Adaptive Privacy Management for Distributed Applications

Maomao Wu

M.Sc. (Liverpool, UK, 2000)

B.E. (Shanghai, P.R.China. 1999)

Computing Department

Lancaster University

United Kingdom

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

JUNE 2007

Abstract

Adaptive Privacy Management for Distributed Applications

Maomao Wu
Computing Department
Lancaster University

Submitted for the degree of Doctor of Philosophy.
June 2007.

In networked computing environments, it becomes increasingly difficult for normal people to manage privacy, i.e., "to determine for themselves when, how, and to what extent information about them is communicated with others". The thesis argues that achieving better privacy is not about hiding as much personal information as possible but enabling personal information disclosure at a level of openness that is as close as to a user's desired level to assist him/her in accomplishing useful tasks. Following Palen and Dourish's observation that privacy management is a dialectic and dynamic boundary regulation process [Palen03], the thesis argues that no set of pre-specified static privacy policies can meet users' changing requirements for privacy in networked computing environments, and therefore a new approach (i.e., adaptive privacy management) is proposed as the process that a user and/or a system to continuously adjust the system behaviour of disclosing personal information according to the user's changing desire for openness.

In this thesis, we propose a set of requirements for adaptive privacy management and

the design and implementation of a middleware that meets these requirements for the target domain of applications that enable intentional sharing of personal information in networked computing environments. The middleware facilitates the creation of adaptive privacy aware applications that allows users or the system on behalf of the user to adjust the balance between openness and closedness; leading to an evolution of the users' privacy preferences as a result of on-going interactions.

A prototype adaptive privacy management system was implemented based on this middleware; demonstrating the feasibility of adaptive privacy management for the target domain. Both the principles of adaptive privacy management and the prototype implementation were evaluated based on the results of a detailed user study using a GSM location sharing application constructed using the prototype platform. The study reveals the our core requirements are important for end users, and that our supporting design did provide adequate support for the characteristics we propose.

Declaration

This thesis has been written by myself, and the work reported herein is my own. Many of the ideas in this thesis were the product of discussions with my supervisor Dr. Adrian Friday. The work reported in this thesis has not been previously submitted for a degree in this, or any other form.

Maomao Wu

Acknowledgements

Firstly, I would like express my most sincere thanks to my supervisor Dr. Adrian Friday. He not only encouraged and inspired my Ph.D. research, but also provided dedicated support throughout the work, from insightful comments and constructive criticism to technical and linguistic guidance. This thesis would never have been completed without his patient supervision and most importantly tremendous support that he dedicated to the writing of the thesis. Moreover, I owe him thanks for offering me the opportunities to work with him on various research projects, and I feel extremely lucky to have worked under his guidance.

I would like to thank Prof. Nigel Davies for his insightful recommendation on evaluating the work in the thesis and for his supervision during Adrian's absence. I would also like to thanks Dr. Joe Finney for reading the whole thesis and provided useful comments. Moreover, I would like to express my thanks to Prof. Alan Dix and Dr. Corina Sas for providing expert advice on conducting the user study. Many thanks to all the participants of the user study, as this work would not have been finished without your generous help. Thanks to all the colleagues and friends in the departments. Special thanks to Dr. Christos Efstratiou for creating the elegant Lancasterian thesis template and providing assistance in preparing the LaTex version of the thesis. Additional thanks to other guys in D23: Oliver Storz, Mark Lowton, Andre Hesse, and Rob Hooper.

Finally, I would like to express my formal acknowledgements to my family. I would like to thank my parents for their support throughout my education, and especially my father for encouraging me to pursue my Ph.D. overseas. Above all, I would like to offer my deepest thanks to my dear wife Yunyan Li, who had to make many sacrifices and compromises during my Ph.D. study. Without your consistent love, encouragement and support, this thesis would not have been completed.

Contents

List of Tables

List of Figures

CHAPTER I

Introduction

Contents

1.1 The Right to Privacy

Novel technologies and their adoption have significant impact on ethical, economic, legal and social issues. Failing to address these issues in the development stage may foster misuse of technology and incur an adverse response to it. When Samuel Warren and Louis Brandeis first introduced the notion of *privacy* in their influential article *The Right to Privacy* [Warren90] in 1890, they were fiercely reacting to the unethical use of portable photography and modern printing technologies to publicly disseminate information relating to individuals' private lives. Since then, privacy has become a multi-disciplinary topic that attracts attentions from psychological, economic, legal, social and technological researchers.

With the advance in computer networks and the popularity of Internet applications, the flow of personal information can easily get out of control: private emails can be intercepted by unscrupulous third-parties; credit card details and mailing addresses can be transmitted by simply filling in a web form and the click of a button; people's online activities are increasingly logged, archived, and searched; the details of online transactions might be maintained and exploited for marketing purposes by companies that you are not even aware of. In contrast with the original definition of privacy by Brandeis, as *"the right to be left alone"*, we believe that privacy is more about controlling the flow of the personal information in the modern information age. As we define based on the writings of Alan Westin in *Privacy and Freedom* [Westin67]:

> **Definition 1**: *Privacy is the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others.*

A brief and memorable definition was also provided by Alan Westin as *"the right to select what personal information about me is known to what people"*. The focus of the thesis is to assist people in effectively managing their private information in networked computing environments.

1.2 Motivation

The last decade has witnessed important technological changes that have had a significant impact on people's personal lives. Wide deployment of the wireless communication infrastructure, most notably in the form of the Global System for Mobile Communications (GSM) and Wireless Local Area Networks (WLAN), has dramatically increased the availability of network connectivity allowing people to be seamlessly connected to each other from anywhere anytime. Coincidentally, decreased cost of terminal devices has enabled an increasing number of people to possess and utilise a variety of electronic appliances in their everyday life. The mobile phone handset has become ubiquitous in many parts of the world, and people have already regarded it as commodity or necessity of their daily life [Davies02]. The high availability of connected devices such as these fosters the development, deployment, and adoption of new applications that empower people to seamlessly communicate with each other from a variety of Information and Communication Technology (ICT) devices. For example, Instant Messenger (IM) is readily found running on PCs, laptops, PDAs and mobile phones, facilitating communication between people via text messaging, voice conversation, and video conferencing; but also increasingly distributing presence and contextual information.

However, these technological changes have negative impact on the privacy of people's personal information. *To be left alone* becomes much more difficult to achieve because of the "always-available" network connections and "always-on" terminal devices. In the near future, we believe IMs will transmit not only people's presence information but also other contextual information such as location or activity, to facilitate people to find and communicate with one another [Hong04a]. With the increase in personal information that can be disclosed, simple "on or off" control will not be sufficient for enabling users to *"determine for themselves when, how, and to what extent information about them is communicated to others"*. For example, people might disclose their location information to their colleagues when they are at work, but they might also disclose location information to family members after working hours. Moreover, with large amount of personal information being recorded, distributed, accumulated, and stored, it becomes increasingly possible that the information will be correlated and analysed over time to reveal other types of personal information, such as working patterns [Beresford03].

These issues have been exacerbated as miniaturized sensors have been integrated

into commodity hardware. GPS devices in vehicles can provide convenient location and route information for drivers while they are on the road, but this apparently private information has already been exploited *by companies against their customers*: car-rental companies have charged their customers for crossing state boundaries in US [Lemos01] and even for speeding [CBS News04]. Radio Frequency Identification (RFID) technology was envisioned to replace barcodes, and its huge code space makes it possible to tag every single product on earth. For example, the 96 bit code and partition scheme in Auto-ID standard not only offers space for 256 million manufacturers and 16 million products per manufacturer, but also leaves 64 billion serial numbers for each individual product model [Stajano05]. Unlike barcodes that can only be read by aligning them with the reader, RFID tags use radio frequency and can be read within range of the RFID reader without any explicit action. With RFID, machines effectively possess an X-ray vision [Stajano05], which enables them to silently scan us at any point to find out detailed information of the items on our body or among or posessions. Machines can not only know what brand and size the items are, but when and where we bought them.

Coupled with these rapid advances in wireless communication and sensor-based computing, recent evolution of microprocessors has enabled computing capabilities to be increasingly embedded into 'smart artefacts'; daily objects that can communicate seamlessly with one another and are increasingly aware of their surroundings or modes of use. In 1991, Mark Weiser envisioned this technological trend and described it in his seminal Scientific American article [Weiser91]; coining the term Ubiquitous Computing (or Ubicomp). For the last decade, UbiComp has become one of the hottest research topics in computer science, and a number of research prototypes [Microsoft Corp.01, HP Corp.01, MIT04, AT&T Labs01] have emerged to explore the potential for building so-called "intelligent" or "smart environments" that are designed to utilise sensing to intelligently serve the needs of the occupants in an unobtrusive manner. In order to provide services that conform to the user's needs and desires without explicit interaction, UbiComp systems will become not only knowledgeable about users' locations or movements, but also their longer-term behaviours and habits. Keeping and exploring such detailed user knowledge is crucial for building UbiComp systems, yet paradoxically such systems clearly have a great potential for invading personal information privacy [Satyanarayanan03]. As Weiser himself acknowledged, privacy will become one of the major challenges in achieving the UbiComp vision:

"If the computational system is invisible as well as extensive, it becomes

> *hard to know what is controlling what, what is connected to what, where information is flowing, how it is being used, what is broken (vs what is working correctly, but not helpful), and what the consequences of any given action (including simply walking into a room).”*

The emergence of early Ubiquitous Computing prototypes using today’s networked environments forms the context for our work. In this thesis, we concern ourselves with the privacy of personal information in networked computing environments, from traditional local area networks, intranets, and the Internet, to emerging mobile computing and ubiquitous computing systems. We describe the scope and aims of our work as follows: in section 1.3 we provide our definition of personal and private information. Section 1.4 examines the impact of modern information technology on personal information privacy and identifies key challenges that make privacy difficult to achieve over the whole lifetime of the information. Section 1.5 investigates the range of privacy vulnerabilities in exiting networked environments. We then define the scope of the thesis in section 1.6. Section 1.7 enumerates the research aims and objectives of the thesis. We conclude with an outline of the overall structure of the thesis in section 1.8.

1.3 What is Private Information?

Our framework protects unwanted sharing of private information. We define private information in terms of personal information, thus:

> **Definition 2**: *Personal Information (or personal data) is any information (or data) that is pertinent to an identified or identifiable person.*

According to this definition, personal information refers to both primary and secondary information [Jones03]. For example, the main content of a personal document in digital format is primary information, and the timestamps for its creation and last modification are secondary information (metadata). Personal information can be represented in different formats in computing systems, e.g., a file within an operating system, a record in a database management system, a piece of data maintained in an application’s working memory, online presence exchanged by an instant messenger, sensory data with timestamps gathered from environmental sensors, etc.

Privacy is an extrinsic, not an intrinsic, property of a piece of information about an individual [Jones03]. Whether a piece of personal information is considered private is not an internal property of the information itself, but rather an external view of whether people regard it as private. Based on the above observation, we define private information as:

> **Definition 3**: *Private Information is any personal information that the user deems private.*

The subtle perception of the human parties involved necessarily means that privacy is not a purely technical problem but rather a complex socio-technical system [Anderson04].

1.4 Technological Impact on Information Privacy

We first taxonomise the threats to information privacy introduced by increased uptake of information technology. We categorise the lifecycle of such information into four stages: collection, storage, dissemination, and use. We examine the impact of technology on personal information privacy in each of these stages.

To facilitate our discussion, we use the following terms to refer to the common entities involved in the process of personal information disclosure and exploitation:

- *Data Subject* is an individual to whom personal data relates;
- *Data Collector* is an individual, a system, or an institution that collects information about the data subject; and
- *Data User* is an individual, a system, or an institution that uses information collected by the data collector. The data collector and data user can sometimes be the same entity.

1.4.1 Information Collection

Rapid deployment and adoption of computerised systems in our society have significantly increased the *quantity* of personal data that is collected. Many of us might have

already experienced the dramatic changes in many parts of our everyday lives and noticed that an increasing number of our daily activities have been recorded by computerised systems, e.g., paying bills using credit card, borrowing books from library, visiting GPs or dentists, booking flights or hotels online, etc. More recently, multimedia technologies have changed the magnitude of the personal data that can be collected, because people's activities in the physical world can be captured in various formats of digital images, audios, and videos, with higher *quality*. For instance, high-definition colour surveillance cameras, which used to be low-definition black and white systems, can focus to resolve minute details and record larger number of high resolution picture frames per second. Moreover, multimedia data contains much richer information than textual data [Adams01b], e.g., audio can reveal a subject's tone of voice, accent, or dialect, and videos can reveal a subject's appearance, mannerism, or body language.

Computers collect not only primary personal data, but also secondary or associated data [Jones03]. For example, all digital files have timestamps for creation and last modification; electronically-edited documents (e.g., a Microsoft WORD document) may contain a record of changes made during the process of preparing the final version; and web servers often log accesses to visited pages together with the IP address of the requesting client. People are often not aware of the existence of this kind of secondary data, and even if they were, they would not necessarily be able to interpret them [Jones03]. Such data can be very informative for others, and we should not overlook the potential that they may be exploited to reveal sensitive personal information. The practice of gathering metadata will continue with the advances of context-aware computing and Ubicomp, where systems increasingly collect contextual information (e.g., time, location, or activity) in addition to supporting primary interactions [Satyanarayanan03].

The *persistence* and *pervasiveness* of embedded sensors and microprocessors greatly increases the chance of personal information being captured along the *temporal* and *spatial* scale. In addition, advanced wireless communication and networking technologies greatly enhanced the capabilities of the standalone sensors and computing devices by enabling them to transmit the captured information to anywhere in the world in almost real time. At the same time, the style of information collection has become increasingly unobtrusive and invisible in order to make the computing technology disappear into the background. A direct result of the invisibility is that individuals lose awareness of their personal information being collected and do not understand the consequences of their behaviours [Greenfield02]. A recent user study of a 'smart environment' for eldercare

concluded that:

> *"reliable, inconspicuous sensing of personal information is problematic because users do not always understand the extent or methods of data collection and thus cannot adequately evaluate privacy issues"* [Beckwith03].

People must make rational choices about their actions for managing their privacy, but they cannot make such choices without knowing or understanding what the system does to what piece of their information [Smith04].

1.4.2 Information Storage

Information storage is about preserving collected information on digital storage devices, e.g., RAMs, hard disks, DVDs, etc. The advances in hardware technology not only increase the amount of data that can be maintained, but also the ease with which it can be stored and retrieved. Unlike the human memory, digital storage media *never forgets*, and therefore anything that is recorded immediately achieves *potential immortality* [Grudin01]. Preserving transient events or states in digital formats for such a prolonged period significantly increases the potential of reuse and exploitation. In public places, people's movements and activities can be recorded by surveillance cameras and potentially accessed in the future for purposes such as crime detection. In intelligent environments, occupants' daily conversations and activities can be recorded and exploited to derive their long-term behaviours and habits [Lester05].

Databases management systems (DBMS) facilitate maintaining and organising large amounts of data, and accelerate retrieving useful information from data sources. The World Wide Web (WWW) and search engines offer ordinary people a friendly user interface to access data sources conveniently, efficiently, and ubiquitously. Online activities have surged with the popularity of Internet applications, and the combination of the above technologies facilitates logging, archiving, and searching people's online activities. *The WayBack Machine* [Internet Archive07] allows users to search the archives of the WWW back to 1996. Google bought Dejanews and created *Google Groups* [Google Inc.07b] offering archival search of the Usenet newsgroups dating back to 1981.

Efficient search technologies not only allow the searcher to spend less time and

expense in executing the search itself, but also incur less burden and intrusion on the person being searched. The latter reduces the legal justification for interfering with the searches, since the legal grounds for restricting searches have been based on the burdens imposed on the person being searched [Lessig98]. In late 2005, the US government subpoenaed records from all major search engines, in order to protect children from harmful materials from the Internet [Rasch]. Almost all of the requested companies, including AOL, Yahoo and Microsoft, complied with the subpoenas except Google [BBC06].

Unlike traditional paper documents, the information on the digital storage media is normally only accessible indirectly through a combination of hardware and software. This indirection makes it difficult to completely dispose of information, as stored information leaves traces (metadata) throughout the system. For example, when a user deletes a file, for efficiency reasons operating systems typically only mark the disk space free and leave the file content untouched on the disk. Deleted files can be recovered by using special tools that use file system APIs directly. Unsecure virtual memory management systems do not encrypt the page file on the hard disk while shutdown, which makes it possible to inspect fragments of memory snapshots if one has physical access to the disk [Stajano02]. Recovering this type of information is essential to the field of 'Computer Forensics', but these same techniques also clearly present a threat to the privacy of the individuals whose information is recovered. The complexity of these software architectures can result in a lack of appreciation over the consequences of one's information disposal actions, often involuntarily leaving information exposed.

1.4.3 Information Dissemination

Information dissemination involves the process of information duplication and distribution. The process of duplicating information was first automated by the invention of modern printing press. The reduced cost and increased speed of duplicating information significantly changed the way information was distributed in Europe, and it became impractical to ban the information dissemination, even for the state and the church [Anderson03b]. Information technology transcends traditional political and legal boundaries and further speeds up the process of information duplication; broadening the types of information that can be duplicated to include those traditionally associated with the broadcast and entertainment industries (i.e. continuous media such as audio and video) — something that has caused enormous controversy in the recording indus-

try [NY Lawyer07]. While duplication is only possible to a limited degree of accuracy with traditional analogue forms of information, digital information technology has made perfect duplication (with 100% accuracy) not only possible but also the norm.

With the increased practices of information dissemination, people are losing track of their previous activities regarding the duplication and distribution of information, e.g., how many copies of the file they've made, when they transferred the copy of the file, to whom, and by what means. The pure existence of duplicated information itself has a significant impact on privacy, because the data subject can not easily detect when information is duplicated and when this is accessed. Moreover, the data subject cannot usually prevent the data receiver from disseminating his personal information to a third-party. From the experience of online P2P file-sharing systems, researchers found that it is a non-trivial technical problem to prevent people from disseminating information that they already have access to [Goldberg02].

1.4.4 Information Use

Integrated sensors and embedded computing devices facilitate the collection of personal information, and advanced communication and networking technologies amplify their capabilities by enabling distribution of collected information to powerful remote servers that can automatically process the information. In 2003, London's Congestion Charging (LCC) system [Anonymous03] deployed 700 surveillance cameras around 203 entrances and exits to the 21 square kilometre central zone of London, and connected those surveillance cameras to an number plate recognition system that automatically identifies a vehicle's number plate from the video streams. The recognition system can increasingly connect to a variety of data sources and link pieces of information together to derive more sensitive personal information. For instance, LCC now connects to the Driver and Vehicle Licensing Agency (DVLA) database, allowing identification of the vehicle's registered owner that has not paid, and issues a fine letter to the mailing address of the owner.

Data Mining technologies extract *implicit, previously unknown, and potentially useful information from data* by employing computational techniques from statistics, machine learning and pattern recognition [Open Sources07a]. Pieces of information that seem insignificant alone can suddenly become very sensitive if many of them are aggregated together [Stajano02]. For example, publishing a few digital pictures taken while

travelling has significantly different privacy implication from publishing the whole collection with timestamps, which is equivalent to disclosing the personal travel itinerary. For the same reason, allowing a friend to view your instant messenger status is a widely accepted practice, but it has dramatically different privacy implication if your friend records this status information over time to find out your long-term usage behaviour. Data Mining not only aggregates a large number of data instances for the same type, but also combines data from different sources that reveal more information. The same thread may arise in intelligent environments that record a combination of occupants' locations, activities and time, to derive long-term behaviour. Personalisation, the process of customising applications according to user's preferences, is not new in computer science, and UbiComp has made it an explicit goal to exploit as much user information as possible in order to anticipate the users' needs and desires without explicit interaction. The existence of the detailed knowledge about a user has the potential risk of being stolen and misused, and the consequences would be much more serious than the *identity theft* we might have already experienced.

Privacy enforcement is often achieved by attaching metadata to information describing how it should be processed. The fluid nature of digital media makes it easy to remove such a "privacy tag" from the information, increasing the potential that it is used for completely different purposes and under different conditions from those originally specified. The LCC system was originally designed to immediately discard the information about vehicle registration number, location, and timestamp if the vehicle has already shown as *paid* in the database. This policy has since been revised so that these sightings are kept, so that the police and other authorities may be granted access to it in the future [Stajano03]. This is not a new problem — early HCI researchers [Mackay91a] noticed numerous potential ethical problems in using videos for different audiences and purposes other than those originally intended. The fact that records of people's activities may be manipulated and used out of their original context in the future clearly poses challenging questions for developers of networked environments, and especially for UbiComp [Bellotti93].

The root of information misuse stems from the neutrality of the technologies, including information technology [Jones03], technology does not indiscriminate how people choose to use them. Therefore, many technologies invented with good intentions have been abused for illicit or malicious purposes. Email was originally conceived to facilitate personal communications, but malicious individuals or groups have exploited it as

a medium to disseminate *unsolicited commercial email (spam)* messages or *worm* programs and viruses [Pfleeger02]. Scanning and filtering technologies used in anti-virus software employed by Google's GMail [BBC04], scans the contents of users' email messages not only to filter spam and detect viruses, but also to provide users with personalised advertisements — a substantially different and more controversial purpose. Information technology serves the purposes to which it is put; information privacy is thus not a purely technical problem but rather a complex socio-technical one that interplays between technology, people, and society as a whole.

1.5 Understanding Threats to Information Privacy

Having explored technological impact on privacy generally, in this section we refine the focus of the thesis by investigating the range of privacy vulnerabilities specifically in existing networked computing systems. Based on Bellotti's dichotomy of privacy problems [Bellotti97], we classify privacy attacks into two categories: malicious or covert attacks on information privacy, and accidental or negligent releases of private information.

Malicious or covert attacks on information security can be attacks on information privacy. One type of common attacks is the covert observation of potentially private information unknown or at best at the periphery of a user's attention. A good example of this type of attacks is network *packet sniffing*, where attackers intercept personal information while data streams transit public networks. By exploiting security vulnerabilities in identity management or user authentication, an attacker can *impersonate* a legitimate user to obtain unauthorised access to personal information, as well as abuse super-user or administrator rights. Malicious software can be downloaded from web sites or received as email attachments, that when run can cause undesirable or/and unknown side-effects including release or exploitation of private information. Innocent looking software may be *Trojans* that enable remote access to the victim's computer, secretly collect personal information without informed consent and potentially send private information (e.g., files, logged keystrokes or web browsing histories) to perpetrators. Bruce Schneier observed that malicious security attacks are getting increasingly sophisticated and abstract, and he categorised them into three classes, i.e., physical, syntactic, and semantic attacks. Recent semantic attacks "target the way people assign meaning to content"

[Schneier00], and *phishing* is a typical example of semantic attacks. Phishing attacks deceive users into disclosing personal information by mimicking a legitimate entity in an electronic communication [James05]. Phishing has been a growing concern for personal information privacy, and 135 legitimate brands have been hijacked according to a report by the Anti-Phishing Working Group in February 2007 [Group]. Addressing these security threats is an active area of research in its own right [Dhamija06, Wu06], and it is out of scope of this thesis. *For the purposes of the thesis, we assume that the developers wish to behave ethically and respect user's privacy to encourage trust.*

Accidental or negligent release of private information occurs if a user's understanding of a system is inadequate. For example, mis-configuration of file access permissions may lead to unintentional sharing of personal information in multi-user computing systems; mis-configuration of security settings of messengers or blogs may incur undesirable exposure of private information. Inadvertent privacy intrusion occurs when consequences of user actions are hidden by system abstraction. When an administrator sets up a web interface for a shared file folder, the users who originally have access to that folder are unaware that the folder is now publicly exposed. People may accidentally publish private information as an unexpected side-effect of legitimate actions, e.g., sensitive credentials may be embedded in URIs to printer queues that are exposed when shared on the network and personal files may even be cached on printers and reprinted remotely. Personal information sharing applications have been designed to make it easy to publish information, but few provide tools for reminding users to stop sharing. Cognitive science taught us that humans are forgetful, and the asymmetry of information disclosure and control leads to human mistakes of accidental information leakage. *In the context of this thesis, we focused on exploring privacy management solutions for accidental or negligent privacy intrusion in distributed systems.* We provide more detailed description for defining the scope of the thesis in the next section.

1.6 Scope of the Thesis

Privacy is a complex socio-technical system that requires interdisciplinary research from the domains of sociology, psychology and computer science [Anderson04]. We believe that technologies can be used to reduce the risk of personal privacy violation but not completely eliminate it, because the technical systems designed for human purposes

are vulnerable to human weakness [Jones03]. We believe that personal information privacy cannot be achieved using technology alone, as a prominent cyber-law scholar, Stanford Law Professor Lawrence Lessig concluded: privacy has to be achieved through a combination of technologies, legislations, social norms, and market forces [Lessig98, Lessig99].

The area of privacy management is extremely broad and complex, and we have chosen to focus on provide privacy management support for applications that enable intentional sharing of personal information in networked computing environments.

People selectively share personal information with others in daily life to fulfil some social goals [Goldberg02], and computer-mediated personal information sharing becomes increasingly popular with the ubiquity of networks and wide adoption of distributed applications. In Computer Supported Cooperative Work (CSCW) systems, sharing individual activities among a user group is critical to successful coordination because it promotes awareness of the activities of others and provides a context of a user's own activity [Dourish92]. Previous studies [Arminen03, Weilenmann04] showed that people tend to ask others' location and situation at the beginning of a phone call, and failing to convey mutual contexts between people results in high proportion of unsuccessful communication attempts [Oulasvirta05]. The recent proliferation of Internet applications provides people means to exchange personal information both synchronously (e.g., instant messaging, audio chat, or video conferencing, etc) and asynchronously (e.g., email, discussion groups, wiki, blog, etc) [Swinth02]. People share personal information by maintaining personal blogs, where people document their personal life or express deeply felt emotions [Nardi04]. In using Instant Messenger (IM) clients, people often provide extra information in the display name field to allow others to know their mood, current location, current activities, or views in the form of personal commentaries [Smale05]. A large number of online communities have emerged to meet people's various personal, social, recreational, and professional needs [Smith99, Kim00, Rheingold00], and personal information sharing within those online communities help people to establish and maintain interpersonal connections with others [Swinth02]. More importantly, sharing dynamic personal contextual information such as location and activities are stepping stone to the emerging context-aware computing and UbiComp [Weiser91, Satyanarayanan03], and a number of research prototypes have been developed to promote social awareness among users [Bardram04, Raento05] to realise the goal for integrating computing capabilities into the physical environment.

Surveys [Mabley00] showed that a large percentage of users were typically willing to share some personal information with service providers and other users for receiving better services. However, numerous user studies [Harper96, Kaasinen03, Barkuus03] demonstrated that many people want to remain in control of their privacy. Existing information sharing applications provide very few options for people to control their private information [Hull03]. For example, Instant Messengers clients (e.g., MSN, Jabber, AOL Messenger, etc) only allow users to make all-or-nothing privacy decisions, i.e., controlling presence information based on buddy-lists. With increased personal information being sharing among people, simple all-or-nothing control will not be sufficient for enabling users to *"determine for themselves when, how, and to what extent information about them is communicated to others"* [Westin67].

In this thesis, we concentrate on designing information privacy management mechanisms in the scope of individual-to-individual interactions mediated by networked applications (as opposed to interactions between individuals and organisations). Based on a fundamental premise of the cognitive sciences that people are mostly rational [Simon96], we assume that developers and users involved in distributed information sharing applications will protect their own privacy and respect the privacy of others [Boyle05]. End users involved in those applications share their private information voluntarily, and there is no unbalanced power relationship between them. Moreover, users involved in those applications have already established social relationships between each other (as opposed to adversaries or attackers in traditional computer security), and therefore we focus on privacy management for *inadvertent privacy infractions* to avoid undesired social obligations or potentially embarrassing situations [Hong05].

1.7 Research Aims and Objectives

On the way to the UbiComp vision, increased availability of computing devices and network connectivity allow people to access many useful services that can improve their everyday lives. In using applications that support intentional sharing of personal information in networked environments, we observe that end users do not require the hiding of as much personal information as possible, rather they have a desired level of openness when disclosing personal information. More importantly, this desired level of openness varies with changes of circumstance, e.g., the recipient of information, the sensitivity

of information, the time of disclosure, the precision of the information, etc. We provide the following definition for better privacy in the context of the thesis:

> **Definition 4**: *Better privacy is when personal information disclosure is at a level of openness that is as close to a user's desired level as possible.*

The reciprocity of traditional face-to-face social interactions enables people to sense sufficient cues from the environment and from their expectations of social behaviour according to social norms [Boyle05]. However, recent development of technology for networked environments has changed this situation: people are increasingly unaware of personal information disclosure; and even if they are, they can hardly be expected to understand the intended or unintended consequences of the disclosure [Smith04]. Grudin argued that technology resulted *desituated* and *decontextualised* actions:

> *"We are losing control and knowledge of the consequences of our actions, because if what we do is represented digitally, it can appear anywhere and at any time in the future. We no longer control access to anything we disclose."* [Grudin01]

Without sufficient cues, people often fail to adjust their behaviour and appearance according to social norms and expectations, a common regulatory process in traditional social interactions known as *self-appropriation* [Bellotti97]. The strategies for self-appropriation in traditional social interactions are both *fine-grained* and *lightweight*, but very few fine-grained yet lightweight strategies exist in computer-mediated interactions [Bellotti97]. It is hard to design computing systems that provides fine-grained and lightweight control, because they are sometimes contradictory goals and require trade-off at the design stage [Boyle05].

In this thesis, we investigate potential solutions that enables regular users to effectively manage their privacy in networked computing environments. We believe that the proposed solution provides insight into the privacy problem that can be carried forward to future UbiComp environments. More specifically, we regard privacy management as a personal decision making process that involves both objective knowledge of changing environmental conditions and subjective views on disclosing personal information. In order for end users to achieve better privacy, we argue that systems should increase the transparency of the personal information usage and allow them to act on this objective

knowledge together with their subjective views through an adaptive approach, which optimises selective disclosure of personal information in response to changing circumstances through a mixture of system-initiation and user-initiation. More concretely, we provide the following three definitions:

> **Definition 5**: *A privacy aware application is an application that employs a privacy framework to ensure that the users' wish for privacy is respected.*

> **Definition 6**: *Adaptive privacy management is the process that a user and/or a system continuously adjusts the system behaviour of disclosing personal information according to his/her changing desire for openness under different circumstances.*

> **Definition 7**: *An adaptive privacy aware application is an application that attempts to achieve better privacy using a privacy framework that supports adaptive privacy management.*

This thesis aims to investigate the issues of incorporating adaptive privacy management into personal information sharing applications in networked environments that work within the existing social contexts. In particular, we explore principles that we believe are essential to constitute an adaptive privacy aware application with end users:

- *Adaptive Privacy Balance and Evolution of Privacy Preferences:* to enable users or/and the system to adjust the balance between openness and closedness depending on situations in dynamic networked environments; and to allow evolution of users' privacy preferences specified in the system over time as a result of on-going interactions between the user and the system.
- *Awareness of System Behaviour Concerning Privacy:* to promote users' awareness of system's behaviours concerning privacy, e.g., what the system can potentially or/and actually do with users' personal information.
- *Convenient and Timely Access to Privacy Controls:* to provide end users with convenient and timely access to privacy controls, in order to encourage them to adjust the system's behaviour of personal information disclosure in response to the change of circumstances

- *Balance between Privacy and User Involvement:* to balance end users' need for information privacy with the level of intrusiveness incurred by privacy-related interactions.

- *Accountability for Privacy-related Behaviour:* to maintain audit trails for privacy-related behaviours (e.g., information disclosed either explicitly by the user or automatically by the system) to increase accountability and traceability of the system.

In order to investigate the aforementioned principles, the thesis presents the design of a middleware platform that provides support for developing adaptive privacy aware applications. A prototype implementation of the platform is presented as well as an adaptive privacy aware application that enables sharing of GSM-based location information. To meet the aforementioned requirements, adaptive privacy aware applications built upon the platform promote privacy awareness through timely notification to end users of critical privacy events; enable multi-modal and multi-device interactions to provide users convenient and timely access to privacy controls; provide support for users to make privacy decisions 'in context' to enable users or/and the system to adaptively balance personal information disclosure; automate privacy decisions using privacy rules to strike a balance between privacy and user involvement; and realise persistence for privacy interactions to increase accountability and traceability of the system. We evaluate the principles of adaptive privacy management through an end user study of the adaptive privacy aware location sharing application.

1.8 Structure of the Thesis

The remainder of the thesis is structured as follows.

Chapter 2 presents an investigation into the nature of privacy itself and existing technical approaches for preserving information privacy. The chapter surveys privacy from historical, social, legal, and technological perspectives and provides important context for the thesis. In this chapter we motivate a new approach, i.e., adaptive privacy management, that enables people to adaptively adjust their level of openness in dynamic networked computing environments.

Chapter 3 critically analyses existing technical approaches to identify the limitations

of these approaches. Based on this analysis, the chapter explains rationales for selecting specific technical mechanisms for adaptive privacy management, to empower people to manage their privacy more efficiently and effectively in dynamic networked computing environments. The chapter concludes with a set of requirements that need to be satisfied in order to develop adaptive privacy aware applications.

Chapter 4 presents the design decisions for private information sharing applications to meet the requirements set in the previous chapter. To facilitate incorporating adaptive privacy management into distributed applications, a middleware platform was designed to support the development of adaptive privacy aware applications. The flexibility of the platform enables developers to customise it to meet requirements in different domain, and facilitates modification, extension and maintenance of applications.

Chapter 5 presents the implementation of a prototype platform that supports the development of adaptive privacy aware applications. The components of the support platform are examined in detail, particularly in terms of the application programming interfaces offered to distributed application developers and plug-in interfaces designed to facilitate the customisation of the platform and privacy request handling algorithms. Lastly, the chapter describes the implementation of a prototype application built using the platform that allows people to sharing GSM-based location information while preserving their privacy.

Chapter 6 presents an evaluation of adaptive privacy management approach from end user perspective. A three-phased user study was conducted during April to May in 2007 with 30 participants over a period of $7^1/_2$ weeks using our privacy aware location sharing application. The chapter describes the experimental methodology of the user study, provides analysis of the gathered usage data, and presents an evaluation of the principles constituting adaptive privacy management.

