems for Pervasive Computing and Social Networking. Poster at the 1st International Conference on Intelligent Systems and Applications: INTELLI 2012. Chamonix FRANCE.
- [35] Gallacher, S., Papadopoulou, E., Taylor, N. K., & Williams, M. H. (2012). The challenge of preparational behaviours in preference learning for ubiquitous systems. In B. O. Apduhan, C. H. Hsu, T. Dohi, K. Ishida, L. T. Yang, & J. Ma (Eds.), 2012 9th International Conference on Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC) . (pp. 233-239). Los Alamitos: IEEE. 10.1109/UIC-ATC.2012.148.
- [36] Gallacher, S., Papadopoulou, E., Taylor, N. K., Blackmun, F. R., & Williams, M. H. (2012). Intelligent Systems that Combine Pervasive Computing and Social Networking. In B. O. Apduhan, C. H. Hsu, T. Dohi, K. Ishida, L. T. Yang, & J. Ma (Eds.), 2012 9th International Conference on Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC). (pp. 151-158). LOS ALAMITOS: IEEE. 10.1109/UIC-ATC.2012.99.
- [37] Gallacher, S., Papadopoulou, E., Blackmun, F., Taylor, N. K., & Williams, M. H., (2012). A Community Enhanced Personalisation System for Digital and Physical Social Spaces. Poster at Pervasive 2012. Newcastle, UK.
- [38] Doolin K, Roussaki I, Roddy M, Kalatzis N, Papadopoulou E, Taylor NK, Liampotis N, McKitterick D, Jennings E, Kosmides P (2012) SOCIETIES: where pervasive meets social. In: Future internet assembly book 2012, pp 30–41
- [39] Taylor, N. K., Papadopoulou, E., Lim, M. Y., Skillen, P., Blackmun, F. R., & Williams, M. H. (2013). Congestrian: Monitoring pedestrian traffic and congestion. In UbiComp 2013 Adjunct - Adjunct Publication of the 2013 ACM Conference on Ubiquitous Computing. (pp. 1355-1358). 10.1145/2494091.2499222

- [40] Taylor, N. K., Papadopoulou, E., Gallacher, S., & Williams, H. M. (2013). Is There Really a Conflict Between Privacy and Personalisation?. In *Information Systems Development* (pp. 1-9). Springer New York.
- [41] Papadopoulou, E., Taylor, N. K., Williams, M. H., & Gallacher, S. (2013). Learning user preferences in a system combining pervasive behaviour and social networking. In *2013 8th International Conference on Information Technology in Asia - Smart Devices Trend: Technologising Future Lifestyle, Proceedings of CITA 2013*. (pp. 1-7). [6637576] IEEE. 10.1109/CITA.2013.6637576.
- [42] Papadopoulou, E., Gallacher, S., Taylor, N. K., Williams, M. H., & Blackmun, F. (2013). Context-aware user preferences in systems for pervasive computing and social networking. In *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*. (Vol. 109 , pp. 10-17). (Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering; Vol. 109 LNICST). 10.1007/978-3-642-36642-0_2.
- [43] Papadopoulou, E., Gallacher, S., Taylor, N. K., & Williams, M. H. (2013). Learning user behaviour in a pervasive social networking system. In *Proceedings of the 8th IASTED International Conference on Advances in Computer Science, ACS 2013*. (pp. 357-364). 10.2316/P.2013.801-013
- [44] Papadopoulou, E., Gallacher, S., Taylor, N. K., Williams, M. H., Blackmun, F. R., Ibrahim, I. S., ... Whyte, S. (2014). Combining pervasive computing with social networking for a student environment. In *Conferences in Research and Practice in Information Technology Series*. (Vol. 152, pp. 11-19). Australian Computer Society.
- [45] Papadopoulou, E., Taylor, N. K., Williams, M. H., & Gallacher, S. (2014). Personalizing system behaviour in a pervasive social networking system. In *2014 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. (pp. 484-488). New York: IEEE. 10.1109/PerComW.2014.6815254.

References

- [1] Weiser, M. (1991). The computer for the 21st century. *Scientific american*, 265(3), 94-104.
- [2] ICO: "Privacy by Design", expert report by the Enterprise Privacy Group (www.privacygroup.org) for the Information Commissioner's Office (ICO), (pub) ICO, UK, November 2008.
- [3] ICO: "Empowering individuals to control their personal information", report of the Work Group on User-Centric Identity (and Personal Information) Management for the Information Commissioner's Office (ICO), (pub) ICO, UK, December 2008.
- [4] Wikipedia, (2015). Targeted advertising. [online] Available at: http://en.wikipedia.org/wiki/Targeted_advertising [Accessed 31 Aug. 2015].
- [5] Aleecia, M. M., & Cranor, L. F. (2008). The cost of reading privacy policies. *Information System: A journal of law and policy for the information society* [online]. Available at: www.is-journal.org.
- [6] Cranor, L. F. (2012). Necessary but not sufficient: Standardized mechanisms for privacy notice and choice. *J. on Telecomm. & High Tech. L.*, 10, 273.
- [7] Satyanarayanan, M. (2001). Pervasive computing: Vision and challenges. *Personal Communications, IEEE*, 8(4), 10-17.
- [8] Carnegie Mellon University Aura project Website. [online]. Available at: <http://www.cs.cmu.edu/~aura/> [Accessed 31 Aug. 2015].
- [9] Garlan, D., Siewiorek, D. P., Smailagic, A., & Steenkiste, P. (2002). Project aura: Toward distraction-free pervasive computing. *Pervasive Computing, IEEE*, 1(2), 22-31.
- [10] MIT Oxygen Project Website. [online]. Available at: <http://oxygen.csail.mit.edu/> [Accessed 31 Aug. 2015].
- [11] Hanssens, N., Kulkarni, A., Tuchida, R., & Horton, T. (2002). Building Agent-Based Intelligent Workspaces. In *International Conference on Internet Computing* (pp. 675-681).
- [12] De Carolis B., Pizzutilo S., Palmisano I. And Cavalluzzi A. (2003). A Personal Agent Supporting Ubiquitous Computing. *UM'03 9th International Conference on User Modeling*.
- [13] Román, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R. H., & Nahrstedt, K. (2002). A middleware infrastructure for active spaces. *IEEE pervasive computing*,