Finally, in the concluding chapter 7 we summarise the work presented in the whole thesis. The most important results are highlighted, and areas of possible future research are discussed. The chapter finishes with our concluding remarks reviewing the major contributions of our work.

CHAPTER II

Background

Contents

2.1 Overview

The objective of this chapter is to provide an overview of the privacy from historical, social, legal, and technological perspectives. It aims to provide important context for our work and to identify contributions and limitations of the existing technical approaches. The first section of the chapter provides the historical view of privacy, introducing different aspects of privacy and the implications for privacy as new technologies have been introduced. Next we examine the privacy issue from the social and legal perspectives, in order to better understand the social and legal context under which the technical mechanisms were applied. Armed with the background knowledge of privacy, we investigate the technical mechanisms for achieving information privacy in detail. This consists of the early research in access control and encryption, anonymity and pseudonymity, recent development in privacy transparency and awareness, privacy enforcement, and work in system support for building privacy aware applications. Last section of this chapter critically analyses the limitations of existing technical approaches (e.g., static-policy approach) and motivates the need for adaptive privacy management in dynamic networked environments.

2.2 Historical View of Privacy

The recognition of individual's right to privacy is deeply rooted in history. According to a report published by Privacy International [Laurant03], the earliest reference to privacy can be traced back to the Qur'an and in the sayings of Mohammed, and the Bible has numerous references to privacy. In 1361, the first legal protection of privacy, the Justices of the Peace Act, emerged in England, which provided for the arrest of peeping toms and eavesdroppers [Laurant03].

As early as 1765, British Parliamentarian William Pitt famously wrote, *"The poorest man may in his cottage bid defiance to all the force of the Crown. It may be frail; its roof may shake; the wind may blow through it; the storms may enter; the rain may enter — but the King of England cannot enter; all his forces dare not cross the threshold of the ruined tenement"*[Pitt78]. This is the first recognition of the privacy of home environment, and it has been extended to the notion of *territorial privacy*, which is about the setting of limits on intrusions into the domestic and other environments such as

workplace or public space [Laurant03]. With the advent of modern technology, violation of territorial privacy refers not only to the physical intrusions and searches of the places, but also to video or electronic surveillance and other remote access methods to physical properties.

Another important aspect of privacy is concerned with people's physical selves, called *bodily privacy*, which originally refers to the protection against invasive procedures such as strip and body searches. The implication of bodily privacy evolved with the technological developments, to encompass protective measures against medical tests such as genetic tests or drug testing [Laurant03].

2.2.1 Modern Privacy

Later in 19th century, the concept of privacy was extended to people's personal appearance, sayings, acts, beliefs, thoughts, emotions, sensations, etc. The formulation of the contemporary legal concept of privacy can be traced back to the seminal article *The Right to Privacy* [Warren90] by Samuel Warren and Louis Brandeis in the Harvard Law Review in 1890. They emphasized *"the right to be let alone"*, reacting fiercely to the unethical use of portable cameras and modern printing presses to facilitate the collection and public dissemination of information relating to individual's private life. It might be the first time that people realised that the advent of new technologies has significant impact on personal private life.

Communication Privacy

With the invention of telecommunication in late 19th century, most notably in the form of telegraph and telephone, the notion of *communication privacy* [Laurant03] has to be re-interpreted to cover not only the security of the conventional letters but also that of the new forms of communications. More recently, the concept of communication privacy has been extended to accommodate the protection of other forms of communication such as emails, web instant messages, SMS text messages, with the popularity of applications using Internet and mobile communication.

Information Privacy

Alan Westin's definition of privacy quoted in the previous chapter is often regarded as the most appropriate for *information privacy*, but his definition 1.1 is mainly from the perspective of the individual data subject. From the perspectives of data collector and data user, information privacy involves the establishment of rules governing the collection and handling of personal data [Laurant03]. In 1960s and 1970s, information privacy, sometimes known as *data protection*, became a hot topic when government departments introduced automated data processing systems, as the potential misuse of powerful computer systems prompted demands for those rules.

With wide adoption of modern information technology, computing systems have been extensively utilised to record increasing amount of personal information in every part of our daily life, e.g., credit card transactions, tax bills, medical records, library borrowing history, etc. Moreover, the development of multimedia technology makes it easy to record, maintain, and disseminate personal information in non-textual formats such as images, audios, and videos. Information privacy has become the most important aspect of privacy due to the pervasiveness of personal information in the Information Age. Moreover, information privacy is the one of the most challenging research topics that requires efforts from multiple disciplines such as psychology, sociology, economics, law, computer science.

We have discussed the impact of technology on the implication of information privacy by explaining the potential threats to information privacy at every stage of the information lifecycle in section 1.4. In the context of this thesis, we concentrate on studies of *information privacy*, and we use *privacy* and *personal information privacy* alternatively in the rest of the thesis to refer to *information privacy*.

2.2.2 Summary

From this historical overview of the privacy, we introduced four aspects of privacy, including territorial privacy, bodily privacy, communication privacy, and information privacy. More importantly, those aspects of privacy are not static concepts, but their implications evolve with the advances and availability of technology. With increasing amount of personal information being collected, maintained, and used by networked computing systems, information privacy becomes much harder to preserve in the net-

worked environments. In this thesis, we concentrate on how to assist users in managing their information privacy in the networked environments. Since information privacy is a complex multi-disciplinary topic, we will discuss it from social and legal perspectives in the next two sections, in order to better understand the context of applying technical mechanisms.

2.3 Social Perspective of Information Privacy

Previous experiences of developing privacy-sensitive information systems have taught computer security scientists that the success of information privacy is not just down to the technical measures. As security engineering expert Ross Anderson acknowledged that personal information privacy is a complex social-technical system, which requires both correct *incentives and policy* and right *mechanism and assurance* [Anderson04].

People's perception of privacy greatly influences their decisions to disclose personal information, and therefore understanding individual's perception of privacy helps to identify potential privacy risks for designing privacy-sensitive technical systems. Empirical user studies have been conducted to develop our understanding of the impact of technologies such as multimedia communications and UbiComp systems on people's perception of privacy. Like the research on perception of privacy, economics of information privacy studies the individuals' privacy decisions at micro-level, and regards individuals as economic agents that make rational decisions to maximise utilities or profit by selectively disclosing personal information. Moreover, economics of information privacy studies the individuals' privacy decisions at macro-level — the aggregate behaviour of the participating entities including data subjects, data collectors, and data users. We will review perception of privacy and economics of information privacy respectively in the following two subsections.

2.3.1 People's Perception of Privacy

Information technology has changed the reality of personal information privacy, due to its significant impact on every stage of the information lifecycle. Equally important, information technology has great influence on people's perception of privacy. Privacy decisions are mostly personal choices, and they largely rely on whether people per-

ceive themselves to be private [Adams01b]. Research on people's perception of privacy bridges between computer science and social psychology, and recent work in this area has been concentrated on empirical user studies from the field of Human Computer Interactions (HCI) [Bellotti93, Adams01b, Beckwith03, Lederer03a, Consolvo05, Olson05, Chatfield05].

In early nineties, research efforts were focused on the investigation of people's perception of privacy within computer meditated communication environments, especially multimedia communication environments. With the experience in deploying an audio and video communication infrastructure into the working environment, Bellotti and Sellen [Bellotti93] identified that lack of control and feedback on information captured by the system tends to break the technologically mediated social interactions, which may *"foster unethical use of the technology"* [Bellotti93] and be *"much more conducive to inadvertent intrusion on privacy"* [Heath91].

In late nineties, Anne Adams [Adams01a] carried out systematic research on people's privacy perception in multimedia communications, and generated a users' privacy perception model using the well-known Grounded Theory Methodology [Open Sources07d] from the domain of social psychology. Her privacy perception model identified that three factors — information sensitivity, information receiver, and information usage — which interplay to form the users' overall perception of privacy, and privacy invasions often occur when users realise that there is a mismatch between their perception and reality of privacy [Adams01b].

In seeking Weiser's vision of UbiComp, HCI researchers have conducted many empirical user studies in UbiComp environments, especially the so-called intelligent environments, in order to explore how people understand the UbiComp technology and its implication on the privacy of their personal information.

Beckwith [Beckwith03] carried out a user study in a working UbiComp environment situated in an eldercare facility, aiming to understand the perception of personal information privacy for different user groups, including residents, their families, and the facility's staffs. The resulting semi-structured interviews and informal observations raised more questions rather than answers for system designers, as the author concluded that *"reliable, inconspicuous sensing of personal information is problematic because users do not always understand the extent or methods of data collection and thus cannot adequately evaluate privacy issues."* [Beckwith03]

Lederer et al. [Lederer03a] conducted a questionnaire-based user study to investigate the relative importance of two factors, the identity of the information inquirer and the user's situation at the time of inquiry, in determining the accuracy that the user preferred while releasing his personal information through a UbiComp system. The finding shows that people consider identity as a stronger determinant than situation for determining their privacy preferences [Lederer03a].

Consolvo et al. [Consolvo05] conducted a three-phased formative user study on people's willingness to disclose their location information to social relations. The results show that the identity of the information requester (i.e., who), the proposed purpose of the request (i.e., why), and the quality of information (i.e., what levels of detail) that is most useful for the requester are the most important factors for people to decide whether to disclose their location. Consistent with the previous work by Lederer et al. [Lederer03a], this user study confirmed that current activity and mood are relatively less important factors than requester identity and purpose for people to make location information disclosure.

Olson et al. [Olson05] studied people's behaviour in sharing different types of personal information (e.g., age, email address, credit card details, current location, etc) with different social relations. In a survey, participants were asked to rate their willingness to share 40 types of information to 19 types of people. The results of the study showed that most people cluster the persons to whom they want to share their personal information with into a similar set of categories, i.e., public, coworkers, manager and a trusted coworker, family, and spouse. The findings suggested that privacy preference for information sharing can be set based on clusters of requesters to reduce the overhead of privacy management.

Focusing on a user study of personalisation in intelligent environment, Chatfield et al [Chatfield05] found users would like to have control over the receiver of their personal information and effective feedback on information usage and dissemination. He argues that intelligent environments should provide information to promote user's understanding of the technological impacts on their privacy so that they can evaluate potential risks to their privacy and the expected benefits of accessing personalised services [Chatfield05].

2.3.2 Economics of Information Privacy

Economics of information privacy regards personal information as a property and uses the language of economics to explain that many privacy failures are due to perverse economic incentives rather than the lack of technical mechanisms [Anderson04].

Computer security experts have spent over two decades developing sophisticated Privacy Enhancing Technologies (PETs) to achieve information privacy for Internet users. Most of the PETs are technically sound and mature, but end user adoption and commercial uptake have been very disappointing [Goldberg02]. Although issues of poor integration and usability do exist in some PET products [Whitten99], the most compelling explanations for the failure use economic arguments to dissect this privacy dilemma [Danezis05]. By gathering detailed user information, companies are able to determine consumers' willingness to pay, and therefore charge different prices to various consumer groups for the sales of identical goods or services. It is the growing incentive to price discriminate and the increasing ability to do so due to the modern technology, that prevented PETs from being widely adopted and deployed by commercial organisations [Odlyzko03].

An analytic framework [Acquisti03] was proposed to reason about the economics of anonymity infrastructures, and it provides some guidelines on how to balance the incentives of the different parties involved so that they all benefit from more anonymity. In fact, hardly any technology, including PETs, will reach widespread adoption unless correct economic incentives have been aligned for the different parties involved [Acquisti04].

Economic analysis of information security and privacy became one of the emerging research fields, and it can be used to explain other privacy dilemmas more clearly and persuasively. One of the many privacy puzzles is that even though people show great concerns about loss of privacy, they are not doing much to protect themselves, e.g., the failure of end user adoption of PETs. Acquisti [Acquisti04] argues that from the economic perspective those who value their privacy but take no action to protect them are actually behaving rationally: they discount the potential losses caused by the misuse of their personal information with the uncertain probability that such an outcome will take place, then compare the resulting value with the total cost of using certain PET, and finally decide not to use it.

At micro-level, economic analysis has been extended to understand the people's privacy preferences by regarding people as rational economic agents, who tend to maximise the utility or profit by making choices based on available information. But recent theoretical studies and user surveys showed that people often take privacy-sensitive decisions under incomplete information and with significant uncertainties about the consequences of their actions [Acquisti05a]. Even if people had equipped with complete information, they are unable to make optimal privacy decisions on large amounts of complex data because of human's bounded rationality, which limits their ability to acquire, memorise, and process all relevant information [Acquisti05a]. Even if individuals could compute the optimal strategies for their privacy decisions with unbounded rationality and complete information, they often fail to behave according to optimal strategies because of various forms of systematic psychological deviations from rationality that have been extensively documented in economic and psychological literature, e.g., hyperbolic discounting, underinsurance, optimism bias, self-control problems, immediate gratification, etc [Acquisti05a].

More recently, empirical research has been conducted in the form of extensive user surveys, to measure the quantitative value of certain piece of private information for individuals. By conducting reverse second-price auctions on personal information such as weight and age over 127 participants, Huberman et al [Huberman05] concluded that people value a piece of personal information more if the revealed trait is more undesirable to them with respect to the group, whether perceived or actual. For example, individuals whose weights are over average request higher monetary values to reveal their wrights. This work demonstrated the contextual nature of the privacy-related decisions. Conducted over a group of undergraduate students in Cambridge, recent survey on location privacy preferences revealed that the value of the students' precise location information largely depends on their travel patterns and the persons they communicate with [Danezis05].

However, information differs from ordinary goods or properties, and the value of private information can be influenced by the subjective evaluation of the consequences due to the loss of privacy. Experimental evidence [Acquisti05b] showed that framing a marketing offer from absolute price to percentage discount has significant impact on the value of the same piece of personal information for same person, and Alessandro et al. argued that the uncertainty and ambiguity of the negative consequences of losing privacy are among the most important reasons to explain this phenomenon [Acquisti05b].

2.3.3 Summary

People's perception of privacy has great impact on their actual behaviours on disclosing personal information, since privacy decisions are mostly personal choices that largely rely on whether people perceive themselves to be private. The networked computing systems are becoming increasingly unobtrusive and the complexity of those systems have reached to the point of incomprehension [Smith04]. Therefore, we argue that we need to empower the users to preserve their information privacy by promoting their awareness of privacy and assisting them in understanding the consequences of their activities in the networked environments.

Economics of information privacy helps to explain that many privacy failures are due to perverse incentives rather than the lack of technical mechanisms [Anderson04]. To make privacy technologies succeed beyond research prototypes, we need to correctly align the incentives of the participating parties. The micro-level economic analysis has more direct influence on our work. In particular, we aim to empower the user by increasing their awareness of privacy and assisting them in understanding the consequences of their actions, to counteract the effects of incomplete information while people make privacy-sensitive decisions. Moreover, user studies on quantitative values of personal information highlighted the contextual nature of privacy decisions, and it prompts us to propose dynamic and flexible approaches for privacy management in the networked environments.

2.4 Legal Prespective of Information Privacy

The first data protection (or information privacy) law in the world was enacted in the Land of Hesse in Germany in 1970 [Laurant03]. It was in the same decade that national laws on information privacy were passed in Sweden (1973), the United States (1974), Germany (1977), and France (1978) [Laurant03].

There are two types of privacy legislations: *comprehensive law* and *sectional law*. Many European countries have a comprehensive law that governs the collection, use, and dissemination of personal information by both the public and private sectors [Laurant03]. Some countries, such as the United States, have no comprehensive legal framework for information privacy, but instead they enact sectional laws governing specific types of

private information, e.g., video rental record, financial report, medical data, etc. In the sectional law approach, new legislations are often required with the introduction of new technologies, and therefore legal protections always lag behind technology. In US, there is still no legal protection for personal information on the Internet. Many countries have sectional laws as a complement for the comprehensive law by providing more detailed protections for certain categories of personal information [Laurant03].

Evolved from the legislations around the world, an important consensus on information privacy was reached at the international level — *the Organization for Economic Cooperation and Development's (OECD) Guidelines Governing the Protection of Privacy and Transborder Data Flows of Personal Data* [OECD80]. OECD's guidelines formulated a set of *Fair Information Practices (FIP)* principles, which described rules on collecting, maintaining, using and disseminating personal information. The most comprehensive information privacy legislation, *Europe Union's Directive 95/46/EC* was greatly influenced by the FIP principles. In this section, we first introduce the FIP principles and the Europe Union's Directive 95/46/EC, and then discuss the impact of legal frameworks on designing technical solutions for information privacy.

2.4.1 Fair Information Practice Principles

The notion of Fair Information Practice principles was first articulated in a report entitled *Records, Computers and the Rights of Citizens* [US Dept. of Health73] by the US Department of Health, Education and Welfare (renamed to Department of Health and Human Services) in 1973. The report formulated a list of requirements for maintaining and processing personal data, which became the major ingredients of the US Privacy Act of 1974 as well as the privacy legislations worldwide. Later in 1980, *the Organization for Economic Co-operation and Development (OECD)* further developed the original requirements and proposed eight FIP principles in *the OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data* [OECD80]. These are the widely accepted FIP principles we refer to today, and they have been working as the foundation for national privacy laws in the United States, Canada, Europe and other parts of the world. We quote here the eight FIP principles verbatim from OECD's Guidelines [OECD80]:

1. Collection Limitation Principle There should be limits to the collection of per-

sonal data and any such data should be obtained by lawful and fair means and, where appropriate, with the knowledge or consent of the data subject.

2. Data Quality Principle Personal data should be relevant to the purposes for which they are to be used, and, to the extent necessary for those purposes, should be accurate, complete and kept up-to-date.

3. Purpose Specification Principle The purposes for which personal data are collected should be specified not later than at the time of data collection and the subsequent use limited to the fulfilment of those purposes or such others as are not incompatible with those purposes and as are specified on each occasion of change of purpose.

4. Use Limitation Principle Personal data should not be disclosed, made available or otherwise used for purposes other than those specified in accordance with the Purpose Specification Principle except:

    a) with the consent of the data subject; or

    b) by the authority of law.

5. Security Safeguards Principle Personal data should be protected by reasonable security safeguards against such risks as loss or unauthorised access, destruction, use, modification or disclosure of data.

6. Openness Principle There should be a general policy of openness about developments, practices and policies with respect to personal data. Means should be readily available of establishing the existence and nature of personal data, and the main purposes of their use, as well as the identity and usual residence of the data controller.

7. Individual Participation Principle An individual should have the right:

    a) to obtain from a data controller, or otherwise, confirmation of whether or not the data controller has data relating to him;

    b) to have communicated to him, data relating to him

        i) within a reasonable time;

        ii) at a charge, if any, that is not excessive;

        iii) in a reasonable manner; and

iv) in a form that is readily intelligible to him;

c) to be given reasons if a request made under subparagraphs (a) and (b) is denied, and to be able to challenge such denial; and

d) to challenge data relating to him and, if the challenge is successful to have the data erased, rectified, completed or amended.

8. Accountability Principle A data controller should be accountable for complying with measures which give effect to the principles stated above.

2.4.2 Europe Union's Directive 95/46/EC

Most European countries have their own legislation on information privacy, and the diversity of these legislations impeded the free flow of personal data within the European Union. Therefore, the European Union proposed *the Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data* [CDT95] to harmonise data protection regulation within EU member states [Open Sources07c]. The Directive regulates the processing of personal data, where processing encompasses "any operation or set of operations which is performed upon personal data, whether or not by automatic means, such as collection, recording, organization, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, blocking, erasure or destruction" [CDT95].

Influenced by the FIP, European Union's Directive 95/46/EC is the most comprehensive information privacy legislation in the world. The principles of the EU Directive fall into three categories: transparency, legitimate purpose, and proportionality [Open Sources07c]. To ensure transparency, the data collector must provide required information such as identity, purpose, and data users to the data subject, in order to ensure fair processing the personal data. Personal data must be *"collected for specified, explicit and legitimate purposes"* and may not be further processed in a way incompatible with those purposes. The principle of proportionality is a fundamental principle of European Union law, and it states that personal data should be processed only insofar as it is *"adequate, relevant and not excessive in relation to the purposes for which they are collected and/or further processed"*.

2.4.3 Impact of Legal Framework

The importance of these FIP principles are not only restricted as the foundation of national or regional privacy legislations for governments, but also as the basis of any self-regulatory process or privacy policy creation for companies or industrial bodies. Moreover, these principles have been worked as guidelines for building automated computer systems and Internet applications that handle information related to individuals.

In 1998, based on a review of a series of reports, guidelines, and model codes by US, Canada, and Europe governments that incorporated the widely-accepted principles concerning fair information practices, the US Federal Trade Commission proposed the following five core principles for online privacy in a report to Congress [FTC98], including *"notice and awareness", "choice and consent", "access and participation", "integrity and security", "and enforcement and redress"*.

More recently, Langheinrich[Langheinrich01] extended the Fair Information Practice principles to the area of UbiComp, and proposed the following six guidelines for developing privacy-aware UbiComp systems, including *"notice, choice and consent", "proximity and locality", "anonymity and pseudonymity", "security", and "access and recourse"*.

Iachello and Abowd [Iachello05] adapted the principle of proportionality to a design framework to create privacy-friendly Ubicomp applications and services. Their design framework consists of three stages, i.e., the establishment of usefulness or legitimacy of the application goals, the evaluation of the appropriateness of the alterative implementing technologies and techniques, and the fine-grained adjustment of technical parameters to make them adequate to application goals and acceptable for the data subject.

2.4.4 Summary

The principles embodied in the legal frameworks have great impact on the design of technological solutions for privacy-sensitive applications. The FIP principles have been adapted to work as guidelines for building automated computer systems and Internet applications that collect and process personal information. Inspired by the legal principle of proportionality, Iachello and Abowd proposed the three-stage design framework for developing UbiComp applications and services. In our work, we aim to use FIP

principles as guidelines for design and evaluation, as well as criteria for clarifying the scope of our research. We will directly address *Collection Limitation Principle, Data Quality Principle, Openness Principle and Accountability Principle*, and assume the legal framework to enforce *Purpose Specification Principle and Use Limitation Principle*. We do not address *Security Safeguards Principle and Individual Participation Principle* in our work, but we will propose methods to incorporate them into our framework.

2.5 Technical Mechanisms for Privacy

With background knowledge on information privacy from social and legal perspectives from previous sections, we provide an overview of the technical mechanisms that exist for helping users maintain their information privacy in this section. We describe technical mechanisms in the following categories: access control and encryption, anonymity and pseudonymity for the Internet, transparency and awareness using machine-readable privacy policies, privacy enforcement architecture, and system support for developing privacy-sensitive applications.

2.5.1 Access Control and Encryption

As the traditional cornerstone of computer security, access control determines which principals (e.g., person or software process) have to access to which system resources (e.g., file or directory) [Anderson01]. To protect information privacy, one principal can impose restrictions on other principals to retrieve certain piece of personal information. Access control works at different levels in a system, from the hardware through operating system and middleware to the application layer [Anderson01]. In this section, we introduce the concept of the access control matrix and its implementation alternatives, i.e., access control lists and capabilities. Then we describe different models of access control, including discretionary, mandatory, and role-based access control. Finally, we provide an overview on encryption techniques used in conjunction with different models of access control.

Access Control Matrix

In computer security, an *object* is an abstraction of all kinds of resources in a computing system, e.g., files, programs, devices, etc. *Subjects* are normally users or software processes executing on behalf of users, which initiate actions or operations on objects within a system. It is important to note that subjects can themselves be objects. In other words, an initiator of one operation can be the target of another [Sandhu94]. For example, a software process in a modern operating system can create child processes in order to accomplish a computing task. These child processes are objects because the parent process can initiate operations such as suspension or termination on them. At the same time, these child processes are subjects because they might initiate operations such as reading or writing on a system file.

An *access control matrix* is a conceptual model which specifies the access rights (e.g., read, write, execute, etc) that each subject possesses for each object [Sandhu94]. It was originally proposed by Lampson [Lampson74], and was further developed to protect resources in operating systems [Graham72, Harrison75]. An access control matrix has a row for each subject and a column for each object. Each element in the matrix specifies the access rights that have been authorised for the subject in the row to the object in the column. The aim of access control is to ensure that subjects can only perform operations on objects that have been authorised by the access control matrix. It is worth mentioning that a prerequisite for access control is authentication. *Authentication* refers to the process of establishing the identity of one principal to another, e.g., establishing a user's identity to a system or application using passwords. The access control matrix model clearly separates the problem of authentication with that of authorisation [Sandhu94].

In practice, most systems do not implement the access control matrix directly, because an access matrix is normally very large and sparse (i.e., most of its elements are empty). One of the most popular approaches for implementing the matrix, called *Access Control Lists (ACLs)* is essentially storing the matrix by columns. Each object is associated with an ACL, maintaining the authorised operations that each subject (e.g., user) in the system can perform on the object. Modern operating systems such as Unix or Windows employ the ACL-based approach for doing access control. An opposite way for implementing the matrix, by storing the rows and using subjects as indexes, is called *Capabilities*. Each subject is associated with a capability list, maintaining the authorised

operations that the subject can perform on each object in the system. IBM AS/400 series systems employed capability-based access control, and Windows 2000 combined capabilities with ACLs to gain the benefits of both [Anderson01]. Another implementation alternative for the access control matrix is *Authorisation Relation*, which is basically a relationship database table maintaining access relationships between subjects and objects. Each row (or tuple) of an authorisation relation normally contains a subject, an object, and a single access right of the subject on the object, and database operations are required to determine whether certain access rights are allowed or not. Relational database management systems typically employ this approach.

Models of Access Control

Different models of access control were designed to meet security and system requirements in different operational environments. In this section, we discuss three different models of access control and their applications for protecting information privacy. *Discretionary Access Control (DAC)* was widely used in commercial and industrial environments, which require flexible control and demand reasonable level of protection. *Mandatory Access Control (MAC)* was initially developed for military environments that demand high level of protection on confidentiality of information. DAC and MAC were both included in the Trusted Computer System Evaluation Criteria (TCSEC) [Open Sources07e] published by US Department of Defense (DoD) in 1985, and therefore they are often referred to as traditional or classical access control models. *Role-based Access Control (RBAC)* was introduced by David Ferraiolo and Richard Kuhn in 1992, and was standardised by American National Standards Institute (NIST) in 2004. RBAC has become the predominant access control model due to its generality and flexibility.

DAC was defined by the TCSEC [Open Sources07e] as *"a means of restricting access to objects based on the identity of subjects and/or groups to which they belong. The controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject (unless restrained by mandatory access control)."* DAC is flexible since it allows individual users to grant or revoke access privileges of any objects under their control [Ferraiolo92]. However, DAC does not provide any assurance on the flow of information within a system, because it does not impose any restriction on the usage of informa-

tion once a user has got access to it [Sandhu94]. For example, a user who can read a file in a system can freely disseminate it to other users, who might not posses the authorised access privilege to read it.

Driven by the demand for higher level of security, MAC was proposed to meet the requirements for handling sensitive information in military or governmental environments. MAC was defined by the TCSEC [Open Sources07e] as *"a means of restricting access to objects based on the sensitivity (as represented by a label) of the information contained in the objects and the formal authorization (i.e., clearance) of subjects to access information of such sensitivity."* In MAC, each subject and each object in a system is assigned a security level, which can be an element of a partially ordered set. Typical security labels used in military environments include Top Secret (TS), Secret(S), Confidential (C), and Unclassified (U), where each label dominates itself and the ones after it. To prevent information in high level objects from flowing to low level objects, two properties have to been ensured by the system. First, a subject's clearance must dominate the security label of the object being read (i.e., no read up). Second, a subject's clearance must be dominated by the security label of the object being written (i.e., no write down). These two properties was proposed in Bell-LaPadula model [Anderson01], which is an instance of MAC to protect the confidentiality of information. There are products using MAC outside military environments and most of them are modified version of Unix, e.g., AT&T's System V/MLS, Security-Enhanced Linux (SELinux), AppArmor in SUSE Linux, etc [Anderson01].

RBAC model introduced an additional layer of indirection, i.e., roles, between subjects and the access privileges. It is important to note that roles are different from group of subjects or users, because roles possess specification of access privileges. Sandhu et al. [Sandhu94] defined role as *"a set of actions and responsibilities associated with a particular working activity."* In the RBAC model, access privileges on objects are specified for roles instead of for subjects in the system, and subjects are authorised to adopt roles after successful authentication. The RBAC model greatly simplifies security management by splitting the specification of user authorisation into two independent tasks: assigning subjects to roles and assigning access privileges to for objects to roles [Sandhu94]. Moreover, Osborn et al. [Osborn00] demonstrated that the RBAC model can be configured to represent both the DAC and MAC models, and therefore they justified the claim that the RBAC model is more general than both DAC and MAC. Due to its flexibility and generality, the RBAC model has become the predominant model

of access control, and it has been widely used in both commercial and non-commercial systems and applications, such as Microsoft Active Directory, FreeBSD, Solaris, and Oracle database management system. Ferraiolo et al. [Ferraiolo01] proposed a unified RBAC model by combining ideas from previous RBAC models in various commercial products and research prototypes, and the proposal was approved as a standard by NIST in 2004.

Covington et al. extended the basic RBAC model by introducing a new type of role called *Environmental Roles*, which can be used to capture environmental contexts related to access control [Covington01]. Environmental roles are activated when environmental conditions specified in the role are met. For example, a system might define environmental roles such as "high network bandwidth (over 50%)", "Sunday morning", or "weekdays". The system must gather contextual information (e.g., derived from sensory data) to determine which environmental roles are active at time of an access request. Therefore, this extended RBAC model can potentially be employed to enforce privacy constraints in dynamic networked environments. Osbakk et al [Osbakk04] introduced the concept of a *Privacy Invasion Value (PIV)* into the basic RBAC model. The PIV represents the extent of a privacy invasion for information disclosure, and it was based on factors other than information requester and purpose, e.g., time of release, environmental conditions, etc. By defining information requester and purpose as roles in RBAC model and replacing other privacy factors with PIV, the authors claimed that the proposed model has the potential to facilitate the task of privacy management in dynamic context-aware environments.

Cryptographic Protection Mechanisms

In implementing different models of access control, cryptographic mechanisms are often employed to protect the personal information from improper disclosure and modification (known as confidentiality and integrity respectively) [Sandhu97]. Encryption is the transformation of data from the original (i.e., the plaintext) to a difficult-to-interpret format (i.e., the ciphertext), and the reversible transformation is called decryption [Answers Corp.07]. Formally, an encryption function is a bijection between a set of plaintext messages and ciphertext messages, and therefore the decryption function is the inverse function of encryption [Beresford05]. Cryptographic keys control operations of encryption and decryption functions, and the security of a system should only rely on

the cryptographic keys according to Kerckhoffs' principle[1].

Traditionally, both encryption and decryption functions use the same cryptographic key for encrypting and decrypting messages. This system is often referred to as *secret key or symmetric key cryptography system*. The problem with the secret key cryptography system is the key distribution, i.e., a single key has to be securely delivered to the users before they are able to communicate confidentially. This problem can be solved by the *public key or asymmetric key cryptography system* originally proposed by Diffie and Hellman [Diffie76]. In public key cryptography system, a pair of cryptographic keys are used for encryption and decryption functions, and the mathematical property of the key pair ensures that it is infeasible to infer one from another. Typically, the key for encryption (i.e., public key) is known to everyone, and the key for decryption (i.e., secret or private key) must be kept secret to the owner. Therefore, confidentiality of information can be protected by encrypting a message using the public key, because only the owner of the corresponding private key can decrypt the ciphertext and read the plaintext. Moreover, the system can be used in a reverse way to maintain the integrity of the information: applying decryption function with the private key generates a ciphertext that everyone can decode using the corresponding public key but can only be produced by the owner of the private key. In practice, decryption function is not directly applied on the original message, but on the message digest (i.e., output of cryptographic hash function on the original message). The generated ciphertext is often called the *Digital Signature* on the message.

In real-world systems or applications, asymmetric cryptography systems are used to exchange shared secret keys (i.e. session keys) for symmetric cryptography systems, which are then employed to encrypt messages transmitted between different parties. This is because symmetric encryption functions are normally much more efficient than asymmetric encryption functions. Secure Socket Layer (SSL) and its successor Transport Layer Security (TLS) are most popular cryptographic protocols providing secure communication on the Internet. Because they sit just above TCP or UDP protocol, they can be used for securing different application layer protocols such as HTTP, SMTP, or FTP. SSL or TLS involves three operational phases: peer negotiation for cryptographic algorithms, key exchange and server authentication using asymmetric key cryptography system, and traffic encryption using symmetric key cryptography system. The original

[1] Kerckhoffs' principle states that system designers should assume that the entire design of a security system is known to all attackers, with the exception of the cryptographic key: "the security of a cipher resides entirely in the key". Claude Shannon rephrased it as "the enemy knows the system".

version of Pretty Good Privacy (PGP) designed by Philip Zimmermann in 1991 is an application program that provides reasonable level of privacy for email messages and attachments. Zimmermann remarked, *"PGP empowers people to take their privacy into their own hands. There has been a growing social need for it. That's why I wrote it"* [Zimmermann91]. PGP employed asymmetric key cryptography system to establish a session key between communication parties, and used the session key to encrypt the messages. PGP products [PGP Corp.07] have been diversified to include digital signature, whole disk encryption, secure shredding of deleted files, networked shared folder access control, instant messenger conversation protection, etc. In summary, PGP can not only protect personal information transferring over insecure network like SSL, but also protect personal information in long-term data storage.

Summary

Access control and encryption mechanisms are the predominant ways for protecting personal information from unauthorised disclosure and modification. Traditionally, the task of deploying access control mechanisms is the responsibility of system administrators, because configuration of access control parameters can be difficult and error-prone [Beresford05]. A case study on PGP 5.0 revealed that most ordinary people with little initial knowledge of computer security failed to effectively use PGP for protecting information privacy for their emails [Whitten99]. In the last decade, very small percentage of users adopted security features built in major email clients to protect information privacy, and Hallam-Baker concluded that it is mainly because of usability problems of the security mechanisms in those programs [Hallam-Baker06]. *We believe that access control is still one of the most important enabling technologies for achieving information privacy, and we will employ them in our research as an underlying technical mechanism.* However, no access control is effective unless it is used properly [Pfleeger02]. We do not focus on the access control itself in our research, but on creating an environment that enables people to use this security mechanism more effectively to manage their information privacy.

2.5.2 Anonymity and Pseudonymity

Information privacy over public networks means not only preventing others from knowing the content of the information being exchanged, but also keeping the identity of the sender and receiver unknown from eavesdroppers. While simple application of cryptography can protect the confidentiality of the information, it is far more difficult to hide *"who is talking to whom, and how often"* [Goldschlag99] from traffic analysis. Anonymity removes a person's privacy-related information and makes it impossible to identify the person within a group of users, by concealing the person's real identity, characteristics or significant features.

Pfitzmann and Waidner [Pfitzmann87] classified anonymity into three different types: *sender anonymity*, the identity of the party who sends a message is hidden; *receiver anonymity*, the identity of the party who receives a message is hidden; and *unlinkability of sender and receiver*, the fact that the sender and receiver communicate with each other cannot be identified. Sometimes, total anonymity over the Internet can be undesirable as long-term relationships (such as reputation) with other entities cannot be established. Combining the advantages of having a known identity with the benefits of anonymity, *pseudonymity* provides a degree of accountability by granting each user a *pseudonym*, while the user's real identity still remains anonymous. In this section, we review the anonymity and pseudonymity technologies from two major application areas, anonymous emails and anonymous networks.

Anonymous Emails

Email has been the most important distributed application at the dawn of the Internet age. The wide adoption and popularity of email bring concerns on information privacy, not only for the content of the email, but for the identities of the sender and receiver. A milestone in the area of email anonymity is the introduction of anonymous remailers [Bacard]. In addition to the forwarding functionality in normal email servers, anonymous remailers automatically strip away identifiable information (e.g., real name and email address) from the email header, and replace the data with dummy information (e.g., pseudonym and dummy address). In a survey paper published in 1997, Goldberg [Goldberg97] classifies the anonymous remailers into three types according to their levels of sophistication and security.

Type0 remailers, e.g., "anon.penet.fi", support sender anonymity by providing basic functionalities of stripping information in email headers that might identify the user and resending. This type of remailer assigns each user a random pseudonym, and maintains a *secret identity table* mapping the user's real email address with his pseudonym. To achieve recipient anonymity, the remailer relays replies to a pseudonym to the user's real email address by looking up into the mapping table. This type of remailer has the following disadvantages: first, users must trust the remailer not to reveal their real identities while sending email through it; second, the anonymity of pseudonymous users relies on the confidentiality of the secret identity table; third, this type of remailer does not prevent traffic analysis attacks that match up incoming and outgoing messages to learn the identities of the senders and receivers. In 1995, the operator of "anon.penet.fi", Johan Helsingius, was forced to reveal the identity of one user under the legal pressure from the Finnish government, and one year later he shut down the famous remailer to prevent against further legal attacks [Helmers97].

TypeI remailers, or *cyberpunk-style remailers*, were designed to solve the problems of single point of failure in type0 remailers. First, support for pseudonyms was abandoned, and therefore no security identity table had to be maintained. More importantly, TypeI remailers do not operate alone, but collaborate to achieve more robust security. In this type of remailer, a user does not send an email via a single remailer, but selects a chain of remailers and arranges his email being relayed through these remailers before it arriving at the recipient. Taking the advantage of cryptography, users can ensure that each remailer in the chain can only know the address of the previous one and the next one, but not the ones further down. An attacker must compromise every remailer in a chain in order to reveal the identity of the sender. Although TypeI remailers can randomly reorder outgoing messages to prevent correlations of ciphertexts, they are still vulnerable to traffic analysis, e.g., examining the size of the encrypted messages.

To prevent this type of attacks, *TypeII remailers*, or *Mixmaster remailers*, explored David Chaum's idea of digital Mix [Chaum81]. In particular, a Mix have the following properties: messages are padded or fragmented into uniformly sized; incoming and outgoing messages are encrypted with different keys; messages are batched and reordered lexicographically; and replay of incoming messages is prevented by removing redundant copies from a particular batch and time-stamping each batch. While constant length message prevents passive correlation attacks by comparing the incoming and outgoing message size, message reordering stops passive correlation attacks based

on timing coincidences. Through the chain of Mixes, each remailer can inject randomly generated dummy packets to hide real messages among noisy traffic. To support pseudonymity, TypeII remailers exploited the benefits of *newnym-style nymservers* [Mazières98], which grant each user a pseudonym without maintaining his real email address. Instead, newnym-style nymservers associate a pseudonym with a *reply block*, which repeatedly encrypts and nests addresses of a chain of TypeI remailers. An attacker must compromise all the remailers mentioned in the reply block in order to determine the email address associated with a pseudonym. Strong recipient anonymity were achieved by a simple mechanism called *message pool*, where senders send encrypted messages to a mailing list or newsgroup so that the recipient is hidden in the readers of the message pool.

Anonymous Networks

The experience of developing anonymous remailers helped to identify the principles of building general anonymity services. In late 1990s, the growing popularity of the World Wide Web (WWW) leads to further research on anonymity for Internet applications, and more generally, anonymous networks. The Internet Protocol (IP) has not been designed to take into account the information privacy issue: IP neither hides the packet itself nor the route that the packet takes through the network. By packet sniffing over the IP network, an eavesdropper can not only learn the content of the packets, but also other information that has the potential of identifying the endpoints of the communication, e.g., IP addresses of sender and receiver, the length of data being exchanged, and the time and frequency of exchanges.

Early efforts have focused on achieving anonymous Web transactions by interposing an additional third party (a special web proxy) between the sender and the receiver. If a sender wants to contact a receiver without revealing its identity, it sends packets to the proxy, which strips the identity information (e.g., IP address) from the packets and forwards them on. All the receiver knows is the proxy's address, and it has no clue of who the original sender is. Examples of such proxies include *Anonymizer* [Anonymizer Inc.07] and the *Lucent Personalized Web Assistant (LPWA)* [Lucent Technologies98]. In addition, the Anonymizer can remove identifying information in the data stream, and the LPWA can provide multiple anonymous identities for each user. A proxy-based approach provides adequate anonymity in many cases, and good usability making it more

popular than its sophisticated cousins. In principle, these systems are roughly equivalent to type0 anonymous remailers [Goldberg02], and therefore they exhibit the same weakness. The proxy itself can determine the user's identity, and it is also a single point of failure.

Onion Routing [Goldschlag99] exploited the idea of Chaum's Mix to provide anonymity protection for communication over the Internet. It provides anonymous bi-directional connections for both connection-oriented and connectionless traffic. In the Onion Routing network, there exist several distributed *onion-routers* (application level proxies), which are implementations of Chaum's Mix. Before sending packets through the Onion Routing network, the sender needs to determine a route through a series of onion-routers. After the initialization, the sender creates an *onion*, which is a layered data structure (recursively encrypted using the public keys of the onion-routers) that specifies properties of the connection at each point along the route. Each layer of encryption of the onion is stripped off by the onion-routers along the established path. Since the onion-routers are built on the concept of Mix, it pads or fragments packets to fix-length, performs cryptographic transformations on them, and forwards them to the next destination in a random order. The core onion-routers in Onion Routing networks are supposed to be under different administrative boundaries, in order to make it more difficult to breakdown the network or compromise a user's privacy.

Unlike Onion Routing, the *Crowds* [Reiter98] protocol was designed by assuming a different threat model, focusing on protecting against individual adversaries, such as the web server or a group of collaborative routers. Crowds does not rely on any encryption techniques, and the communication among Crowds members is open. Approaching anonymity through blending a user into a collection of users, Crowds hides a user's actions within the actions of the group [Reiter98]. More particularly, any request by a member of the crowd is either submitted to the server in question or forwarded to another member of the crowd, and the decision is randomly made by a software process running on Crowds users' computers. If the request is forwarded to another member, the same selection procedure takes place until the request reaches its intended destination. The reply from the server relays back to the original sender through the same Crowds members in the reverse order. As long as the crowd is large enough, responders, eavesdroppers, and other Crowds members never learn which particular Crowds member initiated the request despite of the openness of the Crowds member's identities. The Crowd members gain anonymity at the cost of bandwidth in forwarding other members'

communication. A new protocol called Hordes [Shields00] explored multicast routing technology to reduce the performance overhead inherent in re-routing systems such as Crowds.

Summary

Most of the strong anonymity networks, e.g., Onion Routing, require large infrastructure support, and costs associated with operating and maintaining the high- performance and availability networks are very expensive for commercial companies (unlike anonymous remailers run by volunteers). In order for end users to gain information privacy, the infrastructure of anonymity networks should aggregate a large number of users into the anonymity group. Failing to attract enough paying customers to balance the overheads of running the network resulted the downfall of many commercial ventures such as Freedom Network [Radialpoint Inc.06] and SafeWeb [Symantec Corp.07]. Moreover, some anonymity systems, e.g., PipeNet [Dai07], failed to trade-off and compromise privacy with other system properties such as usability and performance, and the effect of that is disappointing end user adoption and unsuccessful deployment.

Due to the deployment failure of strong anonymity technologies, the remaining anonymity tools can only archive relatively weaker privacy protection. From the experience of these failures, people have increasingly realised that privacy is not just a pure technical problem, but a complex socio-technical system. Over the last decade, we witnessed *"an increased use of combinations of social and technological constructs"* to preserve information privacy, and recognised that *"the desired end result (of privacy) is not in fact the technological issue of keeping information hidden, but rather the social goal of improving our lives* [Goldberg02]. In our work, we are not trying to design bullet-proof technological mechanisms for information privacy, but instead we promote end users' awareness and control to allow them selectively to disclose their personal information to achieve their social goals, e.g., fulfilling some useful tasks.

2.5.3 Transparency and Awareness

Transparency has been one of the fundamental principles for information privacy in the legal frameworks such as the FIP principles and EU's Directive 95/46/EC on personal data protection (section 2.4). Transparency of privacy practices means that informa-

tion collectors should make the data subjects aware of what information is collected and how it is used. Transparency of privacy practices is not new on the Internet, as most companies and governments have already published online their *privacy policies* in natural language. But those privacy policies are often difficult for users to locate, too lengthy to read, too abstruse to understand, and change frequently without notice [W3C03a, Jensen04]. Technical mechanisms for privacy transparency are focused on translating the lengthy privacy policy document into a machine-readable format and employing special software to automate the process of evaluating the privacy policies on behalf of the users. The most noticeable work on privacy transparency is the Platform for Privacy Preferences Project (P3P) [Cranor06].

Platform for Privacy Preferences Project (P3P)

In 1997, the World Wide Web Consortium (W3C) launched the *Platform for Privacy Preferences Project (P3P)* [Cranor06], in order to make Internet websites' privacy practices transparent and empower users more control over their online privacy. P3P takes the same philosophy as a previous effort of W3C, the *Platform for Internet Content Selection (PICS)* [W3C03b], which associates Internet content with metadata called labels to facilitate the rating and filtering services of content. Specifying the syntax and vocabulary for website's privacy policies, P3P standard provides a way of describing privacy policies for websites in a machine and human-readable XML format, which enables service providers to express their privacy practices regarding the collection, use, and distribution of personal information gathered from the user.

Although most companies and websites have already published their privacy policies written in natural language, a study by Harris Interactive in 2001 showed that only 3 percent of online shoppers thoroughly review websites' privacy policies on a regular basis [Saliba01]. The study also showed that 63 percent of the shoppers simply ignore or just briefly skim the privacy policies, and the major reasons for doing that include *"a lack of time and a high level of difficulty in understanding the privacy policies"* [Saliba01]. To change this situation, P3P allows people to delegate the task of reading privacy policies to a software component, called a *user privacy agent*, which is intended to automate the process of privacy management. More particularly, Internet websites announce their privacy policies by displaying them on a well-known place on their websites. The user agents automatically retrieve and interpret them, and compare

them to user's pre-specified privacy preferences to decide on whether to accept or reject the services. User agents can be built into a web browser, plug-ins, or other software.

The P3P specification [W3C02] includes a standard vocabulary for describing a web site's data management practices and a set of base data elements that the web sites can refer to in their P3P privacy policies. Here is an abbreviated example of a web site's P3P policy:

```
01:     <POLICIES xmlns="http://www.w3.org/2002/01/P3Pv1">
02:       <POLICY name="Browsers"
03:           discuri="http://www.catalog.example.com/Browsing.html"
04:           xml:lang="en">
05:         <ENTITY>...</ENTITY>
06:         <ACCESS><nonident/></ACCESS>
07:         <DISPUTES-GROUP>...</DISPUTES-GROUP>
08:         <STATEMENT>
09:           <PURPOSE><admin/><develop/></PURPOSE>
10:           <RECIPIENT><ours/></RECIPIENT>
11:           <RETENTION><stated-purpose/></RETENTION>
12:           <DATA-GROUP>
13:             <DATA ref="#dynamic.clickstream"/>
14:             <DATA ref="#dynamic.http"/>
15:           </DATA-GROUP>
16:         </STATEMENT>
17:       </POLICY>
18:     </POLICIES>
```

The detailed information of the data collector is described in the <ENTITY> element (line 5). <DATA-GROUP> and <DATA> element describes what data is being collected (line 12-15). This policy also describes for whom the data is being collected (<RECIPIENT> element, line 10), for what purpose (<PURPOSE> element, line 9), and for how long (<RETENTION> element, line 11).

The *P3P Guiding Principles* [W3C98] are greatly influenced by the FIP principles, and the above elements reflect their compliance with the essential parts of the principles (section 2.4.1), e.g., the Collection Limitation, Purpose Specification, Use Limitation, and Openness principles. Using a complementary policy specification language, i.e., *A P3P Preference Exchange Language (APPEL)* [Langheinrich02a], users can express their personal preferences regarding the distribution of private information in a set of preference rules (called a ruleset). These rulesets can then be used by the user's privacy management programs, e.g., user privacy agent, to make automated or semi-automated decisions regarding the acceptability of the P3P policy. Although mainly designed for

the domain of Internet websites, the P3P specification allows for the definition of new data elements and data sets by creating data schemas [W3C02] and provides an <EXTENSION> element [W3C02] to allow for the syntax and semantics to be extended.

Summary

It is worth noting that P3P only provides a technical mechanism by which services and their use of personal information are described. P3P does not provide mechanisms by which policies are enforced, nor can policies be used to verify or prove that the services accurately reflect the stated policy. P3P should be regarded as a complementary mechanism to legislative and self-regulatory programs to protect personal information against abuses by unscrupulous companies. Like P3P, we are not trying to replace social and legal privacy regulatory frameworks with pure technical mechanisms, but instead we aim to design technical solutions that operate within those frameworks to assist users to achieve better information privacy. Langheinrich [Langheinrich02b] extended the P3P vocabulary to accommodate the special properties in UbiComp environment and proposed the *Privacy Awareness System (pawS)* to increase privacy awareness for UbiComp systems (see section 2.5.5). Work in privacy transparency and awareness gives another piece of evidence that demonstrated the trend of combining social and technical constructs to approach the information privacy issue.

2.5.4 Privacy Enforcement

The openness of privacy policies, no matter written in natural language or formalised by machine-readable language such as P3P, creates incentives for information collectors and processors to keep their promises on handling personal data, because violating the published privacy policy might incur social or legal penalties. However, without technological mechanisms to support the compliance of privacy practices with policies, it becomes easy to break the privacy promises either intentionally or inadvertently, and it remains hard to detect misuse of personal information.

Recent privacy enforcement technology has concentrated on enterprise environments, mainly to assist them in managing collected personal data according to their stated privacy policies in an auditable way. The major reason for this trend is that companies have increasingly realised the importance of preserving customers' privacy in

establishing long-term customer relationships. One possible way of enforcing information privacy in an open environment, e.g., Internet, is to use so-called *Digital Rights Management (DRM)* [Open Sources07b] technologies. Working upon its supporting infrastructure called *Trusted Computing* [Anderson03a], DRM mechanisms can not only allow creators of information to have the full control of its use and distribution, but also monitor and report to the creators on the activities of individual users. In this subsection, we introduce Enterprise Privacy Technologies and DRM, and discuss their implication on our work.

Enterprise Privacy Technologies

IBM's research on *Enterprise Privacy Technologies* [IBM Corp.06] aims to assist enterprises to manage and enforce their privacy practices throughout their whole IT infrastructure while maximizing the legitimate use of collected personal information. The Enterprise Privacy Technologies consist of three main elements: (1) a methodology for enterprise to design privacy-friendly business processes, privacy-enabling security technology, and enterprise privacy management; (2) a machine-enforceable formal language for expressing enterprise privacy policies; and (3) an architecture for enforcing those privacy policies inside an enterprise environment. Due to their relevance to our work, we focus on describing the latter two elements in detail.

The Platform for Enterprise Privacy Practices (E-P3P) [Karjoth02] specifies a fine-grain policy language that facilitates formalisation and enforcement of enterprise internal privacy practices. The language was later renamed to *Enterprise Privacy Authorization Language (EPAL)* [Ashley03] as submitted to W3C for standardisation. EPAL focuses on the privacy authorization scheme specifying how collected data should be used, while ignoring enterprise-dependent deployment details such as data model and user authentication. In general, a typical privacy policy in EPAL consists of a number of authorisation rules defined in <rule> element, which normally contains the following six elements: <user-category>, <action>, <data-category>, <purpose>, <condition>, and <obligation>. While the first four elements are familiar and have their counterparts in P3P, *conditions and obligations* are unique to EPAL and facilitate the enforcement of the policies within an enterprise. Conditions are Boolean expressions that evaluate when an authorisation rule can be applied. The evaluation might require the context of the request, e.g., some data can be used for marketing purposes only if the person is an

adult and has given explicit consent. The list of context attributes can be defined using <container> element, e.g., variables such as 'age', 'consentToMarketing', etc. After performing operations on personal data, an enterprise is often obliged to take additional actions, e.g., customer's financial data must be deleted within 30 days from the date of the transaction. In EPAL, such consequential actions of certain operations are called obligations, which are returned after certain privacy rule is processed.

The design of the privacy enforcement architecture [Karjoth02, Ashley02] follows the so-called *sticky policy paradigm* that requires privacy policies to be associated with all data collected by the enterprise. In this paradigm, the privacy policy sticks to the data throughout its whole lifecycle, and is used to decide whether certain operations on the data are allowed. The privacy enforcement architecture consists of a *policy evaluation engine*, an *obligation engine*, a number of privacy-aware *resource monitors*, and a resource-independent *privacy management system*. Once a running task of a legacy application requests access to certain fields of collected data, a resource monitor captures the request and forwards it to the resource-independent privacy management system for authorisation. After receiving the authorization query, the privacy management system identifies the data field to be accessed, and translates the task onto a privacy-relevant operation on the data field and a purpose. The policy evaluation engine decides whether certain operation for certain purpose is allowed on the given personal identifiable information types by evaluating the privacy policy together with the context of the request, e.g., the data subject's choices. The policy evaluation engine returns the decision together with any mandated obligations to the privacy management system, which relays the decision back to the resource monitor. If obligations were returned, the privacy management system maps them as the tasks of the application and sent them to the obligation engine. The obligation engine records all pending obligations and triggers them based on values obtained from the *dynamic attribute service*. The resource monitor performs or denies tasks based on the authorisation decisions from the privacy management system, and could send usage logs to an *audit record module*. When an obligation reaches its ready-to-run condition, the obligation engine removes it and sends it to the resource monitor for execution.

DRM and Trusted Computing

The term Digital Rights Management (DRM) was coined by the digital media industry, to refer to a range of technical methods that *"describe, identify, trade, protect, monitor and track all forms of rights usages over both tangible and intangible assets including management of rights holders relationships."* [Iannella01] First generation of DRM imposes direct controls on copying and distribution of the digital media content. An example of first generation DRM is the *Content Scrambling System (CSS)* that employed proprietary 40-bit stream cipher algorithm to prevent users from copying movies on DVD. Second generation DRM incorporated with the capability of reporting back to the content owner on activities of individual users [Cohen03], e.g., attempts to make unauthorised copies. *Digital Watermarking* mechanisms insert hidden copyright notices or other verification messages into digital media file, which provides a means to track an unauthorised copy of the file to the original owner. Most existing Digital Watermarking techniques employed a spread spectrum approach that inserts a pseudo-noise signal with a small amplitude into the digital media file (directly onto itself or onto its frequency domain) [Ku04]. As an extension of MAC 2.5.1, DRM system grants access rights to users by strictly following the security policies written in digital rights expression languages, e.g., Open Digital Rights Language (ODRL), Extensible Rights Markup Language (XrML).

Present DRM mechanisms largely rely on *security by obscurity*, which is against Kerckhoffs' principle and vulnerable to attacks [Anderson06]. Moreover, effective DRM controls have to be enforced on temper-resistant hardware to prevent hardware-level attacks, e.g., hardware-level copying. The so-called Trusted Computing provides such a computing platform *"on which the users can't tamper with the application software, and where these applications can communicate securely with their authors and with each other"* [Anderson03a]. Technically, each Trusted Computing PC has a *Fritz chip* [Anderson03a], a smartcard chip soldered onto the motherboard, which monitors PC's hardware and software states during boot process. If the PC boots into the approved state, the Fritz chip transfers a cryptographic key to the security kernel of the Operating System (OS) that is required to decrypt Trusted Computing applications and data. Moreover, the security kernel in OS works together with the *curtained memory* feature in CPU (e.g., LaGrande Technology for Intel CPUs, TrustZone for ARM processors) to prevent applications to read or write each other's memory [Anderson03b].

Proponents of DRM and Trusted Computing argue that *"creators of digital works should have the power to control the distribution or replication of copyright materials, and to assign limited control over such copies"* [Open Sources07b]. They believe that the technologies of DRM and Trusted Computing are mature enough to be widely deployed and adopted. DRM and Trusted Computing could provide useful security features for controlling digital information within corporate and governmental organisations. We have already noticed successful deployments of those technologies by organisations such as British Library.

Opponents of the DRM and Trusted Computing, including many organisations and security experts, argue that DRM affects users' fair-use rights and Trusted Computing can support remote censorship [Anderson03b]. According to US Copyright Act, the copyright owner does not have the exclusive right to control all uses of a copyrighted work or the right to conduct surveillance of the users [Cohen03]. Recently, British Library provided evidence to the UK Parliament showing that DRM prevents them from exercising their fair-use rights, e.g., long-term access and preservation [Oates06].

In October 2005, Mark Russinovich [Russinovich05] discovered that a Sony-BMG music CD placed a *rootkit* on his Windows PC. Further investigation by independent researchers on Sony-BMG CDs confirmed that two different pieces of DRM software (XCP from British company First4Internet and MediaMax from US company SunnComm) were both *spyware*, which is installed without the user's informed consent, is very difficult to uninstall, and transmits user's activities without notice or consent [Felten06]. Felten and Halderman argued that it is not a coincidence that two rival software companies adopted the same spyware tactics for their DRM systems. DRM system designers faced two technical challenges: (1) to get the software installed even though the user does not want it, and (2) to prevent it from being uninstalled even though the user wants it removed [Felten06]. It is a non-trivial technical problem to protect the rights of the data owner while also respecting the rights of the data user [Open Sources07b].

An important issue with DRM is that we do not have any technical measures to prevent it from abuse, and greedy publishers can place arbitrary restrictions on the use of digital content. A more subtle implication of DRM and Trusted Computing is that they can cause *digital lockdown* and affect free competition in the market economy. For example, software suppliers can make it hard and costly for consumers to switch to their

competitors' products. Anderson [Anderson03a] noticed that *"the fundamental issue is that whoever controls the TC infrastructure will acquire a huge amount of power"*, and *"there are many ways that this power can be abused."*

Summary

Enterprise Privacy Technologies work within closed environments, where clear privacy practice policies can be established and implemented into technical mechanisms. In open and heterogeneous environments such as the Internet, it is unlikely to be feasible to establish dominant privacy policies for all data collectors and users. Therefore, it is a non-trivial technical problem to achieve privacy enforcement in open environment. The DRM and Trusted Computing have the potential to enforce information privacy in open environment, but they are not mature enough to well balance the rights of data owner and the data user [Open Sources07b]. Moreover, wide deployment of DRM and Trusted Computing has profound side-effects on the society, such as affecting the legitimate users' fair-use rights, promoting remote censorship, causing digital lockdown, etc. Our work does not focus on the information collectors' and users' side to assist them in enforcing their privacy promises on processing collected information. Instead, we aim to empower data subjects to make privacy-related decisions before they disclose their personal information.

2.5.5 System Support for Privacy

In this section, we review the technical approaches that have been applied to address system support for information privacy. Those approaches range from design framework, system architecture, supporting platform, to user interfaces. By critically analysing these technical mechanisms, we argue that the existing *static-policy* approach is not sufficient for privacy management. We investigate the ultimate goal for privacy management, identify the privacy management as a dynamic process and motivate the need for adaptive privacy management in dynamic networked environments.

Privacy Support in RAVE

One of the earliest system support for privacy was concentrated on so-called *media spaces* [Harrison88]. A media space involves networked audio and video equipments

to support distributed collaboration work. In the *Ravenscroft Audio Video Environment (RAVE)* project [Bellotti93] at Xerox EuroPARC, cameras, monitors, microphones and speakers were deployed in ordinary offices as well as some of the public spaces, in order to promote communication and collaboration between people. Typical applications in RAVE included *glance* (a one-way video-only connection lasting for a few seconds), *v-phone call* (a traditional phone-like full duplex connection with both audio and video), and *office share* (a background v-phone connection lasting for a long period of time). Whilst the media space technology facilitated communication and collaboration between people, it was found to cause *disembodiment* from the context into and from which one projects information and *dissociation* from one's actions [Bellotti93]. Disembodiment and dissociation break down a variety of behavioural and social norms and practices, which leads to many unintentional invasions of privacy.

Bellotti and Sellen [Bellotti93] emphasised the importance of *control and feedback* in designing for privacy in RAVE. They defined control as *"empowering people to stipulate what information they project and who can get hold of it"*, and feedback as *"informing people when and what information about them is being captured and to whom the information is being made available"*. From the experience in designing and deploying RAVE in work environments, Bellotti and Sellen developed a conceptual design framework that aimed to incorporate appropriate control and feedback mechanisms into the following four aspects:

- Capture: What personal information is being collected (e.g., audio, video, or identity)?
- Construction: What happens to the captured personal information (e.g., is it encrypted or where is it stored)?
- Accessibility: Which people and what software have access to this information (e.g., is it available to a certain user group, what software process can use it)?
- Purpose: How will the personal information be used (e.g., what is the intention of using this information)?

They applied the design framework to RAVE and demonstrated its effectiveness on one significant problem in RAVE — the video connection from a public reading and meeting area at EuroPARC. Their proposed solutions included: a mannequin holding

the video camera to provide unobtrusive and meaningful feedback when capture was occurring; a viewer display showing a list of names and pictures to indicate who has access to the video. They provided no satisfactory solutions for construction and purpose feedback, and only advocated non-technical control mechanisms, e.g., moving off camera, covering the camera, or self-adjusting behaviours. Emphasising privacy from the user-interface design perspective, Bellotti and Sellen concluded eleven criteria for systematic evaluation of privacy solutions for media spaces, and more generally for UbiComp environments, including trustworthiness, appropriate timing, perceptibility, unobtrusiveness, minimal intrusiveness, fail-safety, flexibility, low effort, meaningfulness, learnability, and low cost.

Privacy support in RAVE emphasised the importance of feedback and control at user interface level for helping the users to maintain their privacy. However, the high-level design framework is very abstract and does not address detailed technical problems such as when and how to provide feedback and control. The framework does not provide a procedure that designers could follow from requirement analysis to technical implementation, and we have seen relatively little adoption of the design framework. We acknowledge the importance of the feedback and control, and aim to employ them as fundamental underlying technical mechanisms for our platform.

Privacy Aware System for UbiComp (pawS)

While anonymity and pseudonymity techniques can be applied to protect people's virtual identities such as email and IP addresses on the Internet, they are less useful for UbiComp environments where a large number of sensors and computing devices constantly monitor people's real-world presences and collect personal data such as location and activities. Langheinrich claimed that people's real-world presences cannot be completely hidden and perfectly anonymized unless people want to completely abandon their social lives. Believing that perfect privacy is not realisable by technology, Langheinrich argues that privacy management systems should increase users' awareness of privacy and promote the respect of one another's privacy. His argument was based on the following principle in democratic societies: *"to give people the ability to respect other people's safety, property, or privacy, and to rely on corresponding social norms, legal deterrence, and law enforcement to create a reasonable expectation that people will follow such rules"* [Langheinrich02b]. Instead of trying to guarantee perfect

privacy, the Privacy Awareness System (pawS) assists data collectors and processors in UbiComp environments to make explicit privacy promises and relies on social and legal mechanisms to motivate them to keep their promises.

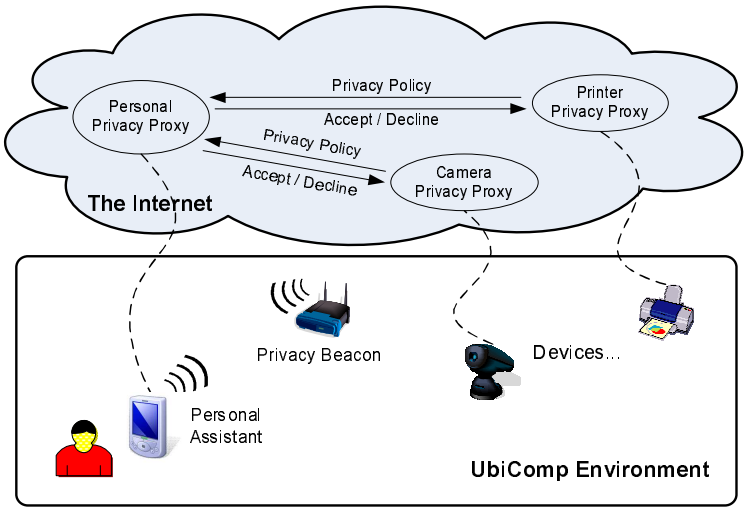


Figure 2.1: The Privacy Awareness System (pawS) [Langheinrich02b]

The design of the pawS architecture (figure 2.1) followed the principles [Langheinrich01] proposed earlier by Langheinrich for preserving privacy in UbiComp, which was in turn based on the framework of FIP principles. The pawS proposed to use a *privacy beacon* to announce the data collection and usage policies for the services in a UbiComp environment via some wireless communication channel. All the privacy-related interactions are delegated to *privacy proxies* for both users and services. Privacy proxies are constantly running services residing somewhere on the Internet. In order to use the services in the privacy-aware UbiComp environment, everyone has to carry a personal digital device, e.g., a PDA, on which runs a program known as *privacy assistant*. The privacy assistant receives the message from the privacy beacon and forwards it to the user's *personal privacy proxy*. The personal privacy proxy then contacts a *service privacy proxy* and compares the service's privacy policy against user's privacy preferences to decide whether to accept or decline use of the service. The pawS employed both P3P and APPEL in the implementation to express the service policies and user preferences respectively, and the vocabulary of the P3P policy language has been extended to accommodate specific properties for UbiComp environment, such as location.

In the pawS, the UbiComp services store the requested personal information in a *privacy awareness database (pawDB)*, together with the individual privacy policy that the data was collected under. By maintaining data with metadata governing its usage,

the database can take care of observing that the usage of the data complies with the privacy policy with respect to the lifetime, usage, and recipient of a certain piece of personal information. Data users need to submit a data usage policy in order to query any of the stored data in the database. Each database query with a reference to its usage policy is recorded in a data usage log, so that data owners are able to inspect the usage of their data through a list of recorded usage policies. The pawDB provides retention enforcement by periodically checking the collection timestamp of the data elements and removing the elements whose valid storage period has expired.

The pawS architecture focused on using policy mechanism to increase awareness of privacy and automate the privacy negotiation process for UbiComp environments. His work does not address the issue of policy generation, and it assumes the existence of online repository for users to download default policies. Complete automation of the privacy management process based on pre-defined polices gives users no chance to modify and override their previous preferences. This static-policy approach does not meet the changing desire of the users, and we argue that the privacy management requires more adaptive approach. Moreover, we believe that system should be designed to assist the users to make privacy-related decisions, instead of replacing the users.

FACES

Lederer et al developed a program called FACES [Lederer04] to facilitate end users to manage their privacy in the UbiComp environments by supporting them to specify their preferences for disclosing personal information. Influenced by sociologist Erving Goffman, the authors believed that social life is like a theatre and people perform different roles or maintain appropriate faces in relation to an audience. Therefore, they selected metaphor of "faces" to represent different disclosure preferences, and engineered a privacy manager for desktop PCs that enables users to specify their preferences prior to any disclosure of personal information. People generate their privacy preferences by specifying three elements:

- inquirer: the identity of the entity requesting personal information, and it can be person or a group of person.
- situation: the encapsulation of the contextual information of the inquiry, including location, activity, time, and nearby people.

- face: the encapsulation of the disclosure preference on the precision of the information. Users can specify the precision of the information to be disclosed at four different levels, from "Undisclosed", "Vague", "Approximate", to "Precise". Users can apply this precision of information to the following dimensions of their personal information, including identity, location, activity, and nearby people.

The formative evaluation of FACES [Lederer03b] exposed some fundamental problems with its design. User studies showed that the participants found it hard to remember the preferences they had specified before, and the participants' privacy preferences for real scenarios always differ from what they previously specified. Lederer et al argued that the separation of the privacy preference specification and the privacy management actions inhibits the users from effectively *practicing* the privacy management through the FACES interface. From their experience with FACES and based on analysis of other existing interactive systems, Lederer et al identified five pitfalls that system designers are likely to fall into when designing privacy sensitive applications. The first two pitfalls are concerned with users' understanding of the system's privacy implication:

- design should not obscure the nature and extent of a system's potential for information disclosure (i.e., "obscuring potential information flow"), and
- design should not conceal the actual information disclosure (i.e., "obscuring actual information flow").

Lederer et al believe that system designers should avoid these two pitfalls to fortify users' comprehension of system's scope, utility, and the implication of information use [Lederer04]. The remaining pitfalls affect users' intuitive actions of privacy management in different situations:

- Designs should not require excessive configuration to manage privacy, but should allow users to carry out privacy management actions as a consequence of their normal engagement of the system (i.e., "emphasizing configuration over action").
- Design should not neglect the top-level mechanism for enabling and disabling information disclosure (i.e., "lacking coarse-grained control").
- Designs should not prevent users from transferring established social practice to emerging technologies (i.e., "inhibit established practice").