References

- 1(4), 74-83 [online]. Available at: <http://dx.doi.org/10.1109/MPRV.2002.1158281> [Accessed 31 Aug. 2015].
- [14] Roman, M., Hess, C., Ranganathan, A., Madhavarapu, P., Borthakur, B., & Viswanathan, P. et al. (2001). GaiaOS: An Infrastructure for Active Spaces. University Of Illinois At Urbana-Champaign, IL, USA.
- [15] Chetan, S., Al-Muhtadi, J., Campbell, R., & Mickunas, M. D. (2005, January). Mobile gaia: a middleware for ad-hoc pervasive computing. In Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE (pp. 223-228). IEEE.
- [16] EasyLiving Project HomePage [online]. Available at: <http://www.cs.washington.edu/mssi/tic/intros/Shafer/> [Accessed 31 Aug. 2015].
- [17] Brumitt, B., Meyers, B., Krumm, J., Kern, A., & Shafer, S. (2000, January). Easyliving: Technologies for intelligent environments. In Handheld and ubiquitous computing (pp. 12-29). Springer Berlin Heidelberg.
- [18] Jørstad, I., van Do, T., Dustdar, S. (2004). Personalisation of Future Mobile Services. 9 International Conference on Intelligence in service delivery Networks, Bordeaux, France, 18-21.
- [19] O'Looney, J. (2001). Personalization of Government Internet Services. In Digital Government Conference Proceedings. Los Angeles, CA.
- [20] McCarthy, J. F. (2001). The virtual world gets physical: Perspectives on personalization. *Internet Computing, IEEE*, 5(6), 48-53. [online]. Available at: <http://dx.doi.org/10.1109/4236.968831> [Accessed 31 Aug. 2015].
- [21] Kobsa, A. (2001). Generic user modelling systems. *User Modelling and User-Adapted Interaction* 11(1-2) 49-63.
- [22] Yang, Y., Williams, M. H., Taylor, N., McBurney, S., & Papadopoulou, E. (2006, January). Handling personalized redirection in a wireless pervasive computing system with different approaches to identity. In *Wireless Pervasive Computing, 2006 1st International Symposium on* (pp. 6-pp). IEEE.
- [23] EU FP6 IST Mobilife on Cordis. [online]. Available at: http://cordis.europa.eu/project/rcn/71853_en.html [Accessed 31 Aug. 2015].
- [24] Sutterer, M., Coutand, O., Droegehorn, O., David, K., & Der Sluijs, K. (2007, January). Managing and delivering context-dependent user preferences in ubiquitous computing environments. In *Applications and the Internet Workshops, 2007. SAINT Workshops 2007. International Symposium on* (pp. 4-4). IEEE [online]. Available at: <http://dx.doi.org/10.1109/SAINT-W.2007.60> [Accessed 31 Aug. 2015].

References

- [25] Nurmi, P., Salden, A., Lau, S., Suomela, J., Sutterer, M., & Millerat, J. et al. (2006). A System for Context-Dependent User Modeling. Springer Berlin Heidelberg, 4278, 1894-1903. [online]. Available at: http://dx.doi.org/10.1007/11915072_97 [Accessed 31 Aug. 2015].
- [26] 3GPP. Generic User Profile (GUP), 3GPP TR 23 941 v6.0.0. [online]. Available at: <http://www.3gpp.org/>, [Accessed 31 Aug. 2015].
- [27] ETSI Personalization and User Profile Management Technical Specification. ETSI TS 102 747 V1.1.1, 2009 [online]. Available at: http://www.etsi.org/deliver/etsi_ts/102700_102799/102747/01.01.01_60/ts_102747v010101p.pdf [Accessed 31 Aug. 2015].
- [28] W3C: 2007, "Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 2.0", [online]. Available at: <http://www.w3.org/TR/2007/WD-CCPP-struct-vocab2-20070430/>, [Accessed 31 Aug. 2015].
- [29] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994, October). GroupLens: an open architecture for collaborative filtering of netnews. In Proceedings of the 1994 ACM conference on Computer supported cooperative work (pp. 175-186). ACM.
- [30] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (pp. 285-295). ACM.
- [31] Breese, J. S., Heckerman, D., & Kadie, C. (1998, July). Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence (pp. 43-52). Morgan Kaufmann Publishers Inc.
- [32] Lam, S. K., & Riedl, J. (2005). Privacy, shilling, and the value of information in recommender systems. In Proceedings of User Modeling Workshop on Privacy-Enhanced Personalization (pp. 85-92).
- [33] Hong, J. I., & Landay, J. A. (2004, June). An architecture for privacy-sensitive ubiquitous computing. In Proceedings of the 2nd international conference on Mobile systems, applications, and services (pp. 177-189). ACM.
- [34] Westin, A. F. (1968). Privacy and freedom. *Washington and Lee Law Review*, 25(1), 166.
- [35] Rao, J.R., and P. Rohatgi (2000). Rao, J. R., & Rohatgi, P. (2000, August). Can pseudonymity really guarantee privacy?. In *USENIX Security Symposium*. [online]

References

- Available at: <http://www.freehaven.net/anonbib/cache/rao-pseudonymity.pdf>
[Accessed 31 Aug. 2015].
- [36] Altman, I. (1977). Privacy Regulation: culturally universal or culturally specific?. *Journal of Social Issues*, 33(3), 66-84.
- [37] Altman, I. (1975). *The environment and social behavior: Privacy, personal space, territory and crowding*. Monterey, CA: Brooks/Cole.
- [38] Petronio, S. (2012). *Boundaries of Privacy: Dialectics of disclosure*. Suny Press.
- [39] Borcea-Pfitzmann, K.; Pfitzmann, A. & Berg, M. (2011), Privacy 3.0: = Data Minimization + User Control + Contextual Integrity, *IT - Information Technology* 53 (1), 34-40.
- [40] Garfinkel, S. (2000). *Database nation: the death of privacy in the 21st century*. "O'Reilly Media, Inc."
- [41] Brar, A. and Kay, J. (2005). *Privacy and Security in Ubiquitous Personalized Applications*. User Modelling Workshop on Privacy-Enhanced Personalization, Edinburgh, UK.
- [42] Lahlou, S., Langheinrich, M. and Rucker, C. (2005). Privacy and Trust Issues with Invisible Computers. *Commun. ACM*, [online] 48(3), pp.59--60. Available at: <http://doi.acm.org/10.1145/1047671.1047705>.
- [43] Campbell, A. J. (1997). Relationship marketing in consumer markets: A comparison of managerial and consumer attitudes about information privacy. *Journal of Interactive Marketing*, 11(3), 44-57.
- [44] Nowak, G. J., & Phelps, J. (1995). Direct marketing and the use of individual-level consumer information: Determining how and when "privacy" matters. *Journal of Direct Marketing*, 9(3), 46-60.
- [45] Wang, H., Lee, M. K., & Wang, C. (1998). Consumer privacy concerns about Internet marketing. *Communications of the ACM*, 41(3), 63-70.
- [46] Punie, Y.; Friedewald, M.; Lindner, R. et al. (2005). *Safeguards in a World of Ambient Intelligence (SWAMI): Dark Scenarios on Ambient Intelligence - Highlighting risks and vulnerabilities*. Deliverable D2. [online] Available at: <http://vg00.met.vgwort.de/na/b5df8484314f05ddd8e0?l=http://publica.fraunhofer.de/eprints/urn:nbn:de:0011-n-340076.pdf> [Accessed 31 Aug. 2015].
- [47] van der Geest, T., Pieterse, W., & de Vries, P. (2005). Informed consent to address trust, control, and privacy concerns in user profiling. *Privacy Enhanced Personalisation, PEP*, 23-34.

References

- [48] Hoffman, D. L., Novak, T. P., & Peralta, M. (1999). Building consumer trust online. *Communications of the ACM*, 42(4), 80-85.
- [49] Grandison, T., & Sloman, M. (2000). A survey of trust in internet applications. *Communications Surveys & Tutorials*, IEEE, 3(4), 2-16.
- [50] Kobsa, A., & Teltzrow, M. (2005, January). Contextualized communication of privacy practices and personalization benefits: Impacts on users' data sharing and purchase behavior. In *Privacy Enhancing Technologies* (pp. 329-343). Springer Berlin Heidelberg.
- [51] Miettinen, M. (2007). Trust and Privacy Management in MobiLife-on the Difficulty of Giving Control to the User. Helsinki, Finland, [online]. Available at: <http://www.cs.helsinki.fi/group/nodes/Helsinki-RutgersWorkshop07/abstracts/miettinen.pdf> [Accessed 31 Aug. 2015].
- [52] Manber, U., Patel, A., & Robison, J. (2000). Yahoo!. *Communications of the ACM*, 43(8), 35.
- [53] Cranor, L. F. (2004). I didn't buy it for myself. In *Designing personalized user experiences in eCommerce* (pp. 57-73). Springer Netherlands.
- [54] EU Data Protection Directive 95/46/EC. [online] Available at: http://ec.europa.eu/justice/policies/privacy/docs/95-46-ce/dir1995-46_part1_en.pdf. [Accessed 31 Aug. 2015].
- [55] EU Directive 2002/58/EC on privacy and electronic communications. [online] Available at: <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32002L0058:en:HTML>. [Accessed 31 Aug. 2015].
- [56] U.S. Dep't. of Health, Education and Welfare, Secretary's Advisory Committee on Automated Personal Data Systems, Records, computers, and the Rights of Citizens viii (1973).
- [57] Jensen, C., & Potts, C. (2004, April). Privacy policies as decision-making tools: an evaluation of online privacy notices. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (pp. 471-478). ACM.
- [58] Jensen, C., Potts, C., & Jensen, C. (2005). Privacy practices of Internet users: self-reports versus observed behavior. *International Journal of Human-Computer Studies*, 63(1), 203-227.
- [59] Spiekermann, S., & Cranor, L. F. (2009). Engineering privacy. *Software Engineering, IEEE Transactions on*, 35(1), 67-82.