In contrast with the *feedback and control* framework by Bellotti and Sellen [Bellotti93], Lederer et al argued that system designers should empower the users to maintain their privacy by enhancing their *understanding* of the privacy implications of their social-technical contexts and assist them in taking socially meaningful *actions*. They believed that technical feedback and control mechanisms are opportunities for understanding and action, and are crucial for the system designer to empower the users to maintain their privacy. This is consistent with our observation that systems should empower users to assist them in making privacy-related decision by supplying them with knowledge of the system and the ability to act on this knowledge. The lessons from FACES taught us that system should not separate the preference specification with the privacy management interactions. We believe that privacy management is a by-product of user's primary task of accessing information and services, and therefore we should minimise the user effort for privacy management by folding privacy management actions into the major task that the user is undertaking.

Houdini Framework

Telecommunication services and web-based applications are increasingly providing services and information tailored to individual customers, e.g., notifying the traveller of their departure gate via Short Message Service (SMS), or recommending additional purchases based on customer's online shopping history. To personalise themselves, these services or applications exploit an increasing amount of personal preference information, called profile data. Profiles typically contain both *static* data (e.g., address, calendar, favourite food, etc) and *dynamic* data (e.g., presence, activity, location, etc). The private nature of the profile data determined that the process of profile data sharing has to be privacy-conscious. While users are willing to share their profile data to other people or business entities, they demand flexible control on who can access which piece of information and under what circumstances. Hull et al [Hull04] argued that the existing approaches for personal data sharing were designed for conventional data management environments and did not address the issues of context awareness inherent in mobile environments, i.e., people's decisions on profile data sharing may depend on their locations, recent and current activities, etc. Identifying the issues of context awareness and privacy consciousness are intertwined in the profile data sharing, Hull et al proposed Houdini framework (figure 2.2) that aimed to facilitate developing context-aware and privacy-conscious data sharing applications [Hull03].

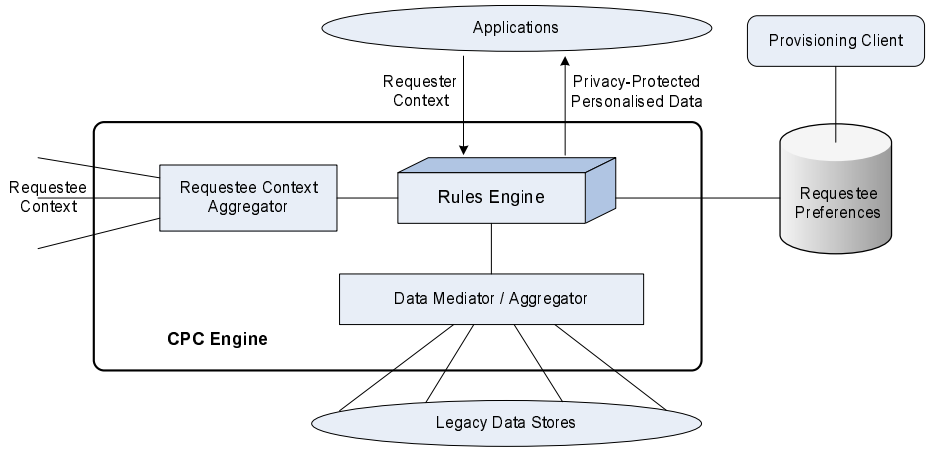


Figure 2.2: The Houdini Framework [Hull03]

The central idea of Lucent's Houdini framework [Hull03] is to put an additional layer, called *Privacy-Conscious Personalising (PCP)* engine, on top of legacy data stores, to control the access to and distribution of the profile data from different requesters. Most customisation infrastructure employs the *value-based approach*: end users provide a collection of personalised values for applications to interpret and perform customisation, but the core logic of the application (including the logic for customisation) is essentially static. Hull et al argued that the value-based approach is inflexible for providing personalisation and privacy control in context-rich mobile and ubiquitous computing environments. The Houdini framework took the *rule-based policy approach*: end users' preferences are translated into policies expressed in rulesets (a ruleset is defined as a collection as rules), and the policies embody both values and part of the application logic, e.g., logic for customisation, logic for profile data sharing. In the Houdini framework, a common rule execution engine evaluates the rulesets for different applications and determines privacy conscious profile data sharing based on four different sources of information, including requestee's static data, requestee's context (dynamic data), requester's context, and requestee's preferences on how to share their profile data. Applications receive privacy-related decisions from the rule engine, and enforce those decisions by executing operations such as blocking, filtering, or transferring.

The Houdini framework facilitates end users' self-provisioning of preferences: it enables the end users to specify their preferences using familiar web-based interfaces and automatically translate them into policies that are expressed in rulesets. Instead of mapping each entry in the web page to a separate rule, the translation process identifies the common structures in parts of the entries and creates *generic rules* for them. This

generic rule approach reduced both the size of the rulesets and the total number of rulesets. The authors developed a tailored version of a rule-based language that is strongly typed and supports forwarding chaining with acyclic rulesets, and the rule engine can make privacy conscious decisions within milliseconds, which is crucial for running near real-time services such as call forwarding and friend locating.

Hull et al identified the importance of contextual information in affecting people's decisions on releasing personal information, e.g., sharing profile data. We have the similar insight that the inherent dynamic nature of mobile environments requires adaptive approach for managing personal information. The rule-based policy approach decoupled the privacy-related decision making from the core logic of applications, and the rule execution engine made privacy-related decisions based on the rulesets translated from people's preferences. The preferences were pre-specified and did not change while user-interactions occur. Pre-defined rulesets could not meet people's changing desire for mobile services, and there is evidence from the failure of the FACES. We believe people's privacy preferences evolve over time while they interact with different services.

Information Exposure Modelling

Dragovic and Crowcroft observed that existing security and privacy mechanisms, built for static and predictable execution environments, failed to provide the flexibility to balance the information availability and privacy control for dynamic UbiComp environments. They noticed that personal information in the UbiComp environments is exposed to constantly changing set of security and privacy threats throughout its lifetime, and they argued that continuous and adaptive approaches are required to maximise information availability to legitimate users while limiting the threats of the information exposure to the surrounding environment [Dragovic05a]. Their approach was largely motivated by their observation of human behaviours: people often adjust the form and characteristics of information to the perceived security and privacy risk in the environment. For example, people tend to lower the volume of their voice or change topics when they realise their private conversation could be overheard. Inspired by their observation, Dragovic and Crowcroft aimed to model security and privacy threats to personal information through sets of contextual attributes and mitigate the risks by manipulating the form and characteristics of the information while maximising legitimate access to the information.

Borrowed the idea of container from spatial reasoning algebra [Egenhofer99], Dragovic and Crowcroft propose to use the notion of a container to define the containment relationship between information and its direct surrounding environment. In their paper [Dragovic05a], a container is defined as the *"physical or virtual enclosure in which a piece of information or a lower level container exists"*. Therefore, the concept of container encompasses hardware such as storage devices or physical displays, as well as software such as files or communication links. The authors can model the real world using a hierarchy of containers, called a *containment tree*. Their work focuses on the minimising information exposure threats, which they defined as the risk of the unintentional information leakage into the environment as a side-effect of the information management procedures in a particular context. To quantify information exposure threats, Dragovic and Crowcroft proposed the *Levels of Exposure (LoE)* model [Dragovic05b] that considers three elements: the context sensing uncertainty, the perceived likelihood of threat occurrence, and the threat effect.

The authors proposed to automatically reason about the information exposure threats using the LoE model and mitigate them by proactive actions on manipulating the properties of container or operating directly on the information itself. *Container manipulation* aimed to lower exposure threat of the information within a container by

- modifying the properties of the existing container, e.g., resizing a GUI windows,
- creating a new container along the path of the containment tree, e.g., file encryption, regarded as "enclosing" a file within a cryptographic container, or
- migrating to another container with less exposure threat, e.g., migration of information from a public display to personal mobile phone display.

Information manipulation does not change the exposure threat of the information, but aimed to make the information more tolerable to the experienced exposure by reducing the quality and quantity of the information, e.g., releasing more coarse-grained location information, degrading JPEG image resolution, or fully eliminating sensitive pieces of information. Dragovic and Crowcroft proposed to automatically reason about appropriate action using the *Information Utility Measure (IUM)* that combines four factors (i.e., information content, locality of information, information accessibility, and user perceived Quality of Service) to rank the available proactive actions. They [Dragovic05b]

applied the theoretical model to the design of a sub-file granularity data repository system, and the implementation is still in its early stage.

Dragovic and Crowcroft's vision of dynamic approach is consistent with our insight: existing static approaches for personal information management limit the availability of information and usage of services, and therefore it requires more adaptive approach for personal information management in the dynamic computing environments. The notion of container is useful to model the world and the privacy risks, and the LoE gives the developers the flexibility to define application-specific functions to compute the exposure. We believe that it is impractical to fully automate the reasoning about proactive actions based on the LoE and IUM, because some of the key elements are not actually computable, e.g., user perceived QoS. Moreover, their approach over-emphasised the completely automatic adaptation while neglecting the important role played by the end users. Users do not have any chance to intervene the process of adaptation, and it might raise undesirable side-effects when the automatic reasoning goes wrong. We borrow their notion of container for modelling privacy risks, and propose adaptive approach to assist users in find the desired level of openness in disclosing personal information.

Confab Toolkit

Hong and Landay, the designers of the *Confab Toolkit* [Hong04a], argued that privacy involves many social and organisational issues that can not be controlled by technological means alone. Influenced by Lawrence Lessig's philosophy of privacy, they believed that privacy has to be achieved through a combination of technology, legislation, corporation policy, and social norms [Lessig98, Lessig99]. Therefore, Hong and Landay aimed to *"empower people with choice and informed consent, so that they can share the right information, with the right people and services, in the right situations"* [Hong04a] in context-aware computing and UbiComp environments. Drawing on their previous work on *Approximate Information Flow (AIF)* [Jiang02a, Jiang02b], Hong and Landay advocate a decentralised architecture that captures, stores, and processes end users' personal information on their personal device as much as possible.

In the Confab architecture (figure 2.3), *InfoSpaces*, network-addressable logical storage units, manage contextual information of entities, e.g., people, places, devices, or services. *Context Tuples* are basic storage units in an InfoSpace, and can be used to describe different types of context data, e.g., relatively static context data such as name or

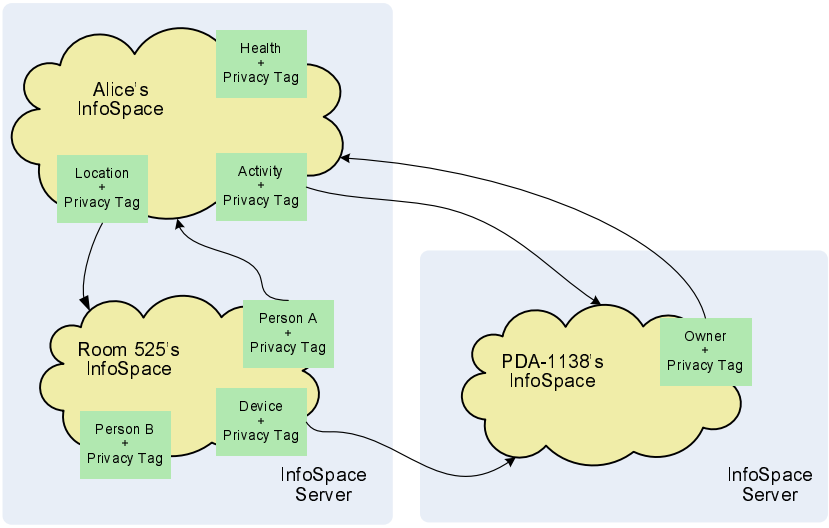


Figure 2.3: InfoSpace Model for Confab Toolkit [Hong04a]

age, dynamic context data such as activity or location. Each context tuple has a *Privacy Tag* that describes how the context information inside the tuple should be used in order to assist in enforcing the usage of the context information. Unlike P3P, privacy tag was tailored to the exchange of dynamic contextual information, and a typical privacy tag can include elements such as *TimeToLive* (to specify how long the data should be kept before being destroyed), *MaxNumSightings* (to specify maximum number of previous values that should be kept for a context tuple), *Notify* (to specify the address for sending notification of second use to), and *GarbageCollect* (to specify hints on when the data should be deleted).

Confab's programming model offers application developers three different pieces of functionality to control the flow of personal information between InfoSpaces, i.e., *operators, service descriptions, and active properties*. For each InfoSpace, in-operators are performed on incoming tuples to enforce access control policies and make sure the tuples can be added to the InfoSpace, and out-operators are performed on outgoing tuples to enforce privacy, e.g., blocking outgoing tuples, adding privacy tags, notifying end users, etc. In addition, on-operators are defined to perform certain tasks periodically, such as garbage collection of obsolete data and generating privacy reports to the owner of an InfoSpace. Confab's service description allows applications to specify different levels of services, each of which can describe different requirements of handling personal information, e.g., what personal information is needed at what precision and frequency. When an application requests some personal information, it transfers the service description to the InfoSpace of the requested person. If the InfoSpace has not

seen the service description before, it displays a GUI to allow the end users to choose the level of service they want. Active properties periodically query the context states of the entities and maintain the last known values.

Using Confab toolkit, the authors developed a few privacy aware applications, including a location-enhanced instant messenger called Lemming [Hong04a], which allows users to request each other's current location. Due to the private nature of location information, Lemming provides the users flexible control on releasing their current location while receiving a request, i.e., a GUI that contains options such as "never allow", "just this one", "ignore for now", or "allow if..." (to set more complex location disclosure conditions).

Confab toolkit aimed to empower end users to manage their privacy, which is consistent to our objectives. Confab gives user options when disclosing personal information and emphasizes the control at the user interface level, but it did not take into account the dynamic changes of the underlying system because their options are static and the same for every situation. Although the GUI in Lemming allows users to specify complex disclosure conditions, their user studies showed that no one actually used it and everyone chose "just for now" [Hong05]. We believe that it is crucial to constantly monitor or observe the changes of privacy constraints of the underlying system and give the user options that are dynamic and suitable for the changes. Moreover, we do not think that keeping personal information on owner's machine as much as possible would result better privacy, and we do not see any pragmatic evidence of that from their work.

2.6 The Need for Adaptive Privacy Management

In this chapter, we have reviewed the history of privacy and identified information privacy as the focus of our research. The concept of information privacy is not static but evolves with the advance and availability of new technologies. Research on information privacy from the social perspective has shown us that personal information privacy is not purely a technical problem but a complex social-technical system [Anderson04]. The legal frameworks we surveyed (e.g., FIP principles) provide a comprehensive list of principles for maintaining information privacy, and as far as the author was aware of, none of the existing technical solution meets all requirements stated in the principles. Moreover, we believe that personal information privacy cannot be achieved using tech-

nology alone, a hypothesis supported by a prominent cyber-law scholar, Stanford Law Professor Lawrence Lessig, as he concluded that information privacy has to be achieved through a combination of technologies, legislations, social norms, and market forces [Lessig98, Lessig99].

With the background knowledge of information privacy, we have reviewed technical mechanisms for achieving information privacy. Sophisticated anonymity techniques on the Internet were not found to succeed beyond research prototypes, and the anonymity tools that did remain provide us relatively weaker privacy protection. Privacy transparency mechanisms (e.g., P3P) do not provide any technical measures to verify whether privacy practices are consistent with the publicised privacy policies, and they rely on social norms and legal frameworks to help people to respect each other's privacy. Over the last decade, we saw very few technical tools to achieve stronger information privacy. Instead, we witnessed increased use of combinations of social and technological constructs for achieving information privacy [Goldberg02]. Goldberg concluded that *"these combinations recognize the fact that the desired end result (of information privacy management) is not in fact the technological issue of keeping information hidden, but rather the social goal of improving our lives"* [Goldberg02]. We assume the existence of the social and legal frameworks, and we aim to propose technical solutions that work within those frameworks instead of replacing them.

From the experience of online file sharing systems, researchers found that it is a non-trivial technical problem to enforce information privacy, especially to prevent personal information from secondary use (e.g. sharing or exploitation) [Goldberg02]. In closed environments such as companies or government departments, technical solutions like Enterprise Privacy Technologies are applicable, but there is no satisfactory technical measure to enforce information privacy in open and heterogeneous environments. DRM and Trusted Computing have the potential to enforce information privacy in an open environment, but they are not mature enough and have thus far failed to balance the rights of the data owner and the rights of the data user [Open Sources07b]. Moreover, wide deployment of DRM and Trusted Computing has profound side-effects on our society [Anderson03a], e.g., affecting the legitimate users' fair-use rights, promoting remote censorship, causing digital lockdown, etc. *Our work does not focus on the information collectors' and information users' side, to assist them in enforcing their privacy promises on processing collected information, e.g., preventing from secondary use. Instead, we aim to empower data subjects to make right privacy-related decisions*

before they disclose their personal information.

In providing system support for information privacy, a number of projects have taken the static policy approach that pre-specifies users' information disclosure preferences in privacy policies and utilises them for user-transparent privacy negotiation with networked services and applications. Although privacy policy languages are useful for describing users' privacy preferences, the static policy approach presents a number of problems.

- First, the vocabulary and structure of privacy preferences have been found to be too complex for normal users to incorporate and use [Hochheiser02]. In a system such as pawS, where it is assumed that users will download default privacy preferences from an online repository [Langheinrich02b], we would argue that the difficulty of the preference language prevents typical users from modifying these preferences to better match their specific needs.

- Second, even if users know how to modify their privacy preferences, researchers have found that users do not expend any extra effort to do this and simply accept the default ones instead [Palen99, Mackay91b, Hong05]. For example, in Confab users were offered a GUI to specify complex location information disclosure conditions after they receive a location request, but everyone in the user studies ignored it and chose the default option (i.e., "just for now") [Hong05].

- Third, with the FACES system, researchers found that even if people did take effort to pre-specify their privacy preferences (actually, people were given a task of setting their privacy preferences using FACES), they may find difficulties in applying them due to the separation of the privacy preference specification and the privacy management actions [Lederer04]. User studies of FACES revealed that people found it hard to remember the preferences they had specified, and the privacy preferences for real scenarios were found to differ from what they previously specified in the majority of cases[Lederer04].

In summary, the static and inflexible policy approach does not meet end users' changing requirements for their information privacy in the networked computing environments, where systems have become increasingly dynamic, complex, and unpredictable. A number of projects (e.g., Houdini, FACES) identified the importance of highly dynamic contextual information in affecting people's decisions on disclosing

personal information, and allowed users to specify their privacy preferences using contexts (e.g., location, activity). *We believe that privacy-related decisions are highly situational, and we argue it is impossible to predict all the situations and impractical to pre-specify them into privacy preferences.* Dragovic and Crowcroft recognised that personal information in the networked computing environments (e.g., UbiComp) is exposed to constantly changing set of security and privacy threats throughout its lifetime, and proposed to automatically reason about the threats model and mitigate them using proactive actions. In studying people's privacy preferences for e-Commerce, researchers found that most end users do not want tools that automatically transfer their personal information to Web sites [Ackerman99]. Moreover, we argue that the automatic mitigation of privacy risks is impractical because some situational factors (e.g., information sensitivity, mood) are highly subjective, and there is no existing technical means to reliably determine them. *We regard the privacy management as a personal decision making process, and the system should be designed to assist users in making sensible decisions instead of replacing users by automated processes.*

The failure of the static policy approach in highly dynamic environments motivated us to investigate the ultimate goal of privacy management. As we concluded earlier, increased use of combinations of social and technological constructs for privacy demonstrated that the desired end result of information privacy management is not about keeping personal information hidden but rather selectively disclosing personal information to fulfil the social goal of improving lives [Goldberg02]. Theoretically, social psychologist Irwin Altman conceptualised privacy as *"selective control of access to the self"*, and claimed the goal of privacy management is *"to adjust and optimise human behaviours for specific social situation to achieve the desired state along the spectrum of openness and closedness"* [Altman77]. To unpack privacy in networked environments, Palen and Dourish adapted Altman's theory and argued that *"privacy management is not about setting rules and enforce them; rather, it is the continual management of boundaries between different spheres of actions and degrees of disclosure within those spheres"* [Palen03]. While building the Crowds system to support anonymous web applications, Reiter and Rubin defined six degrees of privacy along the spectrum of openness and closedness (figure 2.4).

Based on the above investigations, we argue that better privacy is not about hiding as much personal information as possible, but *enabling personal information disclosure at a level of openness that is as close as to a user's desired level to assist him/her in accom-*

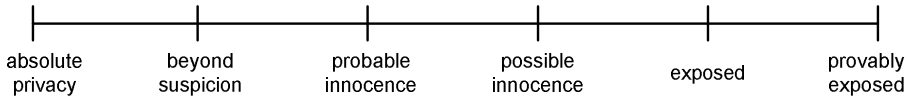


Figure 2.4: Degree of Privacy defined by Reiter and Rubin [Reiter98]

plishing useful tasks. We have the following key observation of privacy management in networked environments:

> *In accomplishing a useful task under specific circumstance, people have a desired level of openness on disclosing their personal information. More importantly, this desired level of openness varies with the changes of the circumstance, e.g., the recipient of information, the sensitivity of information, the time of disclosure, the precision of the information, etc. For example, people might disclose their location information to their colleagues when they are at work but not out of working hours; people might disclose presence information to family members as "out-for-lunch" but to others as "unavailable". Following Palen and Dourish's observation on information privacy [Palen03], we argue that no set of pre-specified control rules can meet an user's changing requirements for privacy in dynamic environments and hence achieve better privacy for the user.*

More specifically, pre-defined privacy preferences as described in policies only set levels of openness for a limited number of circumstances and cannot accommodate changes in the environments, which results in either too much or too little privacy than what people had desired.

> *We believe that users' privacy-related decisions are highly situational. Moreover, people's privacy preferences are not static but evolve over time with their accumulated experiences and increased understanding of the services and applications. Therefore, we argue that the privacy management requires an adaptive approach that optimises selective disclosure of personal information under different circumstances in dynamic networked environments, in order to for the end users to gain the benefits of accessing services and using applications at their desired levels of openness.*

We propose adaptive privacy management as the process that a user and/or a system continuously adjusts the system behaviour of disclosing personal information according

to the user's changing desire for openness under different circumstances in dynamic networked environments.

2.7 Summary

This chapter offered an overview of the privacy issue from four different perspectives: historical, social, legal, and technological. Based on the critical review of the existing static policy approach to information privacy, we identified the need for adaptive privacy management that optimises selective disclosure of personal information under different circumstances in the dynamic networked environments. The adaptive privacy management is not about hiding as much personal information as possible, but aims to enable information disclosure at a level of openness as close as to a user's desired level and to assist the user in accomplishing useful social tasks. An analysis of the importance of this finding is given in chapter 3.

CHAPTER III

Analysis

Contents

3.1 Overview

In order to support privacy management of intentional personal information sharing applications in networked computing environments, we propose adaptive privacy management; where a user and/or a system continuously adjusts the system's disclosure of personal information according to the user's changing desire for openness under different circumstances. This chapter provides an overview of design strategies for information privacy solutions, and critically analyses advantages and disadvantages of the existing technical approaches surveyed in the previous chapter. Building on this analysis, we explain the rationale for selecting specific technical mechanisms for the development of adaptive privacy management. Finally, we identify the set of requirements for supporting adaptive privacy management in personal information sharing applications.

3.2 Design Strategies for Achieving Privacy

We have previously explained how information privacy is a complex socio-technical system and has to be achieved through a combination of technologies, legislation, social norms, and market forces. In this section, we summarise strategies for designing technical mechanisms that work within the existing legal and social frameworks to achieving information privacy in networked computing environments. We provide an analysis of advantages and disadvantages of each design strategy for information privacy, then discuss our rationale for selecting a specific set of technical mechanisms to support the development of adaptive privacy management.

3.2.1 Control at Information Collection

Privacy threats prevail throughout the whole lifecycle of personal information, including at nformation collection, dissemination, primary and secondary use, and storage. The predominant way for enabling information privacy is to prevent personal information from being collected by unauthorised parties. Typical examples of of these types of control mechanisms can be found in traditional access control mechanisms deployed in mainstream file systems and static privacy policy approaches that allow users to impose restrictions on others principals (e.g., other users) on retrieving their personal information. Normally, a user (e.g., a system administrator) has to deploy and configure

such mechanisms before others can get access to the information. Grudin and Horvitz [Grudin03] refer to these types of strategy as *pessimistic* control mechanisms as they prevent unauthorised access to personal information by allowing people to specify access privileges before others can initiate operations on the information.

There are a number of reasons why control at the point of collection is the predominanat mechanism for controlling information privacy:

- Firstly, these form of control mechanism have been well-studied in computer science and engineering (section 2.5.1), and therefore the foundation for applying these mechanisms are sound and mature.
- Secondly, these kinds of mechanisms can be used for most types of application with low integration overhead when compared with other mechanisms such as anonymity [Beresford05].
- Thirdly, this type of control is the most natural for people to understand, because it is very similar to the way that people accept or deny requests for personal information during social interactions.

However, there are a number of problems of using traditional access control or static policies to protect information privacy:

1. Configuration of access control parameters (e.g., privacy policies or preferences) for information privacy can be difficult and error-prone, and ordinary users with little knowledge of control mechanisms often fail to employ such mechanisms effectively [Whitten99, Beresford05]. Based on this observation, we infer that the difficulty of effectively setting control parameters will prevent normal people from modifying pre-specified privacy control parameters in response to changes in their privacy requirements.
2. Previous research [Palen99, Hong05] has shown that people are reluctant to expend extra effort to modify their privacy preferences even if they know how to do so. One reason for this is that privacy is often a secondary goal when accomplishing the primary goal of actually using a service [Whitten99], and moreover, the process of configuring privacy preferences is often separated from primary interactions with the service [Jensen05].

3. Privacy preferences are often hidden in the system as soon as they were specified, and people normally are unaware of their effectiveness and tend to forget their existence overtime [Lederer04].

4. People often find difficulties in applying previously specified privacy preferences to real life situations, and preferences that are de-contextualised from the privacy interactions (e.g. set when users first use the system) often fail to meet their privacy requirements in such scenarios [Lederer04].

3.2.2 Anonymity and Pseudonymity

Anonymity and pseudonymity mechanisms (section 2.5.2) are designed to hide or mask private information within a larger population in order to make it difficult to resolve the identity, characteristics or significant features of the individual to whom the information belongs. Chuam introduced the concept of the 'Digital Mix' that has had a significant impact on anonymity provision in network communications. Chuam's Mix can be abstracted to the concept of the *Anonymity Set*, which later Andreas Pfitzmann and Marit Köhntopp formalised as *"the set of all possible subjects who might cause an action"* [Pfitzmann01]. By hiding personal identifiable information in an anonymity set, anonymity and pseudonymity mechanisms allow a user to remain anonymous within a group of users, i.e., a piece of information could belong to any user in the anonymity set and the set is sufficiently large as to make the exact identity difficult to resolve. Anonymity and pseudonymity mechanisms have been widely applied to Internet applications such as email communication and web browsing, and they are relatively mature technology for concealing real-world identities by making it infeasible to infer them from identities or patterns in secondary data.

Strong anonymity mechanisms are not often desirable for distributed applications. From an end user point of view, total anonymity can be undesirable as long-term relationships (such as reputation and trust) with other entities cannot be established. From an application developer point of view, a truly anonymous application is hard to engineer because information flows in one-direction from the user to the application and the application does not know who to communicate with [Beresford05]. In a survey of anonymity technologies on the Internet, Goldberg [Goldberg02] found that stronger anonymity mechanisms did not succeed beyond research prototypes and the anonymity tools that remained provide relatively weaker technical mechanisms for privacy protec-

tion.

Moreover, anonymity and pseudonymity mechanisms have been found to be less effective in mobile and emerging UbiComp environments than on the Internet, because people's real-world information such as location or presence is much more difficult to perfectly conceal [Langheinrich02b]. A case study of Active Bat system by Beresford and Stajano [Beresford03] demonstrated that static pseudonyms cannot provide sufficient protection for privacy of location information, because attackers can correlate a user's pseudonym with his real-world identity through publicly available data, e.g., university websites, phone books, etc. Beresford and Stajano further proposed to use changing pseudonyms and introduced the notion of mix zones for protecting location privacy. In a mix zone, applications do not receive users' location, and each user changes his pseudonym whenever he enters a mix zone. Therefore, applications cannot link the identities of the users entering the mix zone with those leaving it.

3.2.3 Awareness and Accountability

Existing technical mechanisms for privacy awareness (or transparency) have mainly focused on translating legal privacy frameworks into machine-readable forms that can be checked against user preferences automatically on behalf of the users. P3P (section 2.5.3) is the most noticeable work in this area, and researchers have since proposed to extend P3P to UbiComp environments [Langheinrich02b, Myles03]. Note that these mechanisms for privacy awareness only provide a technical means for services to describe how they collect and use personal information. They do not provide mechanisms by which privacy policies are enforced, nor can policies be used to verify or prove that the services accurately reflect the stated policies. Extending basic privacy awareness mechanisms, Enterprise Privacy Technologies utilise the *sticky policy paradigm* where privacy policies are associated the data collected itself. An extended content management system manages and enforces the privacy policies within the content workflows of the enterprise environment (section 2.5.4). It is unlikely to be feasible to establish dominant privacy policies for all data collectors and users in open and heterogeneous environments, and evidence has showed that it is a non-trivial technical problem to enforce privacy in such environments [Goldberg02]. We note that privacy awareness mechanisms typically rely on social and legal frameworks to detect violation of privacy policies.

Instead of preventing unauthorised collection of information, accountability mechanisms approach the information privacy issue by maintaining traces of information access and usage. In traditional social interactions, people are often aware of personal information disclosure and the recipient of the disclosure is accountable for use of the information [Bellotti97]. In computer-mediated interactions, distributed applications often automate personal information disclosure without the knowledge of the individual. By maintaining traces of personal information access and usage, a system enables users to observe personal information disclosure history and base accountability on it [Raento05]. Quoted in [Weiser91], accountability mechanisms are consistent with Jim Morris' vision of building computer systems *"to have the same privacy safeguards as the real world, but no more, so that the ethical conventions will apply regardless of setting"* . Grudin and Horvitz [Grudin03] proposed the notion of *optimistic* control mechanisms, which grant everyone full access to the information by default but record activities taken on the information. Undesirable operations on the information can be detected from the access traces, and the user or/and system can revoke others' ability to initiate certain operations on the information.

3.2.4 Control at Information Use

In contrast to control at information collection, an alternative approach is to distribute the information in some protected form and delay the control until such a time as the information is used. A simple example of this mechanism is password-protected files, where a user is typically asked to input a password associated with the file in order to use it (e.g., read, write, execute, etc). Traditional encryption techniques can be employed as control mechanisms at information use, because only holder(s) of correct cryptographic key can decode the encrypted information and therefore use it. In a public key cryptographic system, a user can distribute personal information encrypted using a recipient's public key, and only the recipient with the corresponding private key can decrypt the information and use it. However, the aforementioned mechanisms do not provide any technical means to protect information from unauthorised secondary use once it has been decrypted; there is no restriction on distributing the information in its decrypted form. From the experience of online file sharing systems, researchers have found that it is a non-trivial technical problem to prevent personal information from secondary use (e.g. sharing or exploitation) [Goldberg02].

DRM and Trusted Computing have the potential to control secondary usage and therefore to enforce information privacy in open and heterogeneous environments. The first generation of DRM (CSS) imposes direct controls on copying and distribution of digital media content. Second generation DRM incorporated the capability of reporting back to the content owner on activities of individual users, e.g., attempts to make unauthorised copies. Since effective DRM controls have to be enforced on temper-resistant hardware to prevent hardware-level attacks, Trusted Computing platforms were developed to prevent users from tampering with application software using a variety of technologies such as the *Fritz chip* and *curtained memory* (section 2.5.4). However, DRM introduces as many problems as it solves:

- Firstly, existing DRM systems are closed proprietary systems that require proprietary hardware and/or software. A user has to deploy a correct DRM system before they can access the protected information, because different DRM systems (sometimes different versions of the same DRM system) cannot interoperate with each other [Ku04].
- Secondly, DRM systems prevent people from exercising their fair-use rights such as long-term access and preservation [Oates06], and DRM and Trusted Computing provide an easy means for profiling users' consumption behaviours [Russinovich05].
- Thirdly, wide deployment of DRM and Trusted Computing have been cited as digital lockdown and affecting free competition in the market economy [Anderson03a].

3.2.5 Discussion

Control at the point of information collection remains the predominant and most effective way for achieving information privacy. From the analysis in section 3.2.1, we found that existing access control mechanisms (including the static-policy approach) failed to meet end users' changing requirements for information privacy in networked computing environments, because users cannot efficiently and effectively adjust the level of openness to suite their desired level for different situations. Therefore, we claim that existing access control mechanisms are too static and inflexible for achieving information privacy in networked computing environments, and we argue that *the research challenge of employing access control mechanisms at information collection is how to empower normal people to effectively use them*. Recent attempts to address this challenge involve

introducing the metaphor of *Virtual Walls* [Kapadia07] based on physical walls to help users to specify complex privacy policies, and proposing the concept of a *Privacy Invasion Value (PIV)* [Osbakk04] and the notion of *Environmental Roles* [Covington01] to extend basic RBAC model. *In this thesis, we will use access control mechanisms as the underlying enabling technology to achieve information privacy, and propose an adaptive approach for managing information privacy to empower ordinary users to effectively use them.*

Strong anonymity mechanisms failed in real-world deployment, and have been found undesirable in many types of collaborative networked applications. Moreover, anonymity and pseudonymity mechanisms are less effective in mobile and emerging UbiComp environments to protect people's real-world information such as location or presence. For our target domain, i.e., personal information sharing applications, end users of those applications have already established some social relationships, and anonymising personal information is undesirable and unnecessary for preventing inadvertent privacy violations. *Therefore, anonymity and pseudonymity mechanisms are out of scope of our design. However, the failure of strong anonymity mechanisms motivates us to search for the goal for information privacy and provides our own definition of better privacy.*

We believe people's privacy-related decisions vary with their privacy requirements in different situations, and that awareness of privacy is the basis for making informed decisions about personal information disclosure [Langheinrich01]. *Therefore, we will employ mechanisms to promote awareness of system behaviour concerning users' information privacy, in order to empower users to make right decisions in different situations*. We believe accountability mechanisms are important for information privacy because they maintain knowledge of personal information disclosure on behalf of end users that enables people to base accountability on; helping users to detect undesirable operations on their information, and adjust their privacy-related behaviour as they encounter new situations. *We will employ mechanisms to increase accountability of the system behaviour concerning privacy and enable users to adjust their privacy preferences after information disclosure.* In summary, adaptive privacy management incorporates awareness and accountability mechanisms to empower people to effectively exercise control mechanisms at information collection.

Control at information use (especially control of secondary use) is difficult to achieve in heterogeneous distributed environments, and existing technical mechanisms (e.g.,

DRM and Trusted Computing) are not mature enough and have adverse impacts on end users. Our work does not focus on enforcement of privacy promises made by information collectors' and information users', i.e., preventing from secondary use. *Therefore, control at the point of information use is out of the scope of this thesis. Our proposed adaptive privacy management empowers data subjects (e.g., end users) to make right privacy-related decisions before they disclose their personal information.*

3.3 Requirements for Adaptive Privacy Management

We have proposed a new approach, i.e., adaptive privacy management, that enables end users or/and a system to dynamically adjust the amount of released information to achieve a desired level of openness under different circumstances in dynamic networked environments. The previous sections have critically analysed design strategies for information privacy, and provided rationales for incorporating specific technical mechanisms into the proposed adaptive privacy management system. In this section, we identify the set of requirements that we believe can be used to design an appropriate software architecture for supporting adaptive privacy management: enabling users to effectively managing their privacy while sharing personal information.

3.3.1 R1. Adaptive Privacy Adjustment and Evolution of Privacy Preferences

The process of adaptive privacy management is an on-going dialogue and collaboration between a user and the system, constantly negotiating the level of openness as close to the user's desired level for different situations. In contrast to the static policy approach that requires users to specify their privacy preferences as rules or policies a priori, adaptive privacy management should support both pre-specified and *zero* privacy configuration, where no privacy preferences are pre-specified in the system at the beginning of using a system. In case of zero privacy configuration, adaptive privacy management should enable users to make privacy decisions and set privacy preferences interactively when receiving requests for private information. In addition, adaptive privacy management requires enabling evolution of users' privacy preferences specified in the system over time as a result of on-going interactions between the user and the system. For ex-

ample, users should be able to modify privacy preferences as a result of consulting the information disclosure history; the system may also suggest new privacy preferences based on users' history of privacy interaction. The evolving privacy preferences effectively modify the system's behaviour of disclosing personal information according to users' changing desire for openness under different circumstances. Clearly given the potential level of user involvement, the system will need to support different levels of engagement to balance the privacy needs of the user with the effort and intrusion caused by interacting with the system. We discuss this issue further in section 3.3.4.

3.3.2 R2. Awareness of System Behaviour Concerning Privacy

In networked environments, it becomes harder for end users to formulate a *correct* desired level of openness, because ordinary people are often not aware of when their personal information is disclosed and often do not fully understand the privacy implications of releasing their personal information [Smith04]. For example, a user study of a smart environment for eldercare concluded that users cannot adequately evaluate privacy issues because they are not aware of and do not fully understand how and to what extent their private information was collected by the system [Beckwith03]. The first step in supporting this requirement is to promote users' awareness of the system's behaviour concerning their private information. This includes awareness of what the system can potentially do with users' personal information [Lederer04]. More importantly, adaptive privacy management requires promoting user's awareness of system's run-time behaviour concerning privacy and context in which this behaviour is demonstrated [Tsandilas04]. In other words, *the system needs to promote users' awareness of critical events concerning personal information requests and disclosure in a timely manner, e.g., notifying them when information is released and at what time*. Previous research [Lederer04] has showed that awareness of the system's runtime behaviour is crucial for end users to understand the effects of their use of the system and predict the consequences of future usage. Since adaptive privacy management is an on-going negotiation, the awareness of the system's runtime behaviour assists end users in establishing mental models that correctly match the conceptual model of the system [Tsandilas04].

3.3.3 R3. Convenient and Timely Access to Privacy Controls

Fine-grained privacy control interfaces typically are often complex user interfaces that require significant user effort to configure them properly, with the end result of that users often do not change the default settings at all; leading to disclosure of too much or too little personal information to others [Boyle05]. People tend to forget the details of previously specified privacy preferences and find difficulties in applying them in real situations [Lederer04], because de-contextualised privacy preferences [Grudin01] often fail to meet their privacy requirements in real use. Consequently, even if people were aware of system's run-time behaviour and the privacy implications thereof, they can still fail to employ privacy control mechanisms (e.g., access control) effectively [Gurteen02]. *Therefore, adaptive privacy management requires convenient and timely access to privacy controls, to encourage users to adjust the system's disclosure of personal information in response to changes in circumstance, in order to better match users' desire for privacy*. The combination of awareness of system's run-time behaviour and timely access to convenient privacy controls enables people to make contextualised decisions on disclosing personal information, and has the potential to enable the system and/or users to detect behaviour patterns [Grudin03]. We argue that this can enable evolution of privacy preferences over time.

3.3.4 R4. Balance between Privacy and User Involvement

There are very few fine-grained yet lightweight strategies in computer-mediated interactions [Bellotti97], and it is hard to design computing systems that provides both fine-grained and lightweight control for privacy [Boyle05]. The need for information privacy and the need for minimising user intrusiveness and involvement are contradictory goals that require a trade-off at the design stage [Myles03]. Since adaptive privacy management can only be achieved through negotiation and cooperation between the user and the system, *adaptive privacy management requires balancing end users' needs for information privacy with the level of involvement incurred by privacy-related interactions*.

Oppermann et al. identified different levels of user intrusiveness in a computer system as a spectrum of adaptation (figure 3.1). On one end of the spectrum, *adaptable* systems require direct user manipulation (e.g., via graphical user interfaces) to change systems' behaviour. At the other end, *adaptive* systems do not interrupt the user and au-

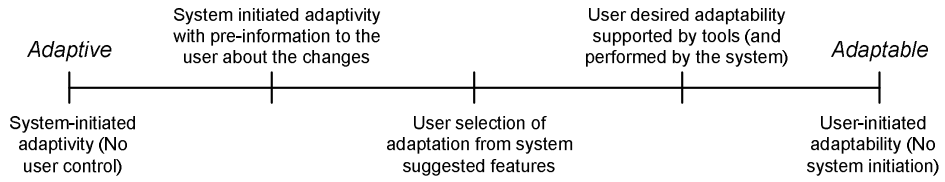


Figure 3.1: Spectrum of adaptation in computer systems [Oppermann97]

tomatically adjust systems' behaviour (e.g., via intelligent software agents). It has been a long debated topic [Shneiderman97] of the benefits of total automation of user needs and direct human manipulation: the Artificial Intelligence (AI) community favours the total automation of user tasks via intelligent agency and the HCI community emphasises the importance of direct user control and decision making via graphical user interfaces [Allen99]. Adaptive privacy management requires that the system supports a spectrum of adaptation that offers different levels of user involvement to enable a mixture of user and system effort in privacy management, i.e., system-initiated adaptation without user control, or user selection from system-suggested features. This flexible interaction strategy is often referred to as *mixed-initiative* [Allen99], because either the user or the system can initiate interactions to accomplish the same task.

3.3.5 R5. Accountability for Privacy-related Behaviour

Inadvertent privacy violations may occur in networked information sharing applications because people's actions and interactions are de-situated and de-contextualised [Grudin01] and people are no longer operating in clearly situated contexts [Palen03]. *Therefore, adaptive privacy management requires maintaining audit trails for privacy-related behaviours (i.e., information disclosed either explicitly by the user or automatically by the system) to increase accountability and traceability of the system*. Following Palen and Dourish's observations on information privacy [Palen03], we argue that no set of pre-specified control rules can meet user's changing requirements for privacy in dynamic environments, and undesirable information disclosure will happen as a result of de-contextualised privacy preferences [Grudin01] or failing to apply them in real usage situations [Lederer04]. Therefore, the audit trail for privacy-related behaviour can be used by the user and/or the system to detect undesirable information disclosure, and to adjust privacy-related behaviour in future situations and hence enable evolution of privacy preferences specified in the system. In particular, the audit trail for privacy-related behaviour can be the basis for users to make future privacy decisions and also act as cues

for a system to modify specified privacy preferences. Although this passive protection method does not prevent private information from flowing to others, maintaining traces of privacy-related behaviours creates a sense of accountability that is consistent with Jim Morris' vision of building computer systems *"to have the same privacy safeguards as the real world, but no more, so that the ethical conventions will apply regardless of setting"* 3.2.3.

3.4 Summary

In this chapter, we analysed design strategies for achieving information privacy in networked computing environments. We explained rationales for selecting access control at information collection as our main underlying technical mechanism, and incorporating awareness and accountability mechanisms to empower people to effectively exercise control mechanisms through the adaptive approach. The key design requirements for supporting adaptive privacy management were presented. The next chapter presents the design of platform and applications to support adaptive privacy management based on the aforementioned requirements.

CHAPTER IV

Design

Contents

4.1 Overview

This chapter presents the design of a middleware platform that incorporates adaptive privacy management into distributed applications that enable intentional private information sharing. We start by highlighting the key design decisions for our target class of applications in order that they meet the requirements for adaptive privacy management identified in the previous chapter. The rest of the chapter focuses on designing the underlying architecture. After providing background knowledge on distributed systems architectures and middleware support, the chapter motivates the need for a flexible middleware platform to support the development of adaptive privacy aware applications. Following this discussion, the chapter provides a high-level description of architectural design for applications that interact with the proposed platform. A prototype implementation of the platform as well as sample applications built using the platform is presented in the next chapter.

4.2 Design Decisions for Adaptive Privacy Management

In this section we explore the key design decisions for supporting adaptive privacy aware applications that we propose for meeting our requirements for adaptive privacy management. In the following sections, we first discuss critical factors that affect people's privacy decisions and present design considerations related to these factors. Next, we propose notifying users of critical events concerning privacy to promote awareness of system behaviour concerning privacy (**R2**); providing multi-modal and multi-device interactions to provide convenient and timely access to privacy controls (**R3**); automating privacy decisions using privacy rules to balance between privacy and user intrusiveness (**R4** and **R1**), facilitating evolution of privacy rules to enable evolution of privacy preferences (**R1**), and maintaining status for privacy-related interactions to create accountability for privacy-related behaviour (**R5**). Finally, we describe the support for plausible deniability that is important for computer-mediated social interaction.

4.2.1 Critical Factors for Privacy Decisions

As Adams [Adams01b] concluded: privacy decisions are mostly personal choices and they largely rely on whether people perceive themselves to be private. Therefore, un-

derstanding critical factors influencing people's privacy decisions is important to design systems for protecting information privacy. This section explores critical factors that affect people's decisions in disclosing their personal information to other parties, and describes the relationship between these factors and design considerations of adaptive privacy aware applications.

Previous research on perception of privacy (section 2.3.1) provides us with guidance for designing the format of *private information requests* (which we refer to as *privacy requests*) and *disclosure* in our system. In studying people's privacy perception in multimedia communications in late nineties, Adams proposed a privacy perception model and identified three factors — *information sensitivity, information receiver*, and *information usage* — that are critical for people to make privacy decisions [Adams01a]. The questionnaire-based user study conducted by Lederer et al. [Lederer03a] showed that the *identity of the information inquirer* is a stronger determinant than people's *situation at the time of inquiry* for making decisions on disclosing personal information. The user study conducted by Consolvo et al. [Consolvo05] on location privacy, demonstrated that the identity of the information requester (i.e., *who*) and the proposed purpose of the request (i.e., *why*) are the most important factors for people to decide whether to disclose their location. Chatfield et al. [Chatfield04] found the most influential factor on user information sharing is the existing relationship between the users. Based on these observations, *we will use the identity of the information requester as the primary index for a private information request*. Moreover, the user study by Olson et al. [Patil05, Olson05] showed that most people cluster information recipients into a manageable set of categories, and *the result of this study has motivated us to design mechanisms to allow users to categorise information requesters into groups* and hence simplify end users' privacy management tasks. We will discuss this in more detail in section 4.2.4.

As illustrated by previous studies [Lederer03a, Consolvo05], factors having secondary influence on privacy decisions include contextual information of the request, e.g., the time of the request, people's location and activity at the time of the request. The aim of our system design is to assist people in sharing their personal information in computer-mediated social interactions while maintaining their privacy, and participants of our target applications have already established social relationships between them. Therefore, we argue that only a little context (e.g., a short message) is sufficient for each information request given participants' existing knowledge about each other. Based on the above observation, *we have decided that privacy requests in our*

system should contain a field for including contextual information, so that information requesters can provide extra information (e.g., proposed purpose of the request) for assisting requestees in making the disclosure decision. We conjecture that much more contextual information would be needed to encourage strangers to disclose personal information to each other, and technical mechanisms such as P3P and reputation systems [Resnick00] have the potential for tackling this type of problem. However, this is outside the scope of this thesis.

Aside from the context of requests, the users' privacy decisions are also influenced by other factors including the time, how frequently the information is asked for, the quality (or fidelity) of the information being disclosed, and how long it will be kept for [Beresford05]. However, making privacy decisions based multiple secondary factors requires significant additional cognitive effort [Boyle05], and could make interactive control of privacy too challenging to do effectively if too many secondary factors are presented to the user interactively. *Therefore, we incorporate fine-grained control of secondary factors into privacy rules so that users can control their personal information disclosure, to balance the need for information privacy and the requirement of minimising user intrusiveness* (**R4**). *Therefore, we incorporate secondary factors for people to set fine-grained privacy rules that control their personal information disclosure, to balance the need for information privacy and the requirement of minimising user intrusiveness* (**R4**).

4.2.2 Notifying Users of Critical Events Concerning Privacy

*To promote users' awareness of system's run-time behaviours concerning privacy (**R1**), we propose notifying end users of critical events concerning privacy requests and information disclosure in a timely manner.*

By notifying a requestee (i.e., the user whose personal information is requested) of incoming privacy requests when they occur, *the requestee will be able to make privacy-related decisions in their current situational context*. An incoming request has to contain relevant information to enable the requestee to make a sensible decision on disclosure. The format of information request in our system will use the identity of the information requester as the main index to the context, and provide a field for providing additional information about the purpose of request (or any other relevant context) to be included in the request. Factors having secondary influence on privacy decisions (e.g., the time of

the request, the service used for disclosing personal information) will also be included in the request, to aid the user and to enable automating privacy decisions using privacy rules.

A requestee's privacy decision results in selective disclosure of private information, and timely notification of information disclosure will enable the requestee to know the effect of the decision and understand how the system works over time. *Fundamentally, the system needs to notify the requestee what personal information is released to whom at what time with a sufficiently low latency* as to be useful to both parties. Since adaptive privacy management allows both interactive processing of privacy requests by users and automation of privacy decisions using privacy rules, information disclosure can be enabled by users' explicit interactions or a privacy rule stored in the system. Previous research has shown that people tend to forget pre-specified privacy rules [Lederer03b], and therefore it is important for users of our system to know the operation and hence effect of the privacy rules they have specified. Users are encouraged to adjust the level of openness as their environment or their requesters' environments change — an important dialectic on-going negotiation, which is crucial for users to adaptively balance level of openness and enable the evolution of their privacy preferences.

4.2.3 Providing Multi-modal and Multi-device Interaction

Today it is common practice to utilise multiple computing devices (including desktop PCs, laptops, personal digital assistants and mobile phones, etc.) to communicate effectively with others in everyday professional and personal life. Increasingly, these devices can be conveniently connected to a computer network and hence become a component in a distributed system. The vision of UbiComp promises that computing capabilities will become further integrated and embedded into the objects we interact with. Human computer interactions are bi-directional: people cannot only receive feedback such as notifications but also exercise controls through user interfaces. Users can interact with a system in multiple modalities, using different communication channels, e.g., receiving feedback via visual and audio channels using conventional displays and speakers, exercising controls using keyboards and mice, pen-based stylii, etc. *For our target domain (which includes interpersonal communication of private information), we believe that adaptive privacy aware applications should provide multi-modal and multi-device interactions for end users, in order to provide convenient and timely access to privacy*

*control (**R3**) as well as promote awareness of privacy (**R2**).*

Multi-modality [Dix98] is a well-researched area in Human Computer Interaction, and multi-modality can increase system usability by offsetting the weaknesses of one modality by the strengths of another. For our system, providing multi-modal interactions using multiple devices simulataneously can facilitate notifying users in a timely manner, because users might only be available on certain devices at a certain instant (e.g. when they are away from the desktop). In addition, multi-modal and multi-device interactions enables users to select the most appropriate modality and device for performing privacy related interactions. For example, a user on the move might choose to type SMS messages from mobile phone for accepting or rejecting private information requests, while a user working on his desktop PC might use traditional keyboard and mouse to create privacy rules for incoming requests. Therefore, offering end users multi-modal interactions available on multiple personal devices can not only promote users' awareness of system's run-time behaviours concerning privacy, but also provide them convenient and timely access to privacy controls.

4.2.4 Automating Privacy Decisions using Privacy Rules

Previous HCI research has shown that a challenge for preserving privacy is to provide sufficiently fine-grained control with little cognitive or physical effort [Bellotti97]. As we've discussed in section 3.3.3, specifying privacy control in one place (e.g. in applicaton preferences) results in complex control interfaces and decontextualised configuration actions, and therefore they often require significant user effort to configure them properly; the end result is that users disclose too much or too little personal information to others [Boyle05]. *To balance between privacy management and user intrusiveness, we propose to give users privacy controls that require different levels of cognitive or physical effort*. Recognising the importance of coarse-grained privacy controls as observed by Lederer et al. [Lederer04], we decide that adaptive privacy aware applications *should provide coarse-grained controls for users to make interactive privacy-related decisions in a disclosure situation*. In addition, we propose that adaptive privacy aware applications should enable users to automate decision-making process in privacy management using privacy rules, where fine-grained privacy controls can be exercised by the system in an autonomous manner. Unlike the traditional privacy policy languages have been found to be too complex for 'normal' people to incorporate and use

[Hochheiser02]; privacy rules in our system should be easy for users to understand and process. *We propose to allow end users to specify privacy rules using familiar end user interfaces (e.g., web forms) and provide them an explanation of their privacy rules in colloquial language*.

As defined by Cuellar [Cuellar02], a privacy policy or rule is an assertion that a certain piece of personal identifiable information may be released to a certain entity or a group of entities under a certain set of constraints. For example, privacy rules can be described in colloquial language as the follows: "to allow my spouse to know my location at the best accuracy anytime", "to allow my colleagues to know my in/out status during working hours on weekdays", or "to disallow my boss to know my location outside of working hours", etc. *In addition to using positive privacy rules that automatically disclose private information, our system should allow users to specify negative privacy rules [Rabitti91] that automatically deny information requests*. Since the identity of the information requester (i.e., who) is the most critical factor for making privacy-related decisions, it is natural to impose restrictions of personal information disclosure based on the identity of the information requester. As demonstrated in previous research [Myles03, Kupper05], there exists a variety of constraints that can be employed as privacy control parameters, including name and type of information, name and type of service for information disclosure, contextual information such as time, location and activities, etc. We propose to use some of these constraints in our system within privacy rules, and we will discuss the detailed structure of a privacy rule in the next chapter (section 5.3.4). Since most people cluster information requesters into a manageable set of categories [Patil05, Olson05], we propose to use social groups to simplify users' privacy rule management. In particular, users should be able to create their own social groups and manage members of each group. For example, a user can create a social group called "family members" and add her spouse, parents and children into the group. Instead of creating a privacy rule for each individual, the user can create a single rule for all of the group members, reducing the number of rules required in the system and facilitating the task of rule management.

4.2.5 Facilitating Management of Privacy Rules

Unlike approaches requiring definition of static policies, our system does not require that preferences are established *a priori*; by default we maintain that users should con-

trol their personal information disclosure interactively as they receive privacy requests. By providing end users with opportunity to make privacy-related decisions 'in context', our system assists them in forming predictable user models within context of use over time [Tsandilas04] and encourages them to create and modify privacy rules. To facilitate users' task of privacy management, our system *should enable them to create privacy rules dynamically in response to processing one or more privacy requests using multi-modal interactions*. We also choose to promote users' awareness of the runtime execution of privacy rules by notifying them when privacy events are handled automatically by one of their rules, e.g., notifying the requestee what information has been disclosed to whom at what time. The notification mechanism enables users to build an understanding of the operation of the system and the effect of privacy rules over time, which we contend, motivates users to adjust their level of openness to their changing requirements for privacy.

To effect this, our system will enable a mixture of system and user effort to create and modify privacy rules, and our design should allow us to remain flexible as to supporting different levels of adaptation to balance between privacy management and user intrusiveness (**R4**) to meet requirements of different problem domains. More specifically, our system should be able to support user-initiated actions for creating a privacy rule, system-initiated actions of autonomously generating new privacy rules, and mixed-initiative actions, where the system suggests new rules or the modification of existing rules based on the disclosure history for the user to make the final decision. The design of our system should be flexible to allow multiple strategies for rule creation, suggestion, and conflict resolution to be plugged-in, so that different preference learning mechanisms and Artificial Intelligence (AI) algorithms [Viappiani02, Pu06] could be employed to implement these strategies.

4.2.6 Maintaining Status for Privacy-related Interactions

As we've previously discussed, undesirable information disclosure can happen as a result of de-contextualised privacy preferences [Grudin01], and inadvertent privacy violations may occur in networked information sharing applications because of people's desituated and de-contextualised [Grudin01] actions and interactions. To increase accountability and traceability of the system (**R5**), adaptive privacy management *requires the maintaining of an audit trail of privacy-related behaviours, e.g., information disclosed*

either explicitly by the user or automatically by the system. Since intentional sharing such information involve interactions such as sending private information requests and receiving decisions on information disclosure that may be personally reviewed by individuals, these interactions are stateful interactions, whose state changes as the result of interactions by the user and the system. We propose that applications should maintain the status of these privacy-related interactions in a repository for inspection by the user and the system.

Importantly, maintaining the status of privacy interactions enables a mixture of end users and the system to detect undesirable information disclosure and to adjust privacy-related behaviour for future situations. Applications can retrieve status information of privacy interactions, and users can view the audit trail that may influence their future actions. In addition, software components can be developed using different algorithms to analyse the audit trial, in order to detect unusual or undesirable information disclosure and automatically generate or suggest new privacy rules for processing requests in situations that undesirable disclosure occurred. Finally, maintaining privacy rules as well as audit trial of privacy-related interactions facilitates evolution and generalisation of privacy rules.

4.2.7 Providing Support for Plausible Deniability

In computer-mediated social interactions, a requestee can often achieve *plausible deniability* by ignoring incoming requests (e.g., messages or calls) without having to explain why, because the requester cannot determine whether the requestee intentionally denied the request. A previous study [Aoki03] on teenagers' behaviour of using mobile phones showed that they do not always respond to the calls from their mobile phones and claim that they didn't hear the ringing or that the battery was dead. Nardi et al. [Nardi00] reported that users of Instant Messenger (e.g., MSN messenger, Jabber, etc.) often use the inaccuracy of presence indicators as a form of plausible deniability, i.e., they ignore requests for online conversation from a requester and requester will not know whether they are really there. Lederer et al. [Lederer04] argued that plausible deniability is important for people to continue exercising established social practices and hence is crucial for designing successful privacy management solutions. Aoki and Woodruff [Aoki05] argued that plausible deniability in social interactions is beneficial for avoiding social embarrassment and maintaining harmony in social relationships. People showed their

demands for plausible deniability in previous user studies [Hindus01, Hong05], and Hong [Hong05] further concluded that the studies indicated end users' requirements for avoiding potentially embarrassing social situations, undesired social intrusions, and unwanted social obligations.

We have decide to provide a means for end users to achieve plausible deniability in personal information sharing applications built using our adaptive privacy management architecture. In particular, when a requester issues a personal information request to a requestee from an application, the system should intercept the request and maintain its status as 'waiting for the recipient's approval'. Although the system will notify the requestee receiving this request, they can choose to ignore it and the requester will not be able to determine whether the requestee deliberately ignored the request. Finally, each information request should contain an *expiration time*; if the requestee does not respond within this interval, the request is expired and no further processing is done.

4.2.8 Summary

In the previous sections, we have presented the following key design decisions to meet the requirements of adaptive privacy management set in chapter 3:

1. We will use the identity of the information requester as the primary index for a privacy requests and incorporate a field for users to specify extra contextual information for a request;

2. The system will notify users of critical events concerning privacy to promote their awareness of system behaviour (**R2**);

3. It will provide multi-modal and multi-device interactions to promote privacy awareness (**R2**) and convenient and timely access to privacy controls (**R3**);

4. It will support the making of privacy decisions within the context of receiving a privacy request to enable adaptive adjustment of the individual's level of privacy (**R1**);

5. Privacy decisions will be automated using 'privacy rules' to balance between privacy and user intrusiveness (**R4** and **R1**),

6. We will facilitate the management of privacy rules to enable evolution of privacy preferences (**R1**);

7. The status of privacy-related interactions will be maintained to create accountability for privacy-related behaviour (**R5**);

8. It will provide support for plausible deniability for computer-mediated social interactions.

4.3 Incorporating Privacy into Distributed Applications

We have presented the key decisions for designing adaptive privacy aware applications in order that they satisfy the requirements set out in the previous chapter. This section focuses on designing an architecture that incorporates adaptive privacy management into distributed applications in our target domain. We start by discussing how distributed applications are built, i.e., background knowledge on distributed system architecture and middleware support. Then, we motivate the need for a flexible middleware platform to support the development of adaptive privacy management. Finally, we provide a high-level description of architectural design for adaptive privacy aware applications that interact with the proposed middleware platform.

4.3.1 Distributed System Architectures

In order to understand how information privacy in distributed system can be achieved, it is first necessary to review how distributed applications are constructed and how the entities in such systems communicate. In this section, we briefly review common distributed system architectures, focusing on the predominant techniques for constructing distributed systems software; including socket programming, synchronous middleware and asynchronous middleware. Finally, we discuss the potential privacy threats in such architectures and identify how information privacy solutions can be incorporated at the middleware layer to support the development of privacy aware applications.

In a distributed system, software processes running on different computers interact with each other to perform the overall task of the system. The interactions between these entities are known is *Inter-process Communications* (or IPC). Such software processes

are often modelled as components that may be strategically placed within the network to optimise a given computational or interactional property.

The most important and widely used architectural paradigm is the *client-server architecture* (figure 4.1). In the client-server model, a *server* is the software process on a machine that provides a service, e.g., offering file access over a network, routing data to a printer, accessing a database, etc. A *client* is the software process on a machine that is requesting the service. The names 'client' and 'server' are only meaningful within the context of a particular interaction, and a server for one service can be a client of another. As shown in figure 4.1, server C acts as a server of client A and client B, and it acts as a client of server D.

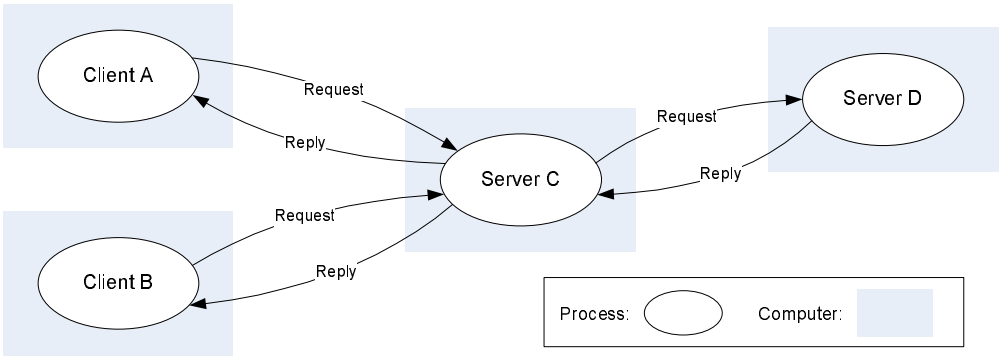


Figure 4.1: Client-Server Architecture Adapted from [Coulouris01]

The most fundamental way for communicating between software processes running on different machines is to exchange messages using the BSD socket application programming interface to a transport protocol such as TCP and UDP, layered over the network protocol IP (the *Internet Protocol*). A socket is defined as "the unique identification to or from which information is transmitted in the network" [Winett71], i.e. the endpoints of the communication. TCP sockets provides a connection-oriented, bidirectional byte-stream abstraction between pairs of processes. UDP sockets provide a message passing abstraction that allows a sending process to transmit a single message (called a datagram) to a receiving process. The programming model offered by the socket programming interface provides us send/receive or read/write primitives with which to design distributed systems and applications.

Typically application designers layer their own 'application layer protocol' on top of these basic transports to suit their own application requirements. Examples of these types of protocols in include a variety of high level request-reply protocols including FTP, HTTP, TELNET and SMTP (implemented using TCP sockets) and protocols such

as DNS, NTP and SNMP (over UDP). With the popularity of the Internet and the ubiquity of web browsers and applications, HTTP-based communications has become a popular choice for all manner of distributed applications.

The most widely used design pattern for web applications is the so called *3-tier software architecture* that extends the basic client-server pattern to include a presentation layer for providing user interfaces of the application and *a data access layer* for storing/retrieving data into/from backend data storage (e.g., a relational DBMS). The middle tier (i.e., *the business logic layer*) situates between the these layers and incorporates the core logic of the application. The middle tier can be multi-tiered itself due to the complexity of the application logic, in which case the overall architecture is sometimes referred to as *n-tier* architecture. By separating concerns of data management, application logic, and user interface into different entities, the 3-tier architecture is argued to provide increased performance, flexibility, maintainability, reusability, and scalability for distributed applications [CMU SEI00].

Finally, it is worth noting, that TCP and UDP sockets (i.e. the Internet protocol stack) are important building blocks underpinning forms inter-process communication often found in high-level middlewares which provide alternative models for designing distributed systems. We categorise high-level distributed system middleware into two classes (i.e., synchronous and asynchronous middleware) and describe each in turn in the following sections.

4.3.2 Synchronous Middleware

According to Hadzilacos and Toueg's definition [Hadzilacos94], a synchronous distributed system has the following properties: first, the time to execute each step of a process has known lower and upper bounds; second, each message transmitted over a channel is received within a known bounded time; and third, each process has a local clock whose drift rate from real time has a known bound. Therefore, it is possible to use timeouts to detect failure of a process in a synchronous distributed system. There are a number of middleware platforms that provide programming interfaces for building synchronous distributed systems.

One of the widely known synchronous distributed system model is the Remote Procedure Call (RPC) proposed by Birrell and Nelson [Birrell84] in 1984, which allows

software processes to call procedures running on remote machines. During an RPC call, the process on local machine is suspended, and the parameters are transferred across the underlying network to the remote machine where the desired procedure is executed. When the procedure finishes and returns its results, the results are passed over the network back to the local machine where the calling process resumes execution. The RPC model allows programmers to use local procedure call semantics to write distributed application instead of the send/receive or read/write interface provided by sockets, and it hides away low-level details of message exchange between processes.

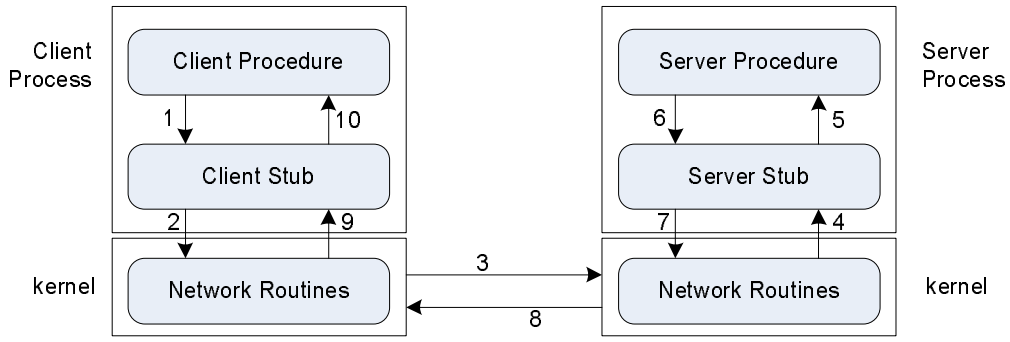


Figure 4.2: Functional Steps in a Remote Procedure Call Adapted from [Stevens90]

RPC, depicted in figure 4.2 (showing sequences of operations involved in a single RPC call), requires the creation of *stub procedures*. A programmer typically specifies definitions of the RPC interface using an Interface Definition Language (IDL), and a separate compiler generates the stub procedures from these definitions. At runtime, the client procedure calls the client stub, which appears like a local procedure but actually contains code for exchanging messages over the network. The client stub packages the arguments to the remote procedure into one or more network messages in an internal format, and this process is called *marshalling*. The actual message exchanges are handled by network drivers in the kernel. The server stub receives the messages and converts them into the arguments for the remote procedure, and this process is called *unmarshalling*. The server stub calls the actual procedure on the server and receives the return value from the procedure. The server stub marshals the return value into network messages and sends them back to the client stub. The client stub receives the messages, unmarshals them into local format, and returns the result to the client procedure.

Sun's RPC [Srinivasan95] as part of the Open Network Computing (ONC) architecture was one of the first RPC based middlewares, and it has been used to build Sun Network File System (NFS). Sun's RPC can use either TCP or UDP for transporting messages across a network, and it defines the eXternal Data Representation language

(XDR) as the format for encoding data exchanged between heterogeneous machines. With the popularity of the object-oriented (OO) programming in late 1980's, the original RPC mechanism was extended to provide support for invoking methods of remote objects. To enable this remote method invocation, the middleware provides support for instantiating remote objects from remote classes, keeping tracking of instances of objects, and providing support for polymorphism.

Common Object Request Broker Architecture (CORBA) [OMG04a] proposed by Object Management Group (OMG) provides support for distributed heterogeneous object-oriented applications. When a client wants to invoke a remote method in the CORBA object model, it makes a request and gets a response through the Object Request Broker (ORB), which hides details of communication, object activation, and storage of server objects from the client. CORBA supports both static and dynamic method invocation. In the static approach, object interface definition specified in OMG IDL was compiled to generate client stubs and server stubs (i.e., skeleton in CORBA), which are then built into distributed applications. The dynamic approach allows a client to discover names of classes and methods at runtime via Interface Repository and invoke methods on a remote object without compile time knowledge of the remote object's IDL. CORBA defined the General Inter-ORB Protocol (GIOP) for supporting interoperability between different implementations of ORB, and specified the Common Data Representation (CDR) as the standard format for encoding method calls into network messages.