References

- [60] Nissenbaum, H. (2004). Privacy as contextual integrity. *Washington law review*, 79(1).
- [61] Barth, A., Datta, A., Mitchell, J. C., & Nissenbaum, H. (2006, May). Privacy and contextual integrity: Framework and applications. In *Security and Privacy, 2006 IEEE Symposium on* (pp. 15-pp). IEEE. [online] Available at: <http://dx.doi.org/10.1109/SP.2006.32> [Accessed 31 Aug. 2015].
- [62] Omoronyia, I., Pasquale, L., Salehie, M., Cavallaro, L., Doherty, G., & Nuseibeh, B. (2012, September). Caprice: a tool for engineering adaptive privacy. In *Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering* (pp. 354-357). ACM. [online] Available at: <http://doi.acm.org/10.1145/2351676.2351745> [Accessed 31 Aug. 2015].
- [63] Omoronyia, I., Cavallaro, L., Salehie, M., Pasquale, L., & Nuseibeh, B. (2013, May). Engineering adaptive privacy: on the role of privacy awareness requirements. In *Proceedings of the 2013 International Conference on Software Engineering* (pp. 632-641). IEEE Press.
- [64] Schilit, B. N., LaMarca, A., Borriello, G., Griswold, W. G., McDonald, D., Lazowska, E., & Iverson, V. (2003, September). Challenge: Ubiquitous location-aware computing and the place lab initiative. In *Proceedings of the 1st ACM international workshop on Wireless mobile applications and services on WLAN hotspots* (pp. 29-35). ACM. [online] Available at: <http://doi.acm.org/10.1145/941326.941331> [Accessed 31 Aug. 2015].
- [65] Mead, N. R., & Stehney, T. (2005). Security quality requirements engineering (SQUARE) methodology (Vol. 30, No. 4, pp. 1-7). ACM.
- [66] Bijwe, A., & Mead, N. R. (2010). Adapting the square process for privacy requirements engineering (CMU/SEI-2010-TN-022). Software Engineering Institute. Carnegie Mellon University. Available at: **Error! Hyperlink reference not valid.** [Accessed 30 Sept. 2015].
- [67] Kay, J., Kummerfeld, B., & Lauder, P. (2003, June). Managing private user models and shared personas. In *UM03 Workshop on User Modeling for Ubiquitous Computing* (pp. 1-11).
- [68] Lederer, S., Dey, A. K., & Mankoff, J. (2002, September). Everyday privacy in ubiquitous computing environments. In *UbiComp 2002 Workshop on Socially-informed Design of Privacy-enhancing Solutions in Ubiquitous Computing*.

References

- [69] Adams, A. (1999). The implications of users' multimedia privacy perceptions on communication and information privacy policies. In Proceedings of Telecommunications Policy Research Conference.
- [70] Langheinrich, M. (2002). A privacy awareness system for ubiquitous computing environments. In UbiComp 2002: Ubiquitous Computing (pp. 237-245). Springer Berlin Heidelberg.
- [71] Ackerman, M. S. (2004). Privacy in pervasive environments: next generation labeling protocols. *Personal and Ubiquitous Computing*, 8(6), 430-439 [online]. Available at: <http://dx.doi.org/10.1007/s00779-004-0305-8> [Accessed 31 Aug. 2015].
- [72] Adam, K., Price, B., Richards, M., & Nuseibeh, B. (2005). A privacy preference model for pervasive computing. Proceedings of the First European Conference on Mobile Government, University of Sussex, Brighton, 10-12 July.
- [73] Organisation for Economic Co-operation and Development ("OECD"), "Guidelines on the Protection of Privacy and Transborder Flows of Personal Data". [online]. Available at: <http://www.oecd.org/sti/ieconomy/oecdguidelinesonthe protectionofprivacyandtransborderflowsofpersonaldata.htm> [Accessed 31 Aug. 2015].
- [74] Organisation for Economic Co-operation and Development ("OECD"), "The OECD Privacy Framework 2013" [online]. Available at: http://oecd.org/sti/ieconomy/oecd_privacy_framework.pdf [Accessed 31 Aug. 2015].
- [75] Privacy Online: Fair Information Practices in the Electronic Marketplace: A Federal Trade Commission Report to Congress [online]. Available at: <https://www.ftc.gov/reports/privacy-online-fair-information-practices-electronic-marketplace-federal-trade-commission>. [Accessed 31 Aug. 2015].
- [76] US Privacy Act 1974 [online]. Available at: <http://www.justice.gov/opcl/privacy-act-1974> [Accessed 31 Aug. 2015].
- [77] Hong, J. I., Boriello, G., Landay, J. A., McDonald, D. W., Schilit, B. N., & Tygar, J. D. (2003, October). Privacy and security in the location-enhanced world wide web. In Proceedings of Fifth International Conference on Ubiquitous Computing: UbiComp.
- [78] Kobsa, A., & Teltzrow, M. (2006). Convincing Users to Disclose Personal Data. *Privacy-Enhanced Personalization*, 14(4), 39.