Java RMI [Pitt01] provides remote method invocation for distributed Java objects, and it employs TCP and object serialisation to transport messages between different machines. An advantage of Java RMI is that it does not require a language- and platform-independent interface definition, and a programmer can define an interface and provide its implementation within a single Java file. Latterly, dynamic invocation of remote Java objects has been provided through Java's reflection mechanism, which allows a client to discover methods of a remote object and invoke those methods with dynamically constructed arguments at runtime.

A common problem with most RPC-based distributed systems is that they do not work well across firewalls, because firewalls may block certain ports used for communication. Different RPC-based distributed systems typically do not interoperate, and it is largely due to the incompatibility between different formats for converting procedure call arguments into network messages. To overcome problems of traditional RPC

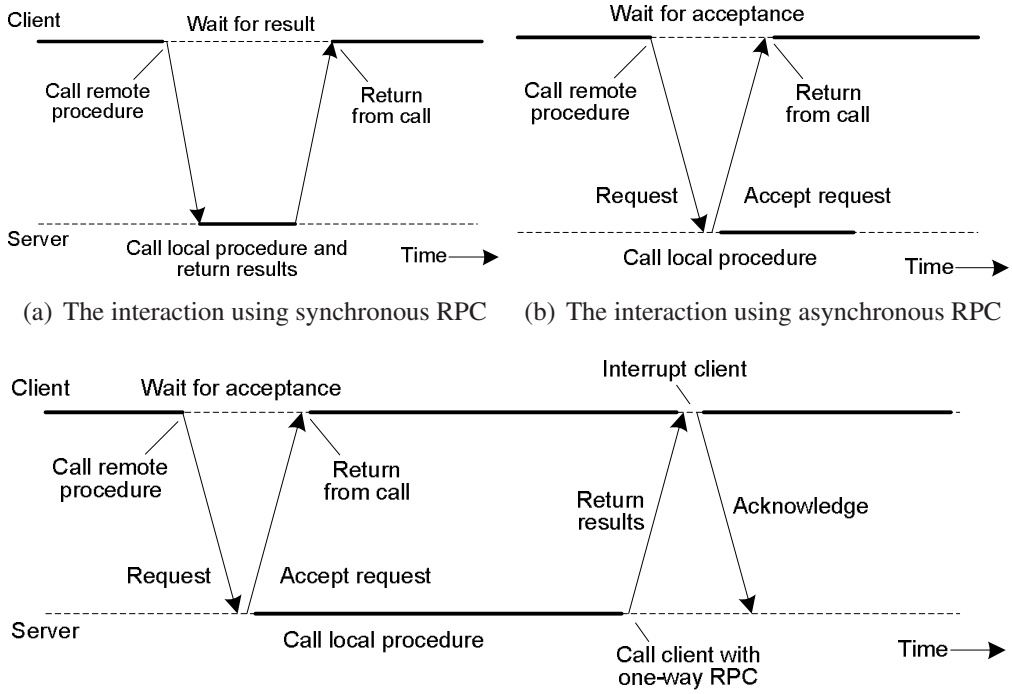
systems, the World Wide Web Consortium (W3C) propose d*Web Services* as *"software systems designed to support interoperable machine-to-machine interaction over a network"* [Booth04]. Interfaces of web services are described in Web Service Description Language (WSDL) [Chinnici06] that provides an XML-based grammar for structured description of web services and operations/methods they expose. A web service description in WSDL contains all the information that is required to dynamically discover and interact with the service. Systems and applications interact with web services using Simple Object Access Protocol (SOAP) [Gudgin03] messages are typically transported using HTTP, which can often traverse firewalls. SOAP is an XML-based protocol for exchanging structured information in distributed systems, and it provides a standard method of converting information for invoking remote services into an open format that can be exchanged over a variety of underlying protocols.

4.3.3 Asynchronous Middleware

In an asynchronous distributed system, it may take an arbitrarily long time to execute a step of a process or to wait for a message to arrive, and the clock drift rate can also be arbitrary [Coulouris01]. Actual distributed systems are very often asynchronous due to the demand for processes to share computational power and network bandwidth. The Internet is a good example of asynchronous distributed system, because there is no intrinsic bound on server and network load for the Internet and therefore we cannot reliably estimate the time for transferring a file or receiving an email. In this section, we examine different types of asynchronous distributed system middleware, from asynchronous RPC to middleware based on the Tuple Space and publish-subscribe paradigms.

When invoking a remote procedure call in a synchronous RPC system, the client will block until a reply from the server is returned. This behaviour is undesirable when the execution of the call on the server takes arbitrary long. Asynchronous RPC was proposed to extend the synchronous RPC mechanism by allowing a client to continue execution after issuing an RPC request, without the need to wait for the server to finish the procedure and return [Tanenbaum06]. In an asynchronous RPC system, the server sends a reply to the client as soon as an RPC request is received, the reply acts as an acknowledgement that the server has received the request and is going to process it. After the client receives the acknowledgement, it continues execution without waiting for the RPC to finish. Figure 4.3 shows the client and server interactions for both syn-

chronous and asynchronous RPCs. After finishing executing the RPC, the server can initiate another asynchronous RPC to interrupt the client and return the result of the RPC. The mechanism of combining two asynchronous RPCs is sometimes referred to as a *deferred synchronous RPC* as illustrated in figure 4.3.



(a) The interaction using synchronous RPC (b) The interaction using asynchronous RPC

(c) A client and server interacting through two asynchronous RPCs

Figure 4.3: Asynchronous Remote Procedure Call from [Tanenbaum06]

The Tuple Space paradigm was originally proposed as a shared distributed memory model for parallel computing in Linda [Gelernter85], and it was adapted to create middleware for distributed systems. Tuples are data structures that consist of a sequence of typed data fields. Tuples can be inserted into the tuple space using the write (*out*) operation, and copied or removed from tuple space using the read (*rd*) or take (*in*) operations. In this paradigm, different systems or applications do not directly interact with each other, but interact indirectly using the tuple space operations. The L^2imbo platform [Davies98] developed at Lancaster University is an asynchronous distributed system middleware designed for mobile computing environments based on the tuple space paradigm. L^2imbo allows multiple tuple spaces to be created across machines by employing an IP multicast based consistency protocol. Bridging agents can be employed to propagate tuples between different tuple spaces.

The Event Heap [Johanson02] developed at Stanford University is another asynchronous middleware based on the Tuple Space paradigm. It was designed to support

the development of UbiComp systems called interactive workspaces, where people can collaborate using a variety of computing devices and large situated displays. To meet the requirements for interactive workspaces, Event Heap extends the original model to provides additional features: every field in a tuple can be described using meaningful name; the field order and size of a tuple in Event Heap are ignored and tuple matching is achieved using the named parameters; a read operation always returns the earliest matching tuple; and all tuples have a "TimeToLive" field specifying how long they will persist in the heap.

Distributed event-based systems extend the local event model by allowing multiple processes on different machines to be notified of events generated by other processes. In the asynchronous event notification or *publish-subscribe paradigm*, processes that generate events (i.e., publishers) are loosely coupled with the processes that subscribe to certain types of events (i.e., subscribers). Publishers and subscribers exchange information based on the message content rather than direct message exchange between designated addresses. Publishers can delegate the delivery of events to the publish-subscribe infrastructure. Subscribers register the event types they are interested in receiving and consume the notifications when they are published. We briefly summarise some important publish-subscribe middlewares.

The CORBA Event Service [OMG04b] allows CORBA objects to communicate with each other using events or notifications. Notifications are delivered as arguments or results of ordinary synchronous CORBA remote method invocations. Notifications can be either pushed from publishers to subscribers or pulled by subscribers from publishers. The CORBA Notification Service [OMG04c] extends the CORBA Event Service to provide support for defining structured events and providing filtering at the event service.

The Cambridge Event Architecture (CEA) [Bacon95, Bacon00] was designed to extend existing synchronous object-oriented middleware such as Java RMI, CORBA, and DCOM with the publish-subscribe paradigm. In CEA, an event type is specified using a language-independent IDL, and any object can publish event types in the IDL that clients can subscribe to, in addition to the object's regular interface description. Each object contains a register method in its interface that enables clients to subscribe to a particular type of event. As illustrated in figure 4.4, the CEA supports direct source-to-client event notification, and it allows event mediators (or event brokers) to be placed

between publishers and subscribers. Event mediators remove the filtering computation from resource deficient publishers, and they can register interest with required event sources and buffer event notifications from these sources. The mediated model offers better scalability and can be potentially useful for mobile computing environments where users might intermittently disconnect from networks.

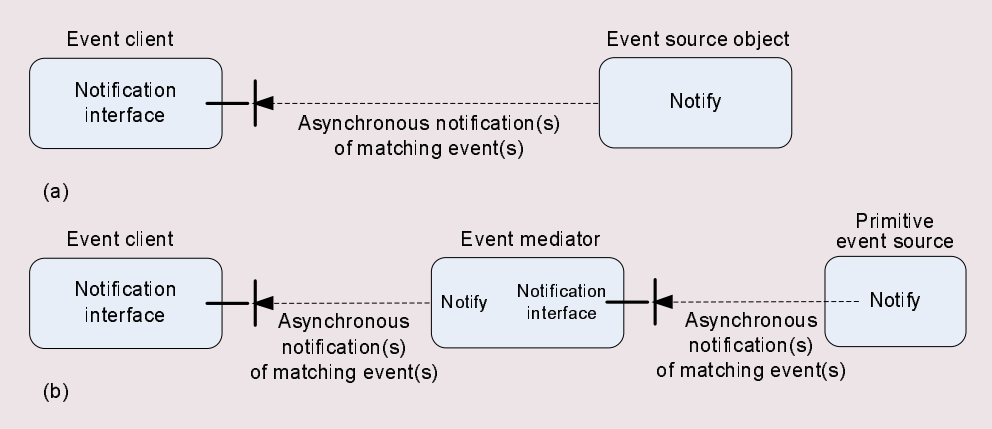


Figure 4.4: Event Notification in ECA: (a) direct and (b) mediated [Bacon00]

The Elvin router [Segall97] is a client-server architecture which acts as an event notification router between multiple connected clients that can be both publishers and subscribers. The notification router is responsible for routing notifications from event publishers to interested event subscribers. Elvin supports both topic-based and content-based subscription: a subscriber can specify event types (i.e. topic) it is interested in, and then supply filtering expressions that can operate on the attributes of this event type. To address the scalability issue of using a central notification server, the original Elvin architecture was extended to allow multiple notification servers (i.e., a *federation*) to route notifications. Elvin clients do not need to know the details of the federation.

4.3.4 Discussion

We have provided an overview of the important classes of distributed system architecture and middleware for constructing distributed applications. In this section, we analyse the privacy threats in distributed systems and examine how privacy solutions might be incorporated.

Traditionally, major threats to information privacy in distributed systems refer to attacks on the confidentiality and integrity of personal information because personal

information can be intercepted while transferring in plaintext over open networks. Classic computer security mechanisms such as encryption and digital signatures (section 2.5.1) are often employed to protect personal information from unauthorised disclosure and modification. For example, Secure Socket Layer (SSL), and its successor, Transport Layer Security (TLS) (section 2.5.1) provide a secure communications channel for different application layer protocols (such as HTTP, SMTP, or FTP) by employed cryptographic mechanisms to TCP messages. More recently, traffic analysis attacks have emerged as a new type of threat to information privacy, which intercept and examine messages transmitted over a network to find personal identifiable information, e.g., the identity of the sender and receiver. Encrypting messages cannot prevent traffic analysis attacks, because these attacks are not performed on the texts in the messages directly, but rather deduced from the patterns of communication, e.g., the frequency and timing of network packets [Goldschlag99]. As a result, anonymity and pseudonymity technologies (section 2.5.2) have been developed which mask the parties' identity.

However, threats to information privacy are not limited to personal identifiable information intercepted by unauthorised parties. For applications in our target domain (i.e., intentional personal information sharing applications), private information can be disclosed voluntarily or accidentally by end users, and automatically disclosed by applications on behalf of end users. Inadvertent privacy violations often occur in networked information sharing applications because users' actions and interactions are desituated and decontextualised [Grudin01] and users are no longer operating in clearly situated contexts [Palen03]. Unlike encryption and anonymity mechanisms that aimed to conceal personal identifiable information as much as possible, privacy solutions for information sharing applications try to provide end users with better privacy (as defined in Chapter 1).

To help end users achieve better privacy, a number of researchers have augmented important application interactions with metadata concerning the privacy implications and requirements, and developed appropriate mechanisms to promote privacy awareness and provide control of personal information flows [Langheinrich02b, Lederer04, Hull03, Dragovic05a, Hong04a] (section 2.5.5). For example, pawS promotes privacy awareness using privacy beacons that announce privacy policies for the services, and employs privacy proxies for handling privacy-related interactions on behalf of end users. In the Houdini framework, personal information sharing interactions are intercepted by the Privacy-Conscious Personalising (PCP) engine, which controls the access to and

distribution of personal information based on requestee's static data, requestee's context (dynamic data), requester's context, and requestee's privacy preferences. InfoSpaces in the Confab architecture extend the basic functionalities of Tuple Spaces by augmenting the operators (e.g., in, out) with privacy-related operations. For instance, in-operators are performed on incoming tuples to enforce access control policies and make sure the tuples can be added to an InfoSpace, and out-operators are performed on outgoing tuples to enforce privacy, e.g., blocking outgoing tuples, adding privacy tags, notifying end users, etc.

From the above analysis, we found that better privacy solutions can be incorporated into distributed applications by intercepting important application interactions and augmenting those interactions with privacy implications. Since most of distributed applications are built using middleware or protocol interactions over the IP suite of protocols, we argue that it would be natural to augment the platform and application layer protocol interactions with privacy related information and introduce better privacy solutions at the middleware layer. However, middleware support for better privacy solutions is a relatively new research area, and it is still the subject of ongoing debate and experimentation. We defer the detailed discussion of this issue to the next section.

4.4 Support for Adaptive Privacy Management

The previous sections provided an overview of important classes of distributed system middleware, and concluded that solutions for better privacy can be incorporated at the middleware layer by intercepting important interactions and augmenting them with privacy implications. In this section, we motivate the need for middleware support for incorporating better privacy solutions into applications, and decide on designing a middleware that facilitates developing adaptive privacy aware applications. Finally, we argue the design of the middleware has to be flexible and configurable, in order to provide customisable adaptive privacy solutions to accommodate the requirements of different problem domains.

4.4.1 The Need for Privacy Middleware

In the previous chapter, we have discussed a variety of established middleware [Pitt01, OMG04a, Booth04, Srinivasan95] for developing distributed applications. However, there is relatively little practical support for incorporating "better privacy" solutions into applications. Design principles and frameworks have been available for many years, and they are still the most important tools that support the development of privacy solutions.

Fair Information Practice (FIP) principles [US Dept. of Health73, OECD80] (section 2.4.1) remain the most influential guidelines for collecting and processing sensitive personal information in an appropriate manner. However, FIP principles are high-level and abstract principles and provide little practical guidance for building privacy-aware applications [Jensen05]. Surveys [FTC98] by the US Fair Trade Commission (FTC) showed that very few of the US websites were fully compliant with these principles, which to some degree may indicate that the FIP principles are difficult to apply even to well understood domains such as the web, and are presumably at least equally difficult to generalise to a new domain such as privacy aware applications in distributed systems.

The lack of practical support for incorporating privacy solutions into applications has motivated various research on adapting and extending FIP principles to establish more detailed and applicable design frameworks [Bellotti93, Langheinrich01, Jiang02a, Hong04b, Iachello05] (section 2.5.5) for developing privacy aware applications. For instance, Bellotti and Sellen's conceptual design framework [Bellotti93] emphasised incorporating appropriate control and feedback mechanisms into the following four aspects: capture, construction, accessibility, and purpose. Langheinrich [Langheinrich01] extended the FIP principles and proposed six guidelines for developing privacy-aware applications within UbiComp environments. Jiang et al. developed the Approximate Information Flow (AIF) model [Jiang02a] to minimise the asymmetry of information between the data owners on one side and the data collectors and users on the other. These design frameworks provide more practical guidance for designing privacy-aware applications, but they remain analytical tools that address high-level concepts rather than architecture and technical issues in implementation. Jensen et al. [Jensen05] found no evidence of wide adoption of these design frameworks in the literature, and they argued it is because these frameworks failed to provide a robust and replicable procedure of getting from requirements to design and implementation.

Most developers are not privacy experts, and there is a steep learning curve for

adapting design principles or frameworks and incorporating privacy solutions into their own distributed applications. As noticed by Ackerman [Ackerman04], the next level of support for information privacy consists of system architecture and middleware layers that facilitate the construction of privacy-aware applications.Middleware is often engineered to be reusable and well-architectured software systems [Jacobson97] that provides support for designing stable system structure and implementing system components. A single middleware can be employed to develop multiple applications, and most of those applications are going to be built after the implementation of the middleware [Gamma95]. For example, the Confab toolkit was employed to develop several new privacy aware applications, including a location-enhanced instant messenger, a location-enhanced web proxy, and an emergency response service [Hong04a].

Middleware facilitates the design and implementation of applications by offering developers a simple and consistent programming environment and masking low-level technical details that would require expertise in a specialist area [Coulouris01]. Since the concept of adaptive privacy management is relatively new, we conceive that it would be difficult for developers to adopt its concepts quickly and incorporate it into their own applications directly. To facilitate the engineering of privacy aware distributed applications using our adaptive approach, we *will provide a middleware that supports adaptive privacy management, allowing it to be more easily adopted into the developers' own applications*. In particular, we will design and implement a middleware platform that embodies the principles of adaptive privacy management proposed in the previous chapter. The middleware will provide a set of programming interfaces for developing adaptive privacy aware applications and masks underlying details of handling privacy related interactions where possible.

4.4.2 The Flexibility of the Middleware

System architecture and middleware support for information privacy is still the subject of ongoing debate and experimentation. As discussed in section 2.5.5, existing system architecture and middleware support (e.g., the pawS architecture, the Houdini framework, the Confab toolkit, etc) are often incompatible or even contradict one another. Therefore, developers have to decide on adopting a specific architecture and middleware when designing and implementing their privacy aware applications. Moreover, a particular middleware provides support for developing applications in certain styles,

and to a large extent, has a significant impact on the features of the applications built using it [Edwards03]. As identified by Jensen et al. [Jensen05], the most important shortcoming for existing approaches is that *"they imply the existence, or the desirability of seeking a universally satisfactory solution"*. Based on Palen and Dourish's view of privacy management as a highly dynamic process of boundary negotiation, Jensen et al. argued that it would be almost impossible to provide a universally accepted solution for privacy management [Jensen05]. Followed this observation, we claim that a fixed middleware architecture *cannot accommodate* the need for providing support for adaptive privacy management, and therefore our middleware *has to be flexible so that it can be configurable and reconfigurable to meet the requirements for developers in different problem domains*.

As Edwards et al. [Edwards03] concluded, previous research [Bass01, Gamma95, Beck04] explicitly acknowledges that defining end-user requirements for applications a priori is practically impossible. Based on this observation, we speculate that it is very difficult to decide in advance the optimum behaviour of a privacy aware distributed application to meet end users' requirements, and the flexibility of the middleware will facilitate developers in fine-tuning the behaviour of the application as well as the middleware during the design and implementation phases. Such flexibility is mandated by adaptive privacy management itself, as we need to support the whole spectrum of adaptation for (e.g.) levels of user intrusiveness (adaptation with and without user control); therefore, we require our middleware to be flexible so that it can be configured to meet the user involvement requirement (**R4**) for a particular problem domain.

In addition, it is highly desirable for developers to be able to integrate different plug-ins (in the forms of algorithms or policies) into the middleware in order to customise or parameterise tasks involved in the privacy management process for their own purposes. For example, there are many methods for automating or supporting the privacy management process [Hull03, Eldin04, Dragovic05a, Henricksen05], e.g., neural networks, Bayesian networks, fuzzy logic, privacy preferences suggestion, etc. For applications that employ privacy rules to automate users' privacy decisions (such as ours), there has been a series of methods for detecting and resolving rule conflicts [Dunlop03], e.g., specific overrides general, assigning explicit priorities to rules, most recent rules have precedence, etc. This logic can easily be expressed as a set of replaceable policies.

Such reusable and flexible software infrastructures are well-known in software en-

gineering. For example, the object-oriented paradigm [Meyer88] employs concepts like encapsulation, inheritance, and polymorphism to promote greater flexibility and maintainability in software development. Creational, structural, and behavioural design patterns [Gamma95] were introduced to describe general and repeatable solution for software design by defining known relations between contexts, problems and solutions [Edwards03]. Proponents of *Extreme Programming* [Beck04] explicitly acknowledged that attempting to define all requirements a priori is impractical, and they believe that the software development process should be able to adapt to changes of requirements at any stage of the process. It is well known that architectural problems are difficult to identify before a system is built and costly to address after the fact [Anderson01]. Our choice to make the middleware flexible facilitates modification, extension and maintenance of middleware and applications, which would save significant amount of effort for developers (especially researchers, including ourselves) in improving implemented prototypes and conducting experiments.

4.4.3 Summary

We have argued that design principles and frameworks fail to provide concrete support for developing privacy aware applications, and motivated the need for a flexible middleware platform that supports the development of adaptive privacy aware applications. The flexibility of the middleware not only helps developers to fine-tune the optimum behaviour of adaptive privacy aware applications during design and implementation stages, but also facilitates modification, extension and maintenance of middleware and applications afterwards.

4.5 Architectural Design

In this section, we provide an overview of system architecture for adaptive privacy aware applications that will be developed using the proposed middleware platform. The high-level architecture of our platform is illustrated in figure 4.5. The platform is responsible for processing all privacy-related interactions, and it allows multiple applications to incorporate adaptive privacy management. For any application that shares sensitive personal information between end users, the middleware platform functions as a gateway that allows applications to send private information requests (or privacy requests) and

receive decisions regarding personal information disclosure. The requirements of adaptive privacy management from chapter 3 are realised in the platform, which eliminate the need for re-implementation in each application.

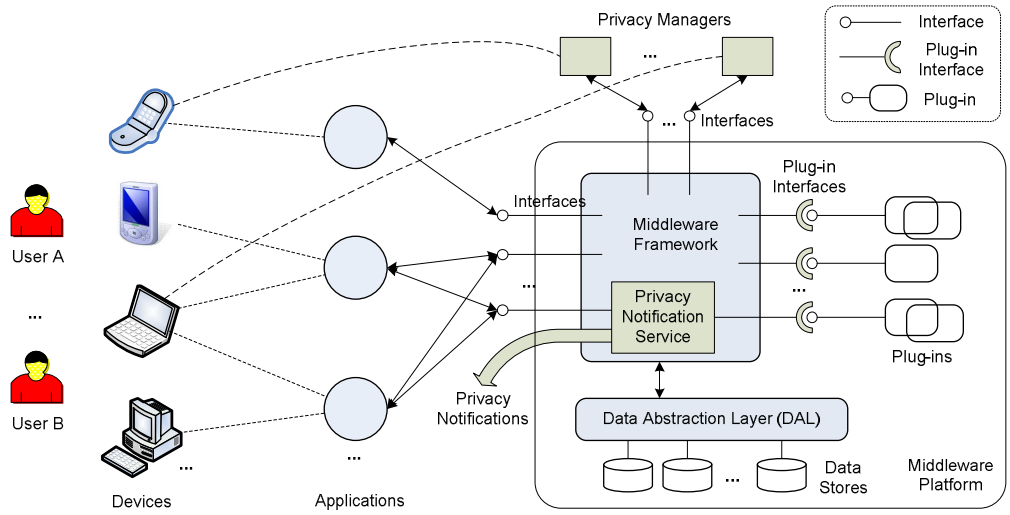


Figure 4.5: Architecture for Supporting Adaptive Privacy Management

Figure 4.5 illustrates the proposed architecture, which allows an end user to interact with privacy aware applications in different modalities from a range of devices (to support **R3**). The middleware platform exposes a set of interfaces for applications to call in order to enable users to share private information according to their privacy requirements. The platform should allow methods on its exposed interfaces to be invoked either locally or remotely, because this location independency facilitates developing applications on system platforms (i.e., operating system and hardware) that are different from the one hosting the middleware. The platform intercepts personal information requests augmented with privacy implications from privacy aware applications, and operates in two basic modes for assisting an end user in making personal information sharing decisions, i.e., *interactive mode* and *delegation mode*.

In contrast to the static approach that requires users to specify their privacy preferences as rules or policies a priori, adaptive privacy management supports zero privacy configuration where no privacy preferences are pre-specified at the start of using the system (**R1**). The platform operates in interactive mode when there are no privacy rules created by a user. In this mode, the middleware platform acts as a gateway for receiving and forwarding information requests and information disclosure decisions from and to applications. When a requester issues an information request to a requestee via a personal information sharing application, the application forwards the request to the

middleware platform by invoking the appropriate public method. The middleware maintains details of the information request in its data stores via the Data Abstraction Layer (DAL), and forwards the request to the requestee using the Privacy Notification Service. Privacy notifications can be dispatched to multiple devices, e.g., email notification to requestee's laptop or desktop, SMS notification to requestee's mobile phone (**R2** and **R3**).

The design of the privacy notification serviceThe design of the privacy notification service should allow two styles of notification, i.e., pull- and push-style. The platform enables adaptive privacy aware applications to pull [Cheverst01] critical privacy events on a periodic basis. Pull-style interface is suitable for constructing monitoring applications (e.g., privacy notifiers) that periodically poll the platform for critical privacy events and generate notifications to users as appropriate. The advantage of pull-style interface is the predictability; because it is applications that initiate queries and thus control the update frequency and can more easily detect when such requests fail. However, pull-style interfaces can be inefficient since they consume computing and networking resources for polling, even when there is no new events to receive (each client creates additional load due to polling at the server). Our platform will also provide push functionality [Cheverst01]for delivering critical privacy events to applications, which is important for promoting users' awareness of system's runtime behaviour concerning privacy in a timely manner. The platform initiates operations of distributing critical privacy events as soon as they happen, so that applications can generate notifications to users without significant delay. Note that push based systems require statefulness at the server and that clients are addressable for notification delivery; this is not always the case in modern networks due to security measures such as firewalls and network address translation (NAT) at the borders of private networks, making hosts not globally addressable.

After receiving privacy notifications, the requestee can make decisions of information disclosure in a given situation using *privacy managers*, which are logical entities enabling the requestee to selectively disclose personal information via any supported interaction modality. In terms of engineering, a privacy manager can be either implemented as a module embedded in privacy aware applications or a separate application that enables end users to manage different types of personal information in one place. A requestee can interact with one or more privacy managers built using the same interfaces exposed by the platform to process privacy requests. The consistency of the requests'

status is ensured because the platform intercepts operations on privacy requests from different privacy managers and changes requests' status accordingly in the central data repository. The requestee can explicitly accept or reject the received request, and in addition he/she can dynamically create a privacy rule for processing the incoming information request as well as future requests with same conditions specified in the rule. Decisions on information disclosure are returned to privacy-aware applications, where actual disclosure of personal information takes place.

In the interactive mode, the middleware platform and an application effectively transform the synchronous operation of sharing personal information into several asynchronous operations, e.g., *issuing a personal information request, notifying requestee of the request, responding to the request, disclosing personal information*, etc. This transformation is necessary because adaptive privacy management requires cooperation between a user and a system and any operation involving end users might take an arbitrary long time to complete. We discuss this issue in more detail when we consider its implementation in the next chapter.

To reduce the intrusiveness of privacy management on the user (**R4**), the platform supports *delegation mode* where incoming information requests can be automatically processed by privacy rules set by the requestee: rules contain specifications of the conditions for processing information requests (e.g., who they are from, what type of information is being requested, etc.) and any restrictions on disclosing the information (e.g., time, location, quality of information, etc.). For example, a user may create a privacy rule that allows his family members to know his location at any time with an accuracy of 100 metres. The restrictions specified in a privacy rule are passed up to the privacy aware application, so that it can impose the restrictions (e.g., change the accuracy of location) before disclosing the information to the requester. Imposing such restrictions are optional and depend on the application specific behaviour. We will discuss rules and rule processing in more detail in section 5.3.4. Privacy rules are created or modified by end users using a privacy manager, and they are maintained in the data store of the platform. In the delegation mode, a privacy rule can be applied to multiple applications for automating privacy decisions. Moreover, keeping privacy rules in a central repository potentially facilitates evolution and generalisation of privacy rules, e.g., general rules can be created or suggested by the system that process requests from different requesters across multiple services.

To support flexibility of design, the middleware platform is to define *plug-in interfaces* allowing the platform to be extended with modules statically or dynamically. We anticipate plug-in interfaces for detecting and resolving privacy rule conflicts to allow experimentation with different conflict resolution algorithms, such as *specific rule overrides general rule, assigning explicit priorities to rules, most recent rules have precedence*, etc. [Dunlop03]. The framework will also include plug-in interface for supporting new methods for distributing privacy event notifications. We explicitly plan to leverage traditional communication channels including email and SMS messages, to deliver notifications to end users. New applications could require additional plug-ins, e.g. an event publisher for Elvin [Segall97]. We discuss this further in the next chapter (section 5.3.1).

To conclude, in this section we presented a high-level design of the key aspects of our middleware platform for supporting adaptive privacy management. The proposed platform enables end users to interactively process privacy requests using different interaction modalities on a variety of devices as befits their personal preferences. Crucially, the platform supports automating privacy decisions on behalf of end users using privacy rules, which can be created and modified as a result of interactions between the system and an end user; enabling a balance to be negotiated between the user and the system for a given situation. Finally, plug-ins can be developed and integrated into the platform to customise the functionality of the system to meet requirements of different problem domains.

4.6 Summary

In this chapter, we have presented the core design decisions for intentional information sharing applications in order that they meet the requirements for adaptive privacy management presented in the previous chapter. Specifically, we proposed notifying users of critical events concerning privacy in a timely way, providing multi-modal and multi-device interactions, supporting contextual decision making within a given situation by presenting users critical factors for privacy, automating privacy decisions using privacy rules, facilitating the management of privacy rules, maintaining status for privacy-related interactions (to allow for human processing as well as system delays introduced by mobility etc.), and providing support for plausible deniability. The re-

mainder of the chapter focused on designing an architecture that incorporates adaptive privacy management into distributed applications. Following a discussion of distributed system architecture and middleware support, we motivated the need for a flexible middleware platform to simplify the development of adaptive privacy management. Finally, we provided a high-level overview of such a middleware platform and presented the description of architectural design for adaptive privacy aware applications that interact with the proposed platform. In the next chapter we present a prototype implementation of the privacy middleware platform as well as some sample applications to illustrate how these can be built using it.

CHAPTER V

Implementation

Contents

5.1 Overview

In this chapter we present the implementation of a prototype of our adaptive privacy management system that includes the middleware platform proposed in the last chapter. We first discuss the challenges associated with incorporating adaptive privacy management into distributed applications using a middleware platform. We then describe the significant engineering details of the prototype system and the supporting middleware. The implementation of a distributed client application that uses the platform API and plug-in architecture to customise the behaviour of the platform is then described. A key contribution of this chapter, is the API methods and plug-in interfaces that enable the creation of adaptive privacy aware applications. Finally, we motivate the need for privacy aware location sharing applications, and present the implemention of such a prototype application to more concretely illustrate how these APIs are used. This application form a core component of the user centred evaluation of the adaptive approach for privacy in the next chapter.

5.2 Adding Privacy to Distributed Applications

Adaptive privacy management aims to empower an end user to collaborate with a system to dynamically adjust the level of openness close to the user's desired level for varying situations in distributed environments. The requirements for designing the adaptive privacy management as discussed in chapter 3 are:

- Adaptive Privacy Balance and Evolution of Privacy Preference (**R1**);
- Awareness of System Behaviour Concerning Privacy (**R2**);
- Convenient and Timely Access to Privacy Controls (**R3**);
- Balance between Privacy and User Intrusiveness (**R4**); and
- Accountability for Privacy-related Behaviour (**R5**).

The design of the middleware platform presented in chapter 4 was directly derived from this set of requirements. In building privacy-aware applications using the middleware platform, we assume that developers are actively involved in incorporating adaptive privacy management into applications. Since privacy-aware applications in different

problem domains may require different privacy solutions, the middleware platform was designed to be configurable so that developers can customise it to meet specific needs for their domains. The designed architecture (refer to section 4.5)enables privacy-aware applications to externalise the privacy-related decisions to the middleware platform, which can operate either in the delegation mode to autonomously accept and deny private information requests or operates in the interactive mode to relay requests and responses between users.

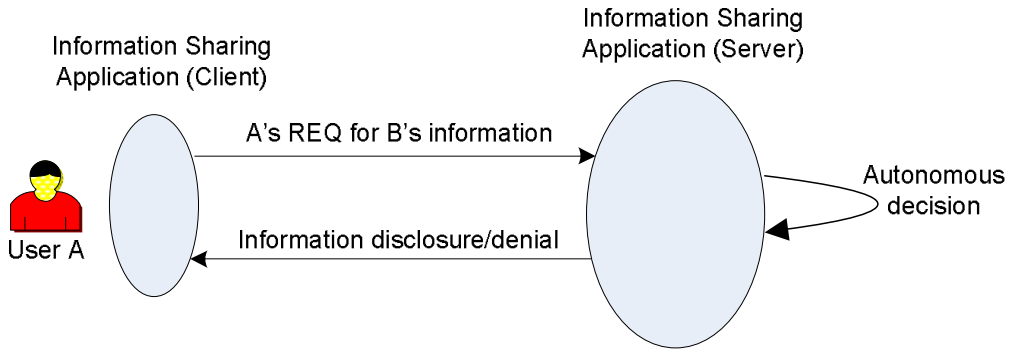


Figure 5.1: Synchronous operation for information sharing

Existing distributed applications share personal information based on hard-coded behaviour built into the application [Silverman05] or pre-specified user preferences [Langheinrich02b], and therefore operations for sharing information are mostly synchronous because applications can make autonomous decisions and return results of such information disclosures immediately (as illustrated in figure 5.1). To incorporate adaptive privacy management in distributed applications, the middleware platform has to intercept these synchronous operations and make privacy decisions potentially collaboratively with an end user. Therefore, interactions between the application and the middleware need to be augmented with the privacy context needed for reaching the disclosure decision; e.g., the identify of information requester, time of the request, the purpose of the request, etc. (as discussed in section 4.2.1).While operating in interactive mode where the user is involved in privacy decisions the need for user interactions may introduce an unbounded delay for the underlying information sharing operations queued in the middleware: a user may receive a notification but may not see or wish to process the request immediately. Such delays break the timeliness assumptions of synchronous middleware (e.g., synchronous RPC) leading to unwanted timeouts and impacting the underlying adaptive mechanisms such as TCP's congestion control. Furthermore, at the application layer, these delays could block applications unpredictably and certainly break the developers reliability assumptions (RPCs encourage developers to think of

distributed operations as equivalent to local function calls, with attendant expectations on delay and reliability).