References

- [79] Wills, C. E., & Zeljkovic, M. (2011). A personalized approach to web privacy: awareness, attitudes and actions. *Information Management & Computer Security*, 19(1), 53-73.
- [80] Fu, H., Yang, Y., Shingte, N., Lindqvist, J., & Gruteser, M. (2014). A field study of run-time location access disclosures on android smartphones. *Proc. USEC*, 14.
- [81] Play.google.com, (2015). *WhatsApp*. [online] Available at: <https://play.google.com/store/apps/details?id=com.whatsapp&hl=en> [Accessed 31 Aug. 2015].
- [82] Play.google.com, (2015). *ESPN App*. [online] Available at: https://play.google.com/store/apps/details?id=com.espn.score_center&hl=en [Accessed 31 Aug. 2015].
- [83] Play.google.com, (2015). *CricInfo App*. [online] Available at: <https://play.google.com/store/apps/details?id=com.july.cricinfo&hl=en> [Accessed 31 Aug. 2015].
- [84] Koliass, C., Koliass, V., Anagnostopoulos, I., Kambourakis, G., & Kayafas, E. (2008, December). Enhancing user privacy in adaptive web sites with client-side user profiles. In *Semantic Media Adaptation and Personalization, 2008. SMAP'08. Third International Workshop on* (pp. 170-176). IEEE.
- [85] TorProject.org. (2015) About Tor Project webpage. Available at: <https://www.torproject.org/about/overview.html.en> [Accessed 30 Sept. 2015].
- [86] Klobucar, T., Senicar, V., & Blazic, B. J. (2004). Privacy and personalisation in a smart space for learning. *International Journal of Continuing Engineering Education and Life Long Learning*, 14(4-5), 388-401.
- [87] Zarsky, T. Z. (2003). Thinking outside the box: considering transparency, anonymity, and pseudonymity as overall solutions to the problems in information privacy in the internet society. *U. Miami L. Rev.*, 58, 991.
- [88] Kobsa, A., & Schreck, J. (2003). Privacy through pseudonymity in user-adaptive systems. *ACM Transactions on Internet Technology (TOIT)*, 3(2), 149-183.
- [89] Leonid Titkov and Stefan Poslad. 2003. Titkov, L., & Poslad, S. (2003, July). Privacy conscious brokering in personalised location-aware applications. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems* (pp. 1138-1139). ACM. [online] Available at: <http://doi.acm.org/10.1145/860575.860834> [Accessed 31 Aug. 2015].

References

- [90] Beresford, A. R., & Stajano, F. (2003). Location privacy in pervasive computing. *IEEE Pervasive computing*, (1), 46-55.
- [91] Lederer, S., Mankoff, J., & Dey, A. K. (2003, April). Who wants to know what when? privacy preference determinants in ubiquitous computing. In *CHI'03 extended abstracts on Human factors in computing systems* (pp. 724-725). ACM.
- [92] Adams, A. (2000, April). Multimedia information changes the whole privacy ballgame. In *Proceedings of the tenth conference on Computers, freedom and privacy: challenging the assumptions* (pp. 25-32). ACM.
- [93] Platform for Privacy Preferences (P3P) [online]. February 2006. Available at: <http://www.w3.org/P3P/> [Accessed 31 Aug. 2015].
- [94] OASIS Extensible Access Control Modelling Language (XACML). [online] Available at: <http://www.oasis-open.org/committees/xacml/> [Accessed 31 Aug. 2015].
- [95] A P3P Preference Exchange Language 1.0 (APPEL1.0) W3C Working Draft [online]. Available at: <http://www.w3.org/TR/P3P-preferences/> [Accessed 31 Aug. 2015].
- [96] Riedl, J. (2001). Personalization and privacy. *Internet Computing, IEEE*, 5(6), 29-31.
- [97] IBM Corporation, Enterprise Privacy Authorization Language (EPAL 1.1). [online] Available at: <http://www.w3.org/2003/p3p-ws/pp/ibm3.html> [Accessed 31 Aug. 2015].
- [98] Anderson, A. H. (2004, June). An introduction to the web services policy language (wspl). In *Policies for Distributed Systems and Networks, 2004. POLICY 2004. Proceedings. Fifth IEEE International Workshop on* (pp. 189-192). IEEE.
- [99] "Pretty Poor Privacy: An Assessment of P3P and Internet Privacy," Electronic Privacy Information Center Report on P3P [online] Available at: <http://www.epic.org/Reports/pretypoorprivacy.html> [Accessed 31 Aug. 2015].
- [100] Eldin, A. A., & Wagenaar, R. W. (2006, March). A Privacy Preferences Architecture for Context Aware Applications. In *AICCSA* (pp. 1110-1113).
- [101] Schulzrinne H., Tschofenig H., Morris, J., Cuellar J., Polk J. RFC 4745. Common Policy: A Document Format for Expressing Privacy Preferences. February 2007 [online] Available at: <ftp://ftp.rfc-editor.org/in-notes/rfc4745.txt> [Accessed 31 Aug. 2015].
- [102] Acquisti, A., & Grossklags, J. (2005). Privacy and rationality in individual decision making. *IEEE Security & Privacy*, (1), 26-33.