To ameliorate these problems, we deliberately partition synchronous operations into a sequence of asynchronous interactions as illustrated in figure 5.2:

1. the platform receives a privacy request with privacy implications (I1),
2. the platform sends a notification of receiving the request to the requestee (I2),
3. the platform receives a response to the request from the requestee (I3), and
4. the platform sends a notification of the response to the requester (I4).

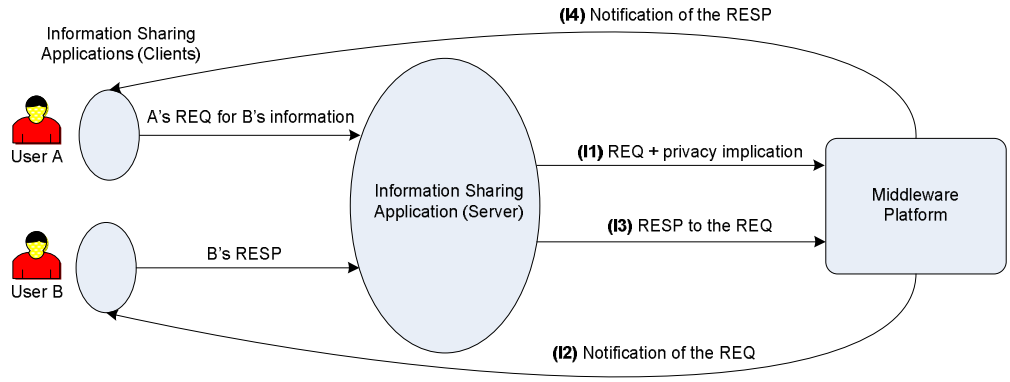


Figure 5.2: Transforming synchronous operation into four interactions

Similar to deferred synchronous RPCs, each asynchronous interaction returns immediately and therefore an information sharing operation should be designed not to block an arbitrary interval before receiving a reply. Note that the semantics of the operation have not changed; it will still take the same time overall to complete the information request, however, operations are now asynchronous and stateful (their completion status can be inspected) — our intention is that application programmers should explicity design for this asynchrony and reflect the ongoing state of interactions up to end users in appropriate forms. The state of ongoing operations have to be maintained in data stores within the platform, because an information sharing operation can be interrupted at any one of the aforementioned asynchronous interactions, e.g., notification of the request has been sent to the requestee but no response has been received by the platform via any of the possible interfaces. Applications can query the state of the operation using either the pull or push styles offered by the platform. In the next few sections we describe the implementation details of the middleware, explaining the above mechanism for transforming synchronous operations into asynchronous operations in more detail.

5.3 Implementation of the Prototype System

The adaptive privacy management system prototype (i.e., an adaptive privacy manager) follows the system architecture and design decisions proposed in the previous chapter. The aim of this prototype implementation is to:

1. Illustrate that the design we have proposed presents a feasible system that can be implemented.
2. Evaluate whether the system resulting from this design supports adaptive privacy management for privacy-aware distributed applications.
3. Investigate the strengths and/or weaknesses of our design.

The implemented prototype of the adaptive privacy manager (for brevity: *privacy manager*) is engineered using the API offered by our privacy middleware. The prototype system consists of a set of services that are implemented using a combination of core middleware components and distributed client-side, applications as illustrated by figure 5.3. In particular, the prototype system exposes the following core services for end users to achieve the adaptive privacy management:

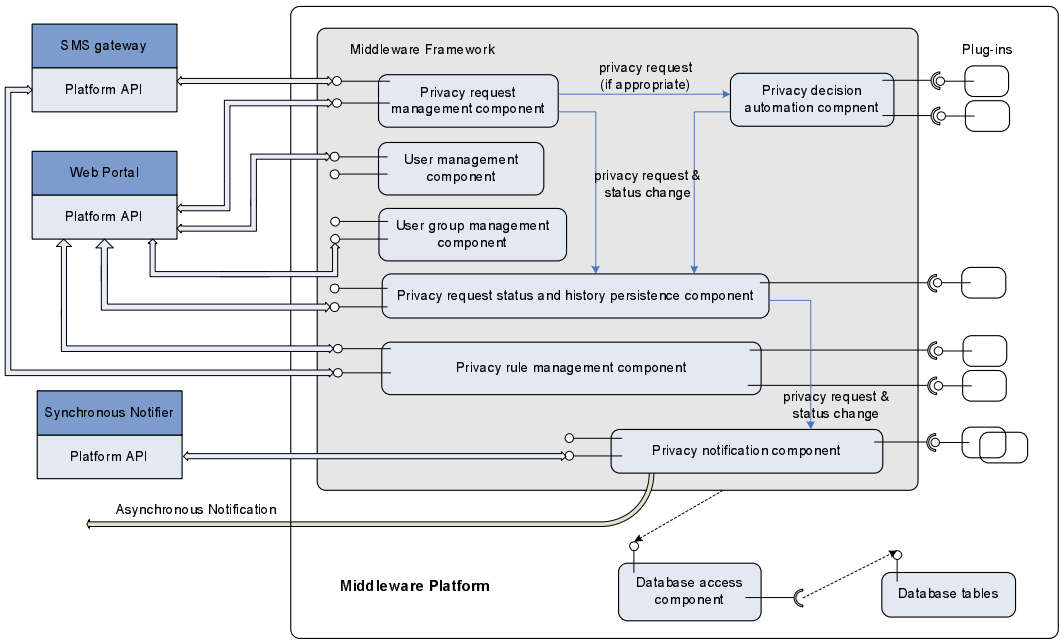


Figure 5.3: System Component Overview

User management service (S1): This service exposes a set of operations through the web portal that enables users to add, delete, and modify their personal information in

the system. The system identifies registered users using a set of credentials (username, password and nickname in the current system), which are prerequisites for enabling other core services, e.g., the privacy notification service, that are based on user identity.

Privacy request management service (S2): This service enables end users to make and process private information requests (e.g., sending, accepting and rejecting requests) using a web-based interface (via the web portal) or SMS messages (via the SMS gateway). This service forwards privacy requests to the privacy decision automation service as appropriate.

Privacy decision automation service (S3): This service automatically processes incoming privacy requests on behalf of a requestee by selecting and applying a most appropriate privacy rule belonging to the requestee. Plug-in interfaces for selecting and applying privacy rules are defined for customising this service.

Privacy interaction persistence service (S4): This service receives the status of privacy requests after they are processed by the privacy request management service or the privacy decision automation service. This service maintains status and detailed information of privacy requests within the system by making them persistent in the underlying database. This service exposes user interface through the web portal that enables user to query for status and history information of privacy requests.

Privacy notification service (S5): This service provides both synchronous and asynchronous notifications of events affecting the user's privacy (e.g., status change events for privacy requests) to promote end user' awareness of privacy. Synchronous privacy notifiers poll the middleware platform periodically using the pull operations and display privacy notifications via their own user interface (e.g., a web-based popup window or system tray notification). Asynchronous privacy notifiers are implemented as plug-ins for the middleware framework that push events to the user. In our current implementation push notifiers are incorporated for email and SMS messaging.

Privacy rule management service (S6): This service offers web-based and SMS message-based interfaces that enable end users to create, delete, and modify their privacy rules. This service is a prerequisite for enabling the automatic privacy request processing. In addition, this service supports creation of plug-ins to customise and extend its behaviour.

User group management service (S7): This service exposes a web-based interface that allows end users to manage user groups and group membership. This service enables

end users to categorise information requesters into a manageable sets with their own associated privacy rules, in order to simplify rule and request management.

The prototype implementation of the middleware platform (i.e., the middleware framework and plug-ins) was engineered in C# (in approximately 6000 lines of code for the platform) using the Microsoft .NET Framework 2.0 [Michaelis06]. It can be hosted on Common Language Runtimes (CLRs) on different versions of Microsoft Windows operating system. The persistent storage required for the operation of the platform are implemented as tables in a relational database using the open-source MySQL 5.0 DBMS. The data access component is implemented using native C# APIs provided by the ADO.NET Driver for MySQL 5.0, which should allow portability to other relational database servers. The web portal, the web-based synchronous notifier, and the SMS gateway are implemented as distributed client applications that interact with the platform using its exposed API methods via .NET Remoting. The web portal and synchronous notifier were developed using C# ASP .NET, and the web pages are hosted on a Microsoft Internet Information Server (IIS) 6.0 on a Windows 2003 Server.

We have provided an overview of the prototype implementation. In the next part of the chapter, we present a detailed description of the prototype system including the middleware platform and the distributed applications. The discussion will focus on the aspects of the system that meet the requirements of the adaptive privacy management (as defined in section 3.3).

5.3.1 Promoting Privacy Awareness via Notification

Adaptive privacy management requires promoting user awareness of privacy, and the implemented privacy notification service (S4) provides end users with both synchronous and asynchronous notifications about critical privacy events in a timely manner (note that in this context we refer to timely on a human rather than a computational system timescale, i.e. O(seconds)). A prerequisite for implementing the privacy notification service is to identify individual users of the system, so that appropriate privacy events can be generated and transferred to corresponding users. When a user signs up for an account of the system, the user management service (S1) assigns them a unique identity (a unique 32-bit integer), generates a username and password pair to identify the user on the web portal, and registers his/her mobile phone number for use with the SMS gateway. The account details of a registered user are saved in the data store, and they

can be retrieved as a `User` object (illustrated in Figure 5.4) via the database access component.

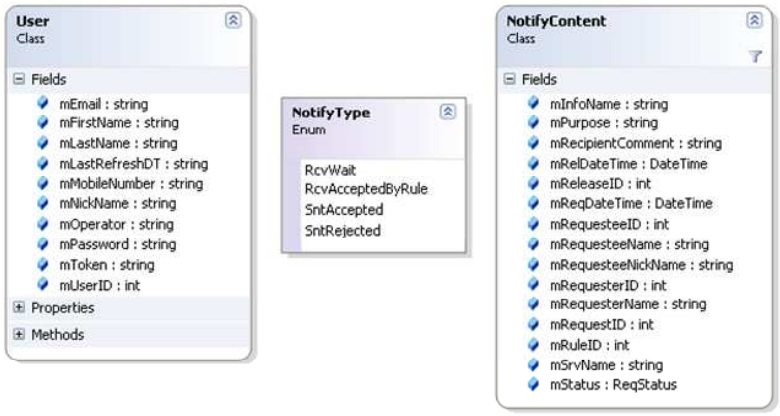


Figure 5.4: Data Structures Required for Notification Service

In the current implementation, the middleware framework supports retrieving of four types of privacy events as specified by the `NotifyType` enumeration. `RcvWait` requests are privacy requests received but waiting for a requestee to process them, in this case notifications are generated to alert them of the incoming requests (i.e., I2 in figure 5.2). `RcvAcceptedByRule` events are requests automatically accepted by a requestee's privacy rule, currently notifications are generated to promote awareness of the information disclosure and the effect of the user's rules. Events that are `SntAccepted` and `SntRejected` indicate requests that were either accepted or rejected, so notifications can be generated to keep the requester aware of the status changes of previously sent requests and optionally private information received from the requestee (i.e., I2 in figure 5.2).

As discussed earlier in section 5.3, the privacy notification component exposes a pull-style interface for developing synchronous notifiers and a plug-in interface for implementing asynchronous notifiers (see figure 5.5).

API method for applications to poll new privacy events from the middleware:

```
DataSet GetNewRequestEvents(
    int userID, NotifyType type, DateTime lastDT);
```

Plug-in interface for pushing or transporting privacy events from the middleware:

```
(P5) delegate void NotifyHandler(
       User u, NotifyType nType, NotifyContent nContent);
```

Figure 5.5: API method and plug-in interface for notification

The `GetNewRequestEvents` method allows clients to poll the platform periodically

to retrieve status change events of a certain type. This pull-style interface is modelled on email style protocol interactions in that it does not require synchronous notifiers to keep a permanent TCP connection open, supporting periodic polling behaviour, e.g., from web pages, or partially connected mobile devices. We have chosen not to use server push style notifications by default, because in modern Internet environments the device hosting a synchronous privacy notifier application may be behind a NAT router or have no permanent IP address (e.g. a mobile host). The `GetNewRequestEvents` method requires the identity of a user, the type of privacy event (in `NotifyType` enumeration), and a timestamp (in `DateTime` class) as input parameters. The call returns any new privacy events since the specified timestamp as a `DataSet` object containing data elements (effectively a *join* of the 'request' and 'releaseinfo' tables, see figure 5.16). DataSet is provided by the .NET framework to hold results of database queries and it can be easily converted to other types or serialised for transportation across the network.

A synchronous privacy notifier application (e.g., like the well-known Gmail notifier) can be engineered by invoking the GetNewRequestEvents method periodically. Developers can adjust the poll frequency according to their requirements. The prototype system implements a web-based synchronous notifier that polls for new privacy events from the middleware platform every five minutes and generates MSN-style popup alert windows when privacy events are found. As illustrated in figure 5.6, a popup alert window shows the number of privacy events of a same type received, and provides a hyperlink that directs a user to more detailed information about the request, e.g., the requester or requestee, time of the request and type of information requested. Different coloured popup windows are used to highlight the different types of privacy events, and multiple popup windows may be shown at the same time if different types of privacy events are retrieved from the platform. Synchronous privacy notifiers are only really useful when end users are working on their desktops or laptops, and the web notifier requires that the user has the main page of the web portal open for the popup windows to be visible. The web portal pages automatically return to this home screen if no user interaction happens within five minutes.

In order that users can receive privacy notifications asynchronously for occasions where they are away from their computer or offline, the push-style plug-in interface `NotifyHandler` (**P5** in figure 5.5) allows developers to implement asynchronous privacy notifiers as plug-ins. The plug-in interface takes three input parameters: a `User` object containing personal information (email address or mobile phone number) required

Figure 5.6: MSN messenger style privacy alerts

for identifying a registered user, the type of privacy event (`NotifyType` enumeration), and a `NotifyContent` object that consists of all the information required to generating the privacy notification, e.g., name of requested information,timestamp, service for disclosing the information,timestamp of information disclosure and the private information itself. Specifying a plug-in interface for generating privacy notifications separates the concerns of *what privacy events should be generated* from *how privacy events should be distributed to the user*.

The prototype system includes two instances of the above plug-in chained together and bound to the middleware framework. These are invoked sequentially at runtime to deliver privacy notifications to users: specifically, an email notifier sends privacy notifications as an HTML email and an SMS notifier sends a more concise version to the user as an SMS text message. The email contains both the details of the event and hyperlinks to allow the user to manage the request, consult their privacy history, manage rules, etc. The SMS notifier communicates with the SMS gateway using a proprietary TCP protocol.

5.3.2 Support for Making Privacy Decisions in Context

As discussed in section 3.3.1, adaptive privacy management requires that users can adjust the balance between openness and closedness depending on situation. The implemented privacy management service (**S2**) provides support for end users to make privacy decisions within context of a private information request. The private information request abstraction is frequently used in the prototype system, and it was modelled internally as a `PrivInfoReq` class, shown in figure 5.7. We base the content of PrivInfoReq on the critical factors for privacy decisions we have identified in section 4.2.1, i.e. the identity of the requester (`mRequesterID`), the identity of the reques-

tee (`mRequesteeID`), required for processing or forwarding a request, `mReqDTStr` for recording the time the request was sent and the `mPurpose` field for specifying any contextual information the requester wishes associated with the request to explain their actions. `PrivInfoReq` contains additional fields that allow applications to specify a name to identify the type of information being requested (e.g. location), the name of the service/ information source to query (i.e. FollowUs in our application), the time of the request, and the expiration time for the request. Finally, the class also contains the `mStatus` field for maintaining the status of the request (a `ReqStatus enumeration`). Privacy requests (`PrivInfoReq` objects) are persistent; they are assigned unique identifiers and stored in the 'request' database table.

Figure 5.7: Representation of a private information request (PrivInfoReq class)

As we have discussed in section 5.2, synchronous application operations must be transformed into a number of asynchronous interactions. The middleware platform exposes the `IssueInfoReq` method to allow privacy aware applications to forward requests augmented with privacy metadata to the platform. The `IssueInfoReq` method takes a `PrivInfoReq` object as an input parameter, and returns the status of the request and optionally a `PrivPref` output parameter containing a privacy rule.

```
public  ReqStatus IssueInfoReq
          (PrivInfoReq req,    out  PrivPref pref);
```

The internal operations for processing a privacy request are illustrated in figure 5.8; where plug-ins are shown as rectangles with dotted lines. The middleware platform receives a privacy request encoded in a `PrivInfoReq` object from a client application, and checks if the rule selection handler or plug-in (**P1**) is registered. The handler is invoked to select the most applicable rule that can process the request. If a handler is not registered, the platform checks for the rule creation handler (**P2**) to dynamically generate a rule based on the this request. The rule evaluation handler (**P3**) is invoked to process

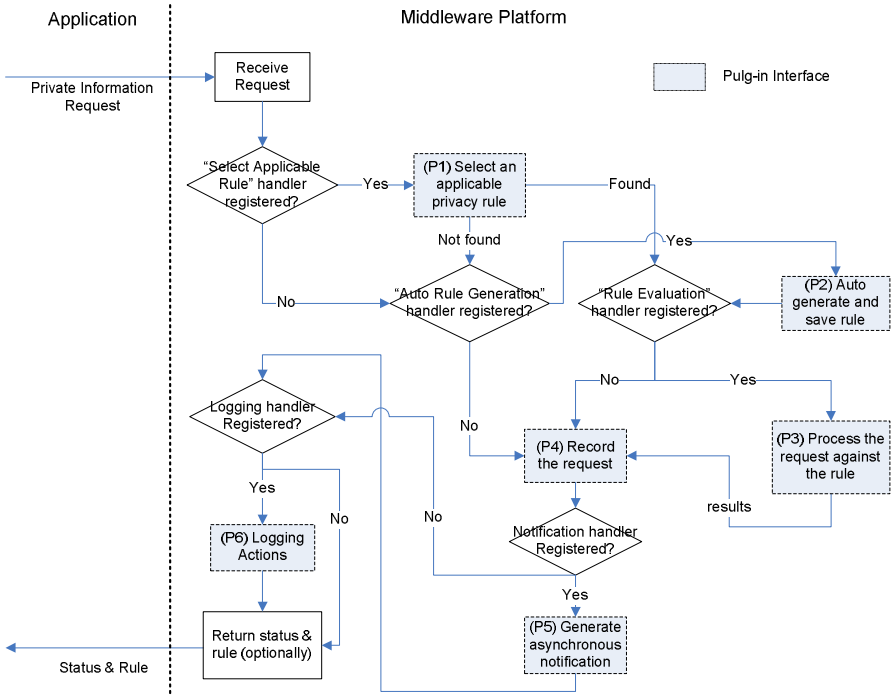


Figure 5.8: Internal sequence of operations for processing a private information request

the request using the rule and the status of the request is updated. Otherwise, the status of the request remains as *waiting for processing* until the user either actions the request or it expires (see plausible deniability, section 4.2.6). The request persistence handler (**P4**) is invoked to store the request in the underlying data store, and any asynchronous notification handlers (**P5**) are invoked to generate notifications to the requestee and/or requester as appropriate. The framework invokes the logging handler (**P6**) to record the interaction, finally returning the status of the request and optionally a privacy rule encoded as a `PrivPref` object to the client application. The returned privacy rule may contain a list of filters to be applied at the client-side (e.g., a granularity filter for location information) that the application should apply on the private information before it is disclosed to the user.

```
public void AcceptRequestsManually(int userID,
    IntegerList reqIDs, string details, string extraInfo);
public void RejectRequestsManually(
    int userID, IntegerList reqIDs, string extraInfo);
public bool AcceptRequestManually(
    int userID, int reqID, string details, string extraInfo);
public bool RejectRequestManually(
    int userID, int reqID, string extraInfo);
```

Figure 5.9: API methods for accepting and rejecting privacy request(s)

When a user is notified of having receiving a privacy request that is not processed

by one of her privacy rules, she can process it interactively (**I3** in figure 5.2) using the privacy request management service (**S2**). The API includes the methods shown in figure 5.9 for accepting and rejecting one or more requests. All the methods require the requestee's identity (because only the requestee has the privilege to process requests sent to him), and the identifiers of one or more requests to be processed. To accept a request, details of the request type are passed as a string parameter, and application developers can choose the format the information is returned in, e.g., encoded in XML format.The `extraInfo` field is appended onto the reply back to the requester to provide context for the request. We describe the web portal and SMS gateway that enable users to manage privacy requests using multi-modal interactions on multiple devices in the next section.

5.3.3 Enabling Multi-modal and Multi-device Interaction

One of the key design decisions discussed in section 4.5 is the provision of multi-model and multi-device interactions for privacy.

The privacy notification service enables multi-modal and multi-device notification: the synchronous notifier displays critical privacy events in web-based popup alert windows on PCs and laptops, and asynchronous notifiers leverage existing communication channels to distribute notifications of events via email and SMS. To allow convenient and timely access to privacy controls, end users need to be able to interactively respond to such notifications using either modality. The prototype system implements a web portal that provides a web based user interface accessible using standard web browsers and a SMS gateway accepts a limited range of text commands for managing privacy interactions.

Architecturally, the web portal and the SMS gateway are both implemented as client applications that remotely invoke privacy request management API methods exposed by the platform using .NET Remoting. Since the middleware platform maintains status and history of privacy requests centrally in its database tables, multiple applications can be implemented on different platforms that provide various interfaces to the user.

We chose a web based interface so that users can access privacy controls from any operating system and from any device that supports web browsing. The web portal retrieves the identity of a user after authenticating him using a username and password (we

embed an authentication token into the URLs in the privacy notifications to allow users to skip the login step in most interactions with the system). These credentials are used directly by the API methods for processing privacy requests, e.g., issuing, accepting, or rejecting requests. The web portal displays received or sent requests in table-like format (see section 5.4.4),that allows end users to see the status of all outstanding requests.

The SMS gateway uses the mobile phone number of the user as a form of identification: this is a compromise, it offers basic authentication with little user effort, however, we would like to note that there are circumstances where phones do not belong to the individual for, e.g. cultural or financial considerations [Bell06] (and even in one case of appropriation and use in our user study, see section 6.4.2). The SMS gateway accepts specially formatted text commands in SMS messages to control privacy interactions, e.g., making new information requests, accepting and rejecting incoming requests and creating simple rules. It parses received SMS messages to obtain the identity of a privacy request to be processed (the request ID is embedded in the message). Heuristically, we select the most recent non-expired request to the user, if no request ID is included in their text message (we encourage users to quote the request when replying). In our current implementation, the interface to the SMS hardware relies on a gateway to legacy SMS hardware and software running on a Linux platform. We extended the existing SMS software by integrating a new module written in Perl that handles text commands in SMS messages related to the privacy management system, and implemented a TCP server that parses our specially formatted TCP messages received from the extended SMS message handling modules and invokes appropriate API methods on our platform.

5.3.4 Automating Privacy Decisions using Privacy Rules

As we've discussed, the adaptive privacy management system can automate privacy decisions using privacy rules, in order to reduce user involvement in privacy management. This is achieved using the privacy decision automation service (**S3**), user interfaces provided by the web portal and SMS gateway for creating (and in the web case managing) the rules stored in the middleware platform.

As illustrated in figure 5.10, a privacy request received by the middleware platform can be automatically processed only if an applicable privacy rule already exists. The middleware framework defines the `FindPrefHandler` plug-in interface (**P1** in figure 5.10) for developing new algorithms for rule selection. The `FindPrefHandler` plug-in

```
(P1) delegate PrivPref FindPrefHandler(PrivInfoReq req);
(P2) delegate int AutoGenSavePrefHandler(PrivInfoReq req);
(P3) delegate ReqStatus EvaluateReqHandler(
              PrivInfoReq req, ref PrivPref pref);
```

Figure 5.10: Plug-in interfaces for customising the privacy decision automation service

interface requires a `PrivInfoReq` object as input and returns a `PrivPref` object. The `PrivPref` class implements an in-memory representation of a privacy rule, and its attributes are illustrated in figure 5.11. The `mAccept` field specifies whether the privacy rule is positive or negative (i.e., accepting or rejecting). The intended information recipient can be an individual user identified by `mRequesterID` or a group of registered users identified by `mReqGroupID`. As before, the information type and service name are used during the matching process. A privacy rule contains filters specifying contextual conditions in which that rule should be applied, and they are represented as fields in `PrivPref` class and declared in `DateFilter`, `TimeFilter`, and `LocGranFilter` types, respectively. All privacy rules are maintained in the 'rule' table of the database, and they can be accessed via the database access component and retrieved into runtime memory as `PrivPref` objects.

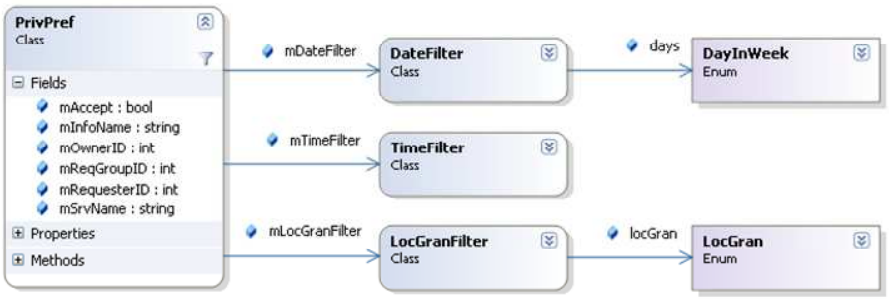


Figure 5.11: Representation of a privacy rule (PrivPref class)

The algorithm implemented as our current `FindPrefHandler` plug-in searches for privacy rules that contain the same properties as the received request: the identity of the requester, the information type, and the service name. In addition, the algorithm validates the server-side filter conditions contained in the rule, i.e., the date and time filters. A date filter specifies the days in the week that the rule should be applied. The time filter specifies active time span during which the rule should be applied. A privacy rule is applicable only if all conditions of server-side filters are met.

Since a request may match multiple privacy rules, the plug-in is responsible for resolving such conflicts. In the current implementation, we classify privacy rules into three categories (i.e., rules for processing requests from an individual user, rules for

processing requests from users in a user group, and rules for processing requests from all registered users). The algorithm assigns a descending priority for each of these categories; i.e., the algorithm searches for rules for individuals before rules for user groups, and finally the rules applying to all users. In addition, the algorithm gives higher priority to negative rules than positive ones in the same category. For example, if one positive and one negative privacy rules exist for processing requests from the individual *Alice* (for the same information disclosed by the same service), the algorithm selects the negative one. The user can have one positive and one negative rule for every individual and group (this is a limitation of the current implementation).

The middleware framework defines the `AutoGenSavePrefHandler` plug-in interface (**P2**) for developing plug-ins that generate new rules to process incoming requests. The plug-in interface requires a `PrivInfoReq` object as input and returns a `PrivPref` object. This plug-in interface is not instantiated for the current implementation, because we argue that adaptive privacy management requires the system to collaborate *with users*, not replace them with automatic processing. We have included this interface to allow developers to experiment with other preference learning mechanisms and Artificial Intelligence (AI) algorithms [Viappiani02, Pu06] in their applications. The middleware framework defines the `EvaluateReqHandler` plug-in interface (**P3**) for introducing new rule evaluation algorithms. The interface requires a `PrivInfoReq` object as input parameter and a `PrivPref` object as an input-output parameter, and returns the status of the request in `ReqStatus` type. The current implementation of the plug-in is quite straight forward: using the selected privacy rule, it either accepts or rejects the request by setting the request to appropriate status, e.g., `AcceptedByRule` or `RejectedByRule`, and returns corresponding status.

5.3.5 Balancing User Intrusiveness and Privacy Rule Management

To enable automatic privacy request processing, the privacy rule management service (**S6**) is required to allow users to create, delete, and modify rules stored in the system. The service provides users both web-based and SMS message-based interfaces, and it is engineered using API methods shown in figure 5.12. The privacy management service can be customised to allow experimentation with different levels of user involvement.

These methods allow applications to manage privacy rules. To save a privacy rule in the underlying database, an application needs to invoke the `RecordPrivPref` method

```
GenRuleStatus RecordPrivPref(int userID, ref PrivPref pref);
GenRuleStatus UpdatePrivPref(int userID, ref PrivPref pref);
void DeletePrivacyRule(int userID, int prefID);
GenPrefStatus AddPrivPrefForReq(int reqID, out PrivPref pref);
bool ReplyAlways(int userID, int reqID, string details);
bool ReplyNever(int userID, int reqID);
```

Figure 5.12: API methods for managing privacy rules

passing in the identity of a user (i.e., rule owner) and a `PrivPref` object containing the relevant attributes of the privacy rule. The method returns a status code and the client application can retrieve the identifier of the privacy rule saved in the database from the `PrivPref` object passed as an in-out parameter. The `UpdatePrivPref` method modifies an existing privacy rule works in a similar way, with the exception that the PrivPref objectcontains both the identifier of the privacy rule to modify as well as the attributes to update. To delete a rule, an application invokes `DeletePrivacyRule`, passing the user identity and rule identifier as parameters. These operations underpin the rule management user interfaces in the web portal. It also can provide a colloquial language translation of the rules to assist users in understanding how the rules will be applied.

```
int AddUserGroup(int userID, string groupName);
void DeleteUserGroup(int userID, int groupID);
string AddUserToGroup(int userID, int ruID, int gID);
string DeleteUserFromGroup(int userID, int ruID);
```

Figure 5.13: API methods for managing user groups and membership information

The API above (figure 5.13) lists the methods for managing user groups and group membership. `AddUserGroup` creates a user group for the user identified by `userID` parameter with arbitrary name (i.e., `groupName`), and `DeleteUserGroup` removes a user group from the system. In addition, an application can invoke the `AddUserToGroup` to add a user (identified by `ruID`) to an existing user group (`gID`). The `DeleteUserFromGroup` method removes a user (`ruID`) from a user group. A user can only belong to one user group — this is a limitation of our current implementation. A set of web pages are provided by the web portal for managing user groups, and hyperlinks to those pages are provided to encourage users to add and arrange groups while they are managing their privacy rules.

To make it easier for users to create rules, the platform provides methods (see figure 5.15) that developers can use to obtain suggested defaults for the rule's optional attributes. The operation of `AddPrivPrefForReq` is shown in figure 5.14: the plat-

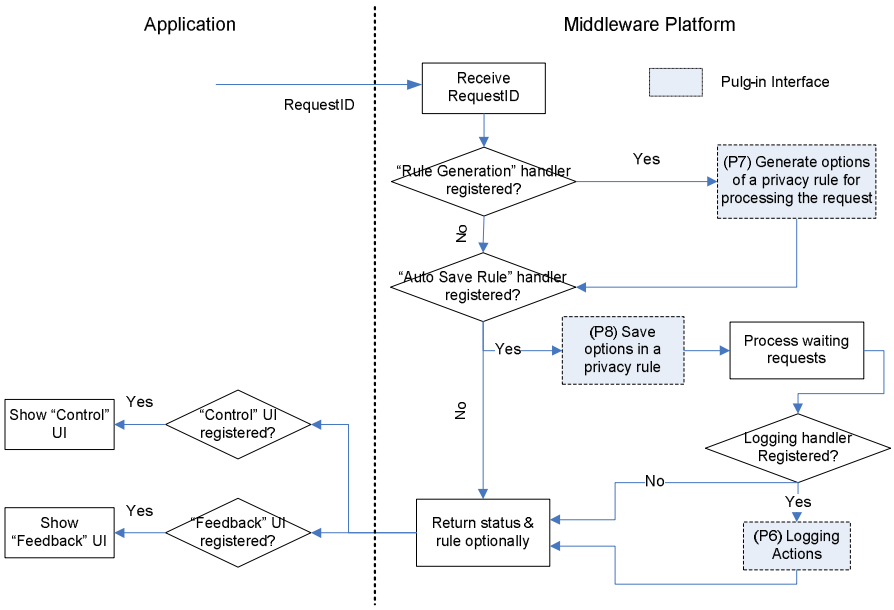


Figure 5.14: Internal operation for AddPrivPrefForReq method

form takes the request corresponding to the unique identifier (`reqID`) and invokes the registered rule option generation handler (**P7**) and uses this rule as the basis for its suggestions (e.g. if the requester is in the 'coworkers' group, it might suggest 'weekdays, 9:00-17:00, city-level' as defaults). If the rule is approved, the option storage handler (**P8**) is invoked to store the generated options in the privacy rule. The rule is then applied to any outstanding requests waiting for processing. Otherwise, the rule options are generated but not saved. Next, the middleware framework invokes the logging handler (**P6**) to record the interaction. In both cases, the status of creating privacy rule (`GenPrefStatus` enumeration type) and generated rule options (`PrivPref` class) are returned to the client application. If rule options were not automatically saved, the application could ask the user to review and adjust the suggested options as necessary. Otherwise, an application can show a feedback interface to notify users that a new privacy rule has been generated and saved in the platform.

```
(P7) delegate PrivPref GeneratePrefHandler(int reqID);
(P8) delegate bool AutoRecordPrefHandler(
            PrivPref pref, out int prefID);
```

Figure 5.15: Plug-in interfaces for customising the privacy rule management service

We have implemented three plug-ins for generating rule preferences as we have described. The `GeneratePrefHandler` plug-in interface requires the unique identifier of a request as input and returns a `PrivPref` object containing the suggested options.