- [103] Kagal, L., Finin, T., Joshi, A., & Greenspan, S. (2006). Security and privacy challenges in open and dynamic environments. *Computer*, 39(6), 89-91.
- [104] Resource Description Framework (RDF) [online]. Available at: <http://www.w3.org/RDF/>. [Accessed 31 Aug. 2015].
- [105] Ontology Web Language (OWL) [online]. Available at: <http://www.w3.org/TR/owl-features/>. [Accessed 31 Aug. 2015].
- [106] Garcia, D. Z. G., & Toledo, M. (2008). A web service privacy framework based on a policy approach enhanced with ontologies. In *Computational Science and Engineering Workshops, 2008. CSEWORKSHOPS'08. 11th IEEE International Conference on* (pp. 209-214). IEEE.
- [107] Robinson, P., Vogt, H., & Wagealla, W. (Eds.). (2006). *Privacy, security and trust within the context of pervasive computing* (Vol. 780). Springer Science & Business Media.
- [108] Kolovski, V., Katz, Y., Hendler, J., Weitzner, D., & Berners-Lee, T. (2005, November). Towards a policy-aware web. In *Semantic Web and Policy Workshop at the 4th International Semantic Web Conference* (p. 31).
- [109] Chatfield, C., Carmichael, D., Hexel, R., Kay, J., & Kummerfeld, B. (2005, November). Personalisation in intelligent environments: managing the information flow. In *Proceedings of the 17th Australia conference on Computer-Human Interaction: Citizens Online: Considerations for Today and the Future* (pp. 1-10). Computer-Human Interaction Special Interest Group (CHISIG) of Australia.
- [110] Chellappa, R. K., & Sin, R. G. (2005). Personalization versus privacy: An empirical examination of the online consumer's dilemma. *Information Technology and Management*, 6(2-3), 181-202 [online]. Available at: <http://dx.doi.org/10.1007/s10799-005-5879-y> [Accessed 31 Aug. 2015].
- [111] Giang, P. D., Hung, L. X., Lee, S., Lee, Y. K., & Lee, H. (2007, April). A flexible trust-based access control mechanism for security and privacy enhancement in ubiquitous systems. In *Multimedia and Ubiquitous Engineering, 2007. MUE'07. International Conference on* (pp. 698-703). IEEE.
- [112] Kowalski, S., & Edwards, N. (2004). A security and trust framework for a Wireless World: A Cross Issue Approach. In *Wireless World Research Forum no* (Vol. 12).
- [113] Almenarez, F., Marin, A., Díaz, D., & Sanchez, J. (2006, March). Developing a model for trust management in pervasive devices. In *Pervasive Computing and*

- Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on (pp. 5-pp). IEEE.
- [114] Carminati, B., & Ferrari, E. (2005). Trusted Privacy Manager: A System for Privacy Enforcement. In Data Engineering Workshops, 2005. 21st International Conference on (pp. 1195-1195). IEEE.
- [115] Nyre, Å. A., Bernsmed, K., Bø, S., & Pedersen, S. (2011). A server-side approach to privacy policy matching. In Availability, Reliability and Security (ARES), 2011 Sixth International Conference on (pp. 609-614). IEEE.
- [116] Papadopoulou, E., McBurney, S., Taylor, N., Williams, H., & Bello, G. L. (2008). Adapting stereotypes to handle dynamic user profiles in a pervasive system. In Proceedings of the 4th IASTED International Conference on Advances in Computer Science and Technology, ACST 2008. (pp. 7-12).
- [117] PrivacyBird.org Tool Website. [online]. Available at: <http://www.privacybird.org/> [Accessed 31 Aug. 2015].
- [118] Kolter, J., & Pernul, G. (2009). Generating user-understandable privacy preferences. In Availability, Reliability and Security, 2009. ARES'09. International Conference on (pp. 299-306). IEEE.
- [119] Bergmann M. (2005). PRIME internal Privacy Preferences survey about Privacy Concerns and Conditions. Technische Universitt Dresden, Technische Berichte, TUD-FI07-04-Mai-2005, May 2005.
- [120] Schaub, F., Könings, B., Weber, M., & Kargl, F. (2012, March). Towards context adaptive privacy decisions in ubiquitous computing. In Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on (pp. 407-410). IEEE.
- [121] Tondel, I. A., Nyre, Å. A., & Bernsmed, K. (2011, August). Learning privacy preferences. In Availability, Reliability and Security (ARES), 2011 Sixth International Conference on (pp. 621-626). IEEE [online]. Available at: <http://dx.doi.org/10.1109/ARES.2011.96> [Accessed 31 Aug. 2015].
- [122] Cornwell, J., Fette, I., Hsieh, G., Prabaker, M., Rao, J., Tang, K., Vaniea, K., Bauer, L., Cranor, L., Hong, J., McLaren, B., Reiter, M., & Sadeh, N. (2007). User-controllable security and privacy for pervasive computing. In Eighth IEEE Workshop on Mobile Computing Systems and Applications (HotMobile).
- [123] McBurney, S., Papadopoulou, E., Taylor, N., & Williams, H. (2008). Adapting pervasive environments through machine learning and dynamic personalization. In

References

- Proceedings of the 2008 International Symposium on Parallel and Distributed Processing with Applications, ISPA 2008. (pp. 395-402). 10.1109/ISPA.2008.63.
- [124] Quinlan, J.R., (1993). C45: Programs for Machine Learning. Morgan Kaufman.
- [125] Papadopoulou, E., McBurney, S., Taylor, N., & Williams, H. (2008). Linking privacy and user preferences in the identity management for a pervasive system. In Proceedings - 2008 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2008. (pp. 192-195). 10.1109/WIIAT.2008.331.
- [126] Walker, D. D. (2007). OR BEST OFFER: A Privacy Policy Negotiation Protocol. MSc Thesis, Brigham Young University, August 2007 [online]. Available at: <http://scholarsarchive.byu.edu/etd/1016/> [Accessed 31 Aug. 2015].
- [127] Preibusch, S. (2006, October). Privacy negotiations with p3p. In W3C Workshop on Languages for Privacy Policy Negotiation and Semantics-Driven Enforcement (Vol. 30) [online]. Available at: <http://www.w3.org/2006/07/privacy-ws/papers/24-preibusch-negotiation-p3p/> [Accessed 31 Aug. 2015].
- [128] Press Release, Fed. Trade Comm'n, FTC Staff Issues Privacy Report, Offers Framework for Consumers, Businesses, and PolicyMakers (Dec. 1, 2010).
- [129] Preibusch, S. (2006, April). Personalized services with negotiable privacy policies. In PEP06, CHI 2006 workshop on privacy-enhanced personalization, Montreal, Canada (pp. 29-38).
- [130] Tamaru, S., Nakazawa, J., Takashio, K., & Tokuda, H. (2003, October). PPNP: A privacy profile negotiation protocol for services in public spaces. In Proc. Fifth International Conference on Ubiquitous Computing (Ubi-Comp2003), Seattle, WA.
- [131] Lee, H. H., & Stamp, M. (2006, November). P3P privacy enhancing agent. In Proceedings of the 3rd ACM workshop on Secure web services (pp. 109-110). ACM <http://doi.acm.org/10.1145/1180367.1180389>.
- [132] Hsu-Hui Lee. December 2006. P3P Privacy Enhancing Agent. Report for the purposes of obtaining the degree of Msc in Computer Science. San Jose State University [online]. Available at: http://www.cs.sjsu.edu/faculty/stamp/students/Hsu-Hui_Lee_298Report.pdf [Accessed 31 Aug. 2015].
- [133] Java Agent Development Environment Homepage [online]. Available at: <http://jade.tilab.com/> [Accessed 31 Aug. 2015].
- [134] Walker, D. D., Mercer, E. G., & Seamons, K. E. (2008, June). Or best offer: A privacy policy negotiation protocol. In Policies for Distributed Systems and Networks, 2008. POLICY 2008. IEEE Workshop on (pp. 173-180). IEEE.

References

- [135] Zhang, N. J., & Todd, C. (2006, January). A privacy agent in context-aware ubiquitous computing environments. In *Communications and Multimedia Security* (pp. 196-205). Springer Berlin Heidelberg.
- [136] Hull, R., Kumar, B., Lieuwen, D., Patel-Schneider, P. F., Sahuguet, A., Varadarajan, S., & Vyas, A. (2004). Enabling context-aware and privacy-conscious user data sharing. In *Mobile Data Management, 2004. Proceedings. 2004 IEEE International Conference on* (pp. 187-198). IEEE.
- [137] El-Khatib K. (2003). A Privacy Negotiation Protocol for Web Services, Workshop on Collaboration Agents: Autonomous Agents for Collaborative Environments Halifax, Nova Scotia, Canada.
- [138] Cheng, V. S., Hung, P. C., & Chiu, D. K. (2007, January). Enabling web services policy negotiation with privacy preserved using XACML. In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on* (pp. 33-33). IEEE.
- [139] OpenID authentication protocol homepage [online]. Available at: <http://openid.net/> [Accessed 31 Aug. 2015].
- [140] WebID Identity open standard specification [online]. Available at: <http://www.w3.org/2005/Incubator/webid/spec/> [Accessed 31 Aug. 2015].
- [141] Microsoft CardSpace WebSite [online]. Available at: <http://msdn2.microsoft.com/en-US/library/aa480189.aspx> [Accessed 31 Aug. 2015].
- [142] Liberty Alliance Project Website [online]. Available at: <http://www.projectliberty.org/> [Accessed 31 Aug. 2015].
- [143] Jøsang, A., Zomai, M. A., & Suriadi, S. (2007, January). Usability and privacy in identity management architectures. In *Proceedings of the fifth Australasian symposium on ACSW frontiers-Volume 68* (pp. 143-152). Australian Computer Society, Inc.
- [144] Hardt, D. (2005). "Identity 2.0". OSCON (2005) Keynote address [online]. Available at: <https://www.youtube.com/watch?v=RrpajcAgR1E> [Accessed 31 Aug. 2015].
- [145] Cameron, Kim. (2005) The Laws of Identity. Microsoft Whitepaper, [online]. Available at: <http://msdn2.microsoft.com/en-us/library/ms996456.aspx> [Accessed 31 Aug. 2015].
- [146] Microsoft. Microsoft's Vision for an Identity Metasystem. (2005) Microsoft Whitepaper [online]. Available at: <http://msdn2.microsoft.com/en-us/library/ms996422.aspx> [Accessed 31 Aug. 2015].

References

- [147] Ives, B., Walsh, K. R., & Schneider, H. (2004). The domino effect of password reuse. *Communications of the ACM*, 47(4), 75-78.
- [148] Credentica website [online]. Available at: <http://www.credentica.com> [Accessed 31 Aug. 2015].
- [149] Spantzel, A. B., Squicciarini, A. C., & Bertino, E. (2007). Trust negotiation in identity management. *Security & Privacy, IEEE*, 5(2), 55-63.
- [150] Shibboleth Federated Identity Management System. Available at: <https://shibboleth.net> [Accessed 30 Sept. 2015].
- [151] EU FP6 DAIDALOS I & II project HomePage [online]. Available at: www.ist-DAIDALOS.org [Accessed 31 Aug. 2015].
- [152] FP7 EU ICT PERSIST Project Homepage [online]. Available at: www.ict-persist.eu [Accessed 31 Aug. 2015].
- [153] Youtube.com. DAIDALOS 'Nidaros' scenario videoclip. Available at: <https://youtube.com/watch?v=bvMnjkdD0m4> [Accessed 30 Sept. 2015].
- [154] Youtube.com. PERSIST demonstration scenario videoclip. Available at: <https://youtube.com/watch?v=Z7ZzqncQjul> [Accessed 30 Sept. 2015].
- [155] Papadopoulou, E., Gallacher, S., Taylor, N. K., & Williams, H. (2012). Personalizing the User's Physical Environment in a Pervasive System. In W. Assawinchaichote, & M. H. Hamza (Eds.), *Proceedings 2nd IASTED Asian Conference on Modelling, Identification and Control (AsiaMIC 2012)*. ACTA Press. 10.2316/P.2012.770-013.
- [156] Bental, D.S., Papadopoulou, E., Taylor, N.K., Williams, M.H., Blackmun, F., Ibrahim, I., Lim, M., Mimitsoudis, I., Whyte, S. & Jennings, E. (2015). Smartening up the Student Learning Experience with Ubiquitous Media, *ACM Transactions on Multimedia Computing, Communications and Applications*, to appear.
- [157] Matos, A., Girão, J., Sargento, S., & Aguiar, R. (2007, November). Preserving privacy in mobile environments with virtual network stacks. In *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE* (pp. 1971-1976). IEEE.
- [158] Clarke, J., Butler, S., Dempsey, S., Crotty, M., Brazil, J., Hauser, C., Neubauer, M., & Blazic, A. J. (2004). Challenges of identity, authentication, and discovery management in a ubiquitous environment: The DAIDALOS perspective. *Interworking 2004, 7th International Symposium on Communications Interworking*, National Research Council Campus, Ottawa.
- [159] JXTA specification. [online]. Available at: <https://jxta-spec.dev.java.net> [Accessed 31 Aug. 2015].