The first plug-in generates rule options for a given requester based on existing rules for the same type of information and service applied to other requesters. The second plug-in identifies the user group that the requester belongs to, and generates options based on pre-specified rule options associated with the group. The third plug-in generates pre-defined default options. The platform invokes the plug-ins sequentially and accepts the first none-empty set of privacy options returned. Our prototype system does not implement the `AutoRecordPrefHandler` plug-in for automatically saving generated privacy options as a rule in the underlying database, because our ethos is to involve users in these decisions. The `AutoRecordPrefHandler` plug-in interface requires a `PrivPref` object containing the suggested options as the input parameter, and returns the identifier of the privacy rule recorded in the underlying data store.

In summary, the current implementation of the web portal provides interfaces that dynamically suggest privacy rule options for processing a received request and requires explicit user confirmation to store it permanently. The rule creation approach is less intrusive, because users do not necessarily have to modify the suggested options for the new privacy rule.

The remaining two methods in figure 5.12 are shortcuts for generating special privacy rules. `ReplyAlways` generates an 'accepting' rule that have following properties the same as the received request: the identity of the requester, the type of information requested, and the service disclosing the information. `ReplyNever` generates an otherwise indentical 'rejecting' rule. The SMS gateway uses `ReplyAlways` and `ReplyNever` to support the 'Always' and 'Never' text commands.

5.3.6 Realising Persistence for Privacy Interactions

As discussed in section 3.3.5, our system requires that we maintain an audit trail of privacy-related interactions in order to increase accountability and traceability for users. However, internally, the framework is also required to maintain the ongoing status of information requests being handled by the platform for transforming synchronous operations into asynchronous interactions. All privacy requests are maintained in the underlying database whose schema is illustrated in Figure 5.16. Each entry in the `request` table contains details about a privacy request, e.g., identity of the requester, identity of the requestee, private information type, name of the service providing the information, time and expiry of the request, status of the request, and any additional contextual in-

formation given. Each entry in the `releaseinfo` table contains details of information disclosure, e.g., time of disclosure, the information disclosed, the identifier of the privacy rule that matched (if applicable). The `request` table and `releaseinfo` table can be left-joined using the `releaseID` key to generate a new virtual table (i.e., a database view), where each entry contains all details about a privacy request and its associated information disclosure.

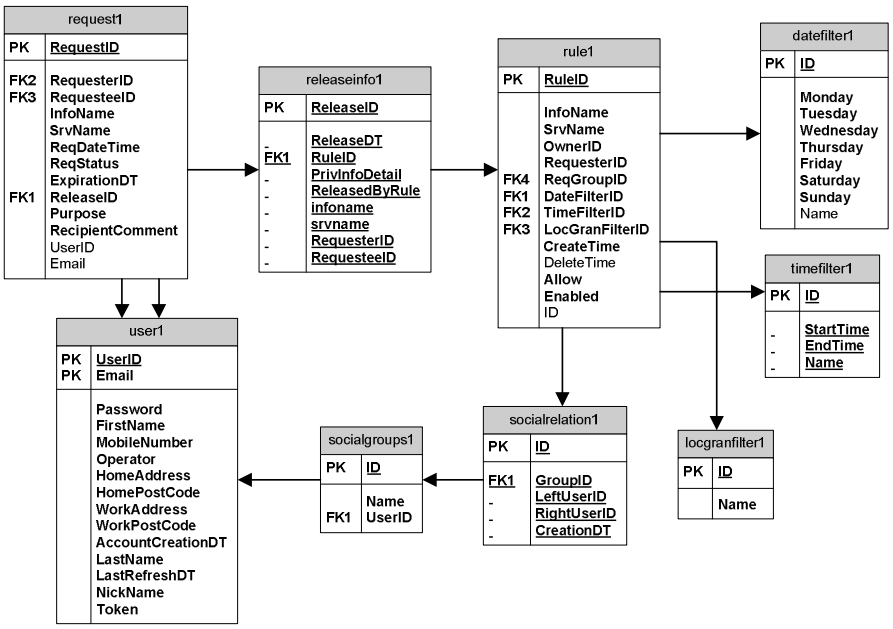


Figure 5.16: Database Table Schemas and Relationships

The database also keeps the privacy rules and user information for the system. Each entry in the `rule` table contains the type of the rule (i.e., positive or negative), private information type, service provider, identity of requester or user group, and identities of filters that are primary keys in the filter tables. User information (such as name, email and mobile number) is maintained in the `user` table, and user groups and group memberships are kept in `socialgroup` and `socialrelation` tables respectively.

```
public DataSet GetReceivedRequestsAllEx(int userID, string status);
public DataSet GetReceivedRequestsExpired(int userID);
public DataSet GetSentRequestsAllEx(int userID, string status);
public DataSet GetSentRequestsExpired(int userID);
public DataSet GetInfoReleaseAllEx(int userID);
public DataSet GetInfoReceiveAllEx(int userID);
public DataSet GetIndividualRulesEx(int userID);
public DataSet GetGroupRulesEx(int userID);
public DataSet GetAllPrivacyRulesEx(int userID);
public DataSet ProcessedRequestsByRule(int ruleID);
```

Figure 5.17: API methods for retrieving persistent information from the database

The persistence API methods shown in figure 5.17 are provided to allow applica-

tions to retrieve data from the underlying database. All methods return a `DataSet` object that holds in-memory cache of data retrieved from the database. The .NET framework offers several classes (e.g., `DataView` , `GridView` , `TreeView` , etc.) that can provide customised representations of the data cached in the `DataSet` . To retrieve requests sent to (or from) a user, client applications invoke `GetReceivedRequestsAllEx` (or `GetSentRequestsAllEx`) supplying the user identity and request status type as input parameters. The status type can be any string value of `ReqStatus` enumeration or `all` to retrieve all types of requests. To promote a user's awareness of private information flowing in and out of the system, client applications can invoke `GetInfoReleaseAllEx` (or `GetInfoReceiveAllEx`) supplying the user identity as the only input parameter. For instance, the returned `DataSet` object of `GetInfoReleaseAllEx` contains all information relevant to private information disclosure from a given user, i.e., the information disclosed, time of disclosure, the recipient, and the identifier of the related request. To retrieve privacy rules (for individuals or groups) created by a user, applications can invoke `GetIndividualRulesEx` or `GetGroupRulesEx` passing in the user identity. The platform provides the `ProcessedRequestsByRule` method to allow applications to retrieve privacy requests that have been processed by a particular privacy rule

Developers can use different mechanisms for achieving persistence of privacy requests by writing a plug-in for interface (**P4**). `AppendRequestHandler` takes a `PrivInfoReq` object as input and returns the unique identifier of the stored request.

```
(P4) public delegate int
        AppendRequestHandler(PrivInfoReq req);
```

Using this interface it would be possible to (for example) filter duplicate requests or aggregate similar requests on as they are received at the platform. The current implementation of this handler inserts the privacy requests into the `request` database table.

5.3.7 Discussion: Flexibility and Extensibility of the Prototype System

As we have argued, we do not believe we have proposed a universally acceptable solution for privacy management; the system architecture must remain flexible to adapt to changes of requirements in different problem domains in the future. In this section we have described how the middleware can be adapted and extended via a number of

plug-in interfaces, allowing new algorithms and policies to be introduced. The plug-in interfaces specify "what should be done" for privacy-related operations within the middleware framework, and defer "how those operations should be done" to the implementation of plug-ins. We have provided default algorithms for the adaptive privacy management case.

To summarise, the platform provides the following extension points:

1. new algorithms can be implemented (interface **P1**) for selecting the most applicable privacy rules for automatically handling privacy requests on behalf of users
2. developers can implement a plug-in (**P2**) for processing requests automatically
3. alternative rule evaluators can be introduced (**P3**)
4. input filters and aggregators can be introduced (**P4**)
5. new asynchronous notification transports can be added (**P5**)
6. new logging mechanisms can be added (**P6**)
7. rule configuration options can be suggested to users (**P7**)
8. new storage subsystems or schemas can be introduced (**P8**)

In addition to customising the behaviour of the platform, developers can customise the message formats used for all the notifications in the system by creating text or HTML templates (see Appendix E for sample templates). Message templates are instantiated at runtime and special fields in the templates are replaced with their dynamic values, e.g., the user's name, time issued, information disclosed, hyperlinks to add, etc. The implementation of the template mechanism is supported by a text template class in `C#` [Pruitt04] that uses regular expressions to dynamically fill out the tagged fields with appropriate runtime values. The file template mechanism separates the specification of the notification message format with the mechanism of getting runtime values from privacy events. Message templates can be modified without rebuilding or restarting the adaptive privacy management system.

Finally, developers can customise the .NET Remoting that is the underlying enabling technology for applications to remotely invoke methods exposed by the platform. .NET

Remoting can be extended to use alternative communication channels and message formatters [Esposito02]. The current implementation is configured to use the default binary format to encode messages and TCP to transport messages, and client applications set the configuration file correspondingly. To modify transport mechanisms (e.g., protocol or port number) or message encoding formats, developers can simply change the configuration files for .NET Remotingat runtime. This feature facilitates the deployment of the middleware platform and distributed client applications onto different platforms.

5.4 Case Study: A Privacy-Aware Location Sharing Application

We have discussed the prototype implementation of an adaptive privacy management system, i.e., the adaptive privacy manager, which supports multi-modal and multi-device user interactions for privacy management. The privacy manager is engineered using the APIs and customised plug-ins of the privacy middleware platform, and the prototype system is designed to meet the set of requirements we have identified for adaptive privacy management in section 3.3. To evaluate the design and implementation of adaptive privacy management, we integrate the privacy manager with a location sharing application in which users regulate released of their own location information. In this section, we motivate the need for a privacy-aware location sharing application, and present its design and implementation as a case study of adaptive privacy management. The implementation of the adaptive privacy manager and the location sharing application will be the basis of end user evaluation of adaptive privacy management in the next chapter.

5.4.1 Motivation for Location Sharing and Privacy

People share personal information with others in daily life to fulfil social goals [Goldberg02], and location information exchange has been a common practice for social disclosure in computer-mediated applications such as phone conversation, SMS, instant messaging and email [Smith05]. Studies of teenagers' behaviour of using SMS in England [Grinter01] and Germany have reported that two of the three top usages of SMS are to indicate the need for location information exchange within the content of SMS messages. Previous research [Oulasvirta05] demonstrated that a high proportion of un-

successful social communication attempts are due to the failure to conveying mutual contexts such as location and activities between people. To some extent, this finding explains the common behaviour of asking anothers' location and situation at the beginning of a phone call to establish mutual understanding of the mutual contexts [Arminen03, Weilenmann04]. Smith et al. [Smith05] further argued that location information (or a notion of place) is often sufficient to provide background knowledge for a wide variety of social communications, because a large amount of social state has already been shared between people who had established social relationships such as family members, friends, and colleagues.

Advances in location sensing technologies (e.g., GPS, 802.11, Bluetooth, mobile phone localisation, ultrasound, RFID, ultrawideband positioning, etc.) promote development and deployment of location-based applications that provide useful services based on users' current location [Hazas04] (often referred to as Location-Based Services or LBS [Kupper05]). For example, TomTom offers real-time travel navigation services using GPS device embedded in automobiles; mobile tour guides [Baus05] provide tailored information about tourist attractions based on users' current location using 802.11 based location sensing technology; GSM phone location tracking services in UK [Netcetera Limited07, Trace a Mobile.com07, MobileLocate Ltd.07] allow users to find the real-time location of registered mobile phones using a web interface or via SMS message, which can be used for people to locate their family members and friends. Network operators and services providers frequently hail LBS as the next 'killer application' for mobile computing with claimed potential revenues that exceed SMS messaging and ring tone downloads [UMTS Forum00].

However, the wide adoption of LBS has not happened as quickly as expected, and research found it was largely due to the lack of a clear regulatory framework and consumer privacy concerns [Escofet03]. In 2001, a U.S. Public Opinion Poll [Hendricks01] reported that 43% of the 1503 respondents felt that LBS would threaten their privacy and 70% of respondents said they were 'extremely' or 'very' interested in seeing US Congress pass related privacy legislation. Numerous user studies [Harper96, Kaasinen03, Barkuus03] have demonstrated that most people want to remain *in control* of their privacy while using LBS. As a result of people's privacy concerns, new legislation and regulations have been proposed that demand more protection for people's privacy. For instance, EU Directive 2002/58/EC on data protection and privacy [Communities02] requires explicit user consent before personal location information

can be made available for location service providers. The Mobile Broadband Group (MBG) in UK led an industrial working group and proposed an Industrial Code of Practice for the use of mobile phone technology for LBS [MBG06]. This code describes detailed requirements related to location privacy such as explicit user consent, age verification, random alerts to locatee, and the capability to stop the service at any time, The Internet Engineering Task Force's Geopriv working group identified a requirement for *"securely gathering and transferring location information for location services, while at the same time protecting the privacy of the individuals involved"* [Cuellar04]. The Geopriv protocol proposed the notion of Location Objects (LOs) that encapsulate location data with associated privacy requirements specified in privacy rules, and employed a variety of technical mechanisms (e.g., encryption, digital signature, unlinked pseudonyms, etc.) to prevent LOs from unauthorised use. In summary, achieving location information privacy is one of most crucial and challenging problems barring the mass success of LBS.

People's need for location sharing and the advances in location sensing technologies have encouraged the development and deployment of numerous applications [Google Inc.07a, Netcetera Limited07, Trace a Mobile.com07, MobileLocate Ltd.07, Smith05] that allow users to exchange location information with other members of a social network either manually or automatically. One of the research challenges for this type of application is to enable end users to balance the need for location sharing with the requirement for location privacy under different situations in dynamic environments. We decided to develop such a location information sharing application using the prototype adaptive privacy management middleware. Through engineering of the prototype application, we are able to demonstrate the feasibility of implementing adaptive privacy management into privacy-aware applications. Moreover, the implemented prototype application will be the basis for evaluating adaptive privacy management in the following chapter.

5.4.2 Intended End User Experience

The following user scenario illustrates the typical social contexts of intended use of the location sharing application and explain its basic functionality from an end user point of view.

David in Lancaster was driving to Nottingham for a second project meeting on Wednesday with Bob. They planned to have dinner together in the Tandoori restaurant at 18:00. Before leaving the office, Bob sent out a request for David's location from a web page at 17:00. David saw the request in an SMS message while having coffee at the motorway services at 17:30. By replying "always", David releases his current location and grants Bob access to his location from then on. Bob receives David's location on his mobile and knows that he is still outside Nottingham. At 18:15, Bob sends another request via SMS while waiting at the restaurant, and the system automatically releases David's location. Knowing David is just three streets away from the restaurant, Bob starts ordering their favourite Poppadams as a starter.

After the project meeting on Wednesday, David noticed one email informing him that Edward (one of his PhD students) had requested his location at 15:00. Realising Edward might need to find him to discuss his thesis, David clicks a link in the email and adds a privacy rule through a web page: allow user group 'PhD Students' to access my location at city granularity all day during weekdays. David also sends an email to Edward proposing a meeting around 14:00 on Thursday. At 14:05 on Thursday, Edward sends a request for David's location and found him to be about five miles away from the department. Edward makes himself a cup of coffee and starts preparing for the meeting.

Over the weekend, David is reviewing the location requests he has received. He finds the privacy rule for Bob that he created on Wednesday is still there. Realising the next project meeting would be in three months, David selects the rule and deletes it. David noticed that Edward requested his location last night, and his privacy rule had automatically released his location. Feeling uncomfortable that Edward can find his location in the evenings, David modifies the privacy rule to only allowing him to find out his location from 9:00 to 17:00 on weekdays.

The above scenario demonstrates the major capabilities of the privacy-aware location sharing application:

- send private information requests via a web interface or SMS;

- explicitly accept or reject requests via either interface;
- set up privacy rules to automatically process requests;
- set up simple privacy rules via SMS messages;
- receive privacy notification from email, SMS, or web pages;
- monitor status of information requests, information flow, and effect of privacy rules.

5.4.3 Location Sensing and Map Services

In order to develop a location sharing application, we have to decide what location sensing technology to use. We have employed a location sensing service that uses location data of GSM mobile phones for the following two reasons: first, many people have already regarded GSM mobile phones as a commodity that they carry with them most of the time [Davies02], and therefore GSM phone based location sensing does not require people to carry additional devices such as GPS receivers or RFID tags. Second, GSM phone based location sensing works both indoors and outdoors, as long as a mobile phone can register with a GSM cell tower nearby.

GSM network operators can simply use the location of the GSM cell tower associated with a handset to estimate a phone's position. In addition, network operators can use timing differences in the arrival of the uplink signal from the handset to several (at least four) cell towers to calculate its position [Kupper05]. Network operators maintain the location data of handsets in a database, and have made this location data available to third-party location service providers to create LBS. The access to the location data is regulated by legislation (e.g., EU Directive 2002/58/EC as discussed in section 5.4.1), and provision of LBS needs to follow a strict industry code of practice [MBG06]. The third-party service we use (i.e., FollowUs) provides real-time location of registered phones on a 'pay per request' basis. The accuracy of location data varies depending on the density of cell towers, from a few hundred metres in built-up areas and up to a few kilometres in less densely covered rural areas. Recent research has also shown that mobile phones are often not as close to individuals as we might expect [Patel06].

```
<?xml version="1.0" encoding="utf-8"?>
<GsmLocationInfo>
  <InfoDateTime>2007-01-15T22:17:07</InfoDateTime>
  <Address>MEADOW PARK GALGATE LANCASTER LA2 0NJ</Address>
  <Longitude>-2.800757</Longitude>
  <Latitude>53.9952164</Latitude>
  <Radius>1.96</Radius>
</GsmLocationInfo>
```

Figure 5.18: Location data sample in XML format

To use the 'FollowUs' services for location sensing, we created an account on its website and added the mobile phone numbers of the trial participants to the account. We developed a utility class in C# that programmatically retrieves and parses the required HTML pages from the FollowUs web site to extract the real-time location data. As illustrated in figure 5.18, the location data is wrapped in an XML format that contains geographic coordinates (i.e., longitude and latitude), postal address, timestamp, and a radius that the mobile phone is within. To make the location data more meaningful and useful, we employed the 'Google Maps' service [Google Inc.06] to plot the location of users. End users can explore the location with standard services provided by Google Maps, e.g., zooming, street map, satellite map, etc. To embed Google Maps in our web pages developed in C# ASP .NET, we have employed a free utility [Reimers07] that wraps raw Google Maps 2.0 Javascript APIs into an ASP .NET server-side control.

5.4.4 Integrating with the Adaptive Privacy Manager

As shown in figure 5.19, the location sharing application is integrated with the adaptive privacy manager as one of the private information sharing services. Each user can interact with the system via the web portal or the SMS gateway after registering with the system. For instance, user A can follow the hyperlink in the web portal or send a text message from his mobile phone to the SMS gateway, to issue a location request to user B. Both interaction modalities allow user A to provide additional contextual information with the request, e.g., the reason for requesting the user's location. The web portal or SMS gateway receives user interactions, and invokes the `IssueInfoReq` method on the platform. The synchronous notifier on B's office PC pulls the latest privacy events from the platform periodically, and notifies B of receiving a privacy request via the web-based popup alert window. Asynchronous notifications are also pushed from the platform via email and SMS to B's personal devices (see sample notification email and SMS message

in Appendix D).

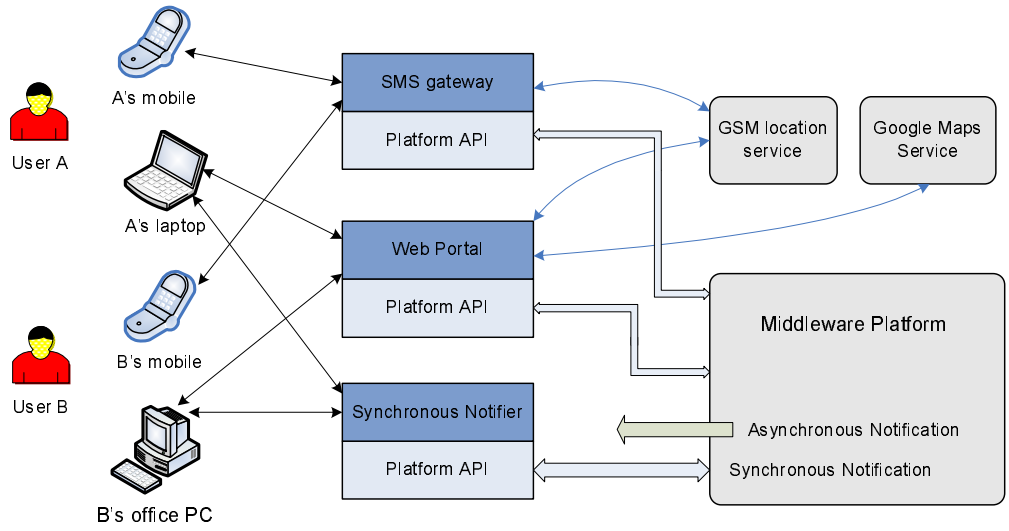


Figure 5.19: Integrated privacy-aware location sharing application

User B (i.e., the requestee) can choose to process the request using either interaction modalities by responding to the privacy notifications: B can follow the hyperlink to the request management page in the pop-up alert window and email notification, or reply to the SMS message with the identity of the request. Since the location sensing technology only provides location in terms of street name or coordinates, our system allows a requestee to provide additional contextual information with the reply, e.g., personal description of the location, activities taking place at the location, etc. The decision on private information disclosure is received by the web portal or SMS gateway, which contacts the GSM location service if needed and invokes API methods for accepting or rejecting request(s) (see figure 5.19).The status change of the request and details of information disclosure are recorded in the database. B will receive privacy notifications of the information disclosure event and A's location information (see notification email and SMS message in Appendix D).B can choose to show the location on maps through the web portal, which contacts the Google Maps service to retrieve and display geographical and street maps for the location (as illustrated in figure 5.20).

A user can specify rules that either accept or deny location requests from an individual or a group of users, and he can define date and time filtering conditions for processing requests. Moreover, users can intentionally introduce ambiguity about location information by setting the granularity field in their privacy rules (e.g., "Street Level" 1.5km, "Area Level" 3.0km, "City Level" 5.0km, "County Level" 20.0km, and "Country Level" 100.0km.), and disclose less accurate location information (courser

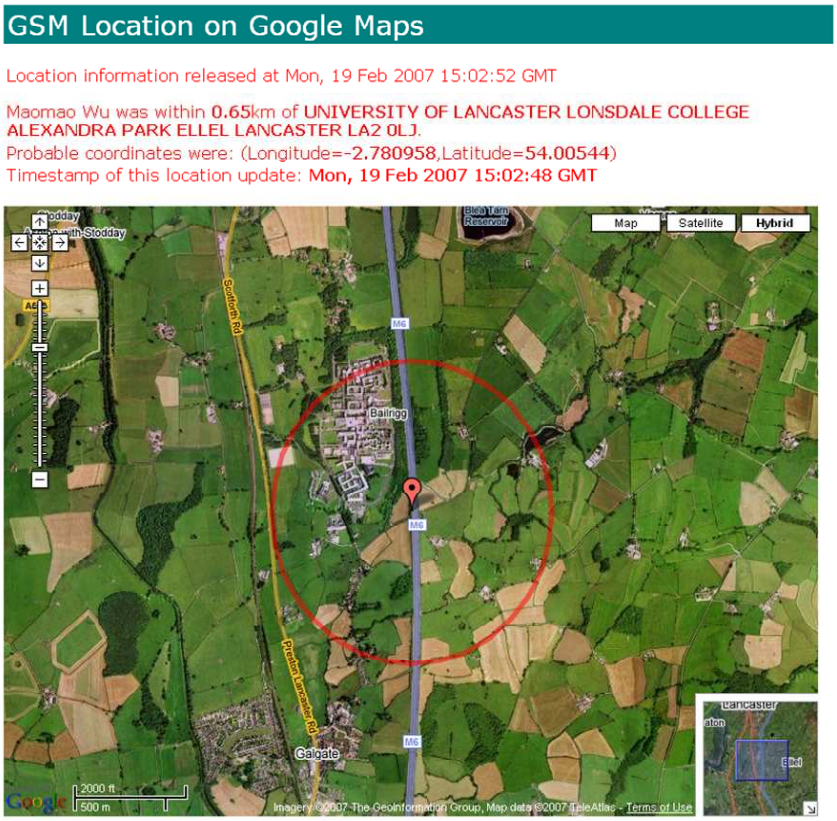


Figure 5.20: Location information shown on Google Maps

granularity) to give the requester a 'rough idea' of whereabouts without revealing their precise location. In the end user scenario, David had created a privacy rule that allowed user group "PhD students" to receive his location information between 9:00 17:00 on weekdays at city level granularity (i.e., around 5.0km radius).

The current implementation of the system enables users to add rules at any time by following the "add a privacy rule" link. In addition, users can create rules while processing privacy requests. For example, a user can ask the system to suggest options for a privacy rule to process a particular request, and then the user can modify the suggested options as appropriate and save them into the privacy rule. For users on the move, they can reply "Always" or "Never" to the SMS privacy notification to create a privacy rule that allows or disallows location disclosure to the requester all the time. This can be refined later using the web interface.

Finally, the web portal provides web interfaces for users to monitor information requests (sent and received), information flows (in and out) and created privacy rules, because they persist within the adaptive privacy management system. For instance, a user can show all received information requests in a table-like user interface (as illustrated in

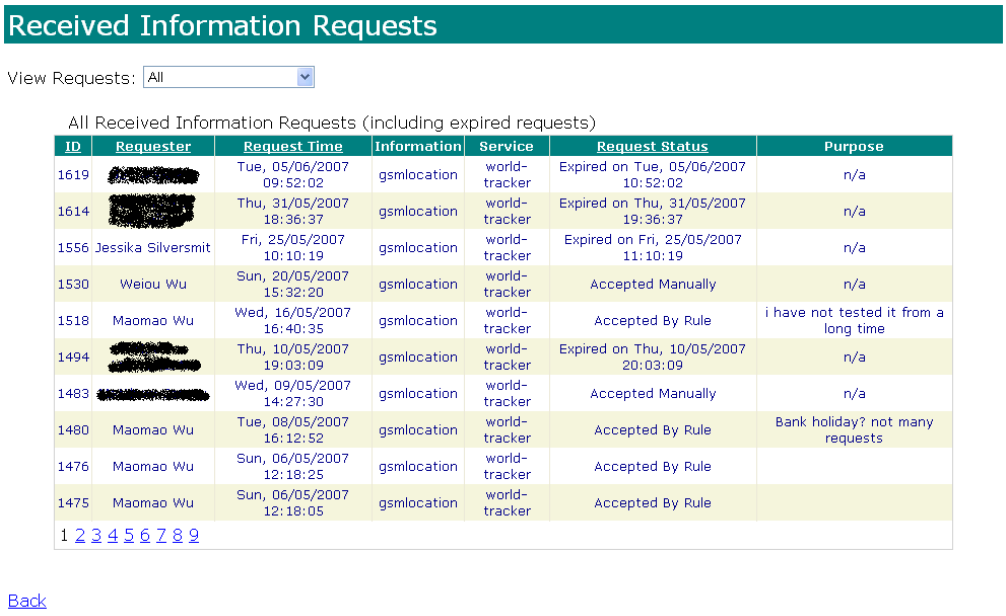


Figure 5.21: Received information requests monitoring page

figure 5.21) that display all relevant information of each request, e.g., requester name, request time, request status, time of information disclosure, a hyperlink to detailed location information, etc. A user can filter requests by their status using the dropdown list control at the top of the page and sort requests by a request field using the hyperlink on each column header. The web portal also provides a link for users to retrieve privacy requests sent and received for a certain date, and a user can browse privacy events for the day and optionally leave comments about interesting privacy events. This is the basis of the privacy diary system that has been used to gather subjective feedback from participants in the user trial (see section 6.2.2).

5.4.5 Improving Usability

To improve the usability of the privacy-aware location sharing application as well as the adaptive privacy manager, we have employed three well known HCI techniques for evaluating their user interfaces, i.e., expert heuristic evaluation, cognitive walkthrough by non-expert users, and hierarchical task analysis (HTA).

A heuristic is a general principle or rule of thumb for user interface design. Heuristic evaluation was developed by Jacob Neilsen and Rolf Molich as a method for criticising the usability of a system using a set of simple and general heuristics [Dix98]. We employed ten widely-adopted usability heuristics proposed by Nielsen [Nielsen94] to

identify potential problems and improve the usability of the web portal. Some user interface features of the web portal followed the usability heuristics proposed by Nielsen. For example, to promote "flexibility and efficiency of use", each web page provides hyperlinks to major tasks in the left pane and hyperlinks to contextual tasks. To increase "visibility of system status", the main page of the web portal provides status information of the system (e.g., receiving or sent requests that are waiting for processing, privacy rules that have been specified, etc), and pop-up notification windows are displayed when critical privacy events occurr. To "match between the system and real world", the web portal describes major user tasks in plain English without technical jargon, and provides explanation of privacy rules in colloquial language. To follow the "consistency and standard" heuristic, we standardised different words and actions that mean the same thing. Finally, to provide "help and documentation", the web portal contains hyperlinks to help pages (e.g., FAQ, HowTo, Introduction, etc) on every web page.

Cognitive walkthroughs require a detailed review of a sequence of actions, and an action sequence refers to the steps that an interface will require a user to perform in order to accomplish a given task [Dix98]. We have selected a small number of non-expert evaluators and given them a list of representative tasks that most users will want to perform for interacting with the privacy-aware location sharing application, e.g., sending and processing location requests, creating a privacy rule to automate request processing. For each task, an evaluator steps through the sequence of required user interactions (i.e., walkthrough) to criticise the user interface of the application and make suggestions for usability improvements. We received a few useful suggestions for improving the usability of our system, e.g., using table frames to separate different parts of privacy rule options for the 'adding privacy rule' page, using tabs or hyperlinks to filter privacy requests based on their status and using tick boxes for selecting multiple requests to process (like web-based email client interfaces).

Hierarchy Task Analysis (HTA) [Dix98] decomposes major user jobs into a hierarchy of tasks and subtasks as well as plans describing in what order and under what conditions the subtasks are performed. The output of a HTA can be recorded in a textual outline format or in a tree diagram. We employed the HTA to evaluate the existing structure of the web portal interface. More particularly, we identified the major tasks for interacting with the privacy-aware location sharing application, and we describe those tasks in some end user scenarios. Identifying the major tasks of the system helped us to generate the top-level structure of the web portal user interface which determined the

menu structure. Next, we decomposed the major tasks (e.g., sending a location request, accepting or rejecting location requests, creating a privacy rule, etc.) into subtasks and plans, and describe them in a textual outline format. The task sequence obtained from a task decomposition can be used when designing contextual hyperlinks with each web page, e.g., the creating privacy rule page may contain a hyperlink for creating a user group, the changing user membership page may contain a link for creating a privacy rule, etc. The task decomposition and plans helped us to identify frequently performed subtasks, and therefore make it easier for users to access those subtasks by organising them conveniently.

5.5 Summary

In this chapter we have presented the prototype implementation of an adaptive privacy management system, i.e., an adaptive privacy manager, and an associated supporting middleware. The chapter describes the core API methods exposed by the platform that are essential for supporting adaptive privacy management. We have identified the important plug-in interfaces for customising and extending this basic functionality, as well as the internals and operations of the platform and algorithms employed for developing the default plug-ins. The discussion of the prototype implementation focuses on how the aspects of the system meet the requirements of the adaptive privacy management, i.e., promoting privacy awareness via notification, support for making privacy decisions in context, automating privacy decisions using privacy rules, balancing user involvement and privacy rule management, releasing persistence for privacy interactions, and promoting flexibility and extensibility. Finally, the chapter has discussed a proof-of-concept location sharing application integrated with the adaptive privacy manager, which both demonstrates the feasibility of the architecture and illustrates the workings of the platform. This location sharing application will be the basis of the end user evaluation described in the next chapter.

CHAPTER VI

Evaluation

Contents

6.1 Overview

The previous two chapters presented the design and implementation of a prototype platform that supports the development of adaptive privacy aware applications as well as a location sharing application built using it. This chapter presents an evaluation of the principles of the adaptive privacy management based on the deployment and end user trial of the implemented location privacy system. We first describe the experimental methodology of the user trial that involved three different phases and employed multiple evaluation techniques. Then we present general findings from the study, including analysis of quantitative results of the usage data and qualitative discussion reflecting on user experience of the system. Finally, we present an evaluation of the principles of the adaptive privacy management as well as the design and implementation of the implemented location privacy system, followed by discussions reflecting on the strengths and limitations of the implemented system.

6.2 Experimental Methodology

We conducted a three-phase user study based on the implemented location privacy system during April to May in 2007 with 30 participants. In phase 1, we gathered participants' background of using computers and mobile phones as well as their privacy attitude and initial thoughts of sharing location. In phase 2, participants used the deployed system over a period of $7^1/_2$ weeks to initiate and respond to location requests as real need dictates. Our experimental system logged core usage data and provide a web-based privacy diary system for participants to record non-overt information, e.g., intention for sending a location request. In phase 3, we conducted surveys and interviews to allow participants reflect on their experiences and attitude toward adaptive privacy management. In the following sections, we describe each phase of the user study in more detail.

6.2.1 Phase 1: Preparation Tasks and Opening Questionnaire

We chose our target population as people who have been using computers and mobile phones for sometime, because it eliminated the need to provide training for them and they may have experience of online privacy issues. We sent a solicitation email

to departmental mailing list inviting people to participate in our study. We encouraged people to introduce their friends and family members, because we expected that more interactions will happen between members of a same social group. Another reason for choosing this set of participants is that they would be forgiving any unexpected technical and ethical problems. The solicitation email contains a URL to a short introduction of the system, briefly explaining why we built it, what it can do, how it works, and how it could benefit the users. The study participation is completely voluntary, although we promised some compensation (i.e., an Amazon voucher) for each participant as a further incentive. Next, we asked each respondent to sign a research protocol form and complete an opening questionnaire. The research protocol form (Appendix A) explained the follows to the participants, including purpose of the study, procedure of the study, risks of participation, benefits, cost and compensation, etc. The opening questionnaire (Appendix B) was designed to gather participants' background of using computers and mobile phones as well as their attitude toward privacy and initial thoughts of sharing location information. Both the research protocol form and opening questionnaire were completed by the participants before they were allowed to start using the location privacy system. We contrast the anticipated usage of the system from this questionnaire with actually recorded use in section 6.5.4.

Participant Profiles

30 people, 21 (70%) male and 9 (30%) female, participated