- [160] FP7 EU ICT SOCIETIES Project Homepage [online]. Available at: www.ict-societies.eu[Accessed 31 Aug. 2015].
- [161] Play.google.com, (2015). *Google Play Store*. [online] Available at: <https://play.google.com/store> [Accessed 31 Aug. 2015].
- [162] Itunes.apple.com, (2015). *App Store Downloads on iTunes*. [online] Available at: <https://itunes.apple.com/us/genre/ios/id36> [Accessed 31 Aug. 2015].
- [163] Gallacher, S., Papadopoulou, E., Taylor, N. K., & Williams, M. H. (2013). Learning user preferences for adaptive pervasive environments: An incremental and temporal approach. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 8(1), 5 [online]. Available at: <http://doi.acm.org/10.1145/2451248.2451253> [Accessed 31 Aug. 2015].
- [164] Kalatzis, N., Roussaki, I., Liampotis, N., Kosmides, P., Papaioannou, I. and Anagnostou, M. (2014). Context and Community Awareness in Support of User Intent Prediction. *Springer New York*, [online] pp.359-378. Available at: http://dx.doi.org/10.1007/978-1-4939-1887-4_23.
- [165] Doolin, K., Taylor, N., Crotty, M., Roddy, M., Jennings, E., Roussaki, I., & McKitterick, D. (2014). Enhancing mobile social networks with ambient intelligence. In *Mobile Social Networking* (pp. 139-163). Springer New York.
- [166] Liampotis, N., Roussaki, I., Papadopoulou, E., Abu-Shaaban, Y., Williams, H., Taylor, N.K., McBurney, S., & Dolinar, K. (2009). A privacy framework for personal self-improving smart spaces. In *Proceedings - 12th IEEE International Conference on Computational Science and Engineering, CSE 2009 - 2009 IEEE International Conference on Privacy, Security, Risk, and Trust, PASSAT 2009*. (Vol. 3, pp. 444-449). 10.1109/CSE.2009.148
- [167] Liampotis, N., Roussaki, I., Kalatzis, N., Papadopoulou, E., Gonçalves, J., Papaioannou, I. and Sykas, E. (2014). Context-Sensitive Trust Evaluation in Cooperating Smart Spaces. Springer New York, [online] pp.187-201. Available at: http://dx.doi.org/10.1007/978-1-4939-1887-4_13.
- [168] Liampotis, N., Papadopoulou, E., Kalatzis, N., Roussaki, I. G., Kosmides, P., Sykas, E. D., Bental, D., & Taylor, N. K. (2016). Tailoring Privacy-Aware Trustworthy Cooperating Smart Spaces for University Environments. In A. Panagopoulos (Ed.), *Handbook of Research on Next Generation Mobile Communication Systems* (pp. 410-439). Hershey, PA: Information Science Reference. doi:10.4018/978-1-4666-8732-5.ch016.

References

- [169] Orwell, G. (1949). *Nineteen Eighty-four: Animal Farm*. Chancellor Press.
- [170] Google.com, (2014). *Privacy Policy – Privacy & Terms – Google*. [online] Available at: <http://www.google.com/policies/privacy/archive/20141219/> [Accessed 31 Aug. 2015].
- [171] Bbc.com. (2015). *BBC - Privacy Policy - Privacy and Cookies*. [online] Available at: <http://www.bbc.com/privacy/information/policy/> [Accessed 31 Aug. 2015].
- [172] www.hw.ac.uk. (2014). *Heriot-Watt University Data Protection Policy*. [online] Available at: from <http://www.hw.ac.uk/documents/heriot-watt-university-data-protection-policy.pdf> [Accessed 31 Aug. 2015].
- [173] Wikipedia, (2015). *Reading (process)*. [online] Available at: [https://en.wikipedia.org/wiki/Reading_\(process\)#Reading_rate](https://en.wikipedia.org/wiki/Reading_(process)#Reading_rate) [Accessed 31 Aug. 2015].
- [174] Commission Nationale de l'Informatique et des Libertés [online]. Available at: <http://www.cnil.fr/> [Accessed 31 Aug. 2015].
- [175] Cnil.fr, (2012). Google's new privacy policy: incomplete information and uncontrolled combination of data across services - CNIL - Commission nationale de l'informatique et des libertés. [online] Available at: <http://www.cnil.fr/english/news-and-events/news/article/googles-new-privacy-policy-incomplete-information-and-uncontrolled-combination-of-data-across-ser/> [Accessed 31 Aug. 2015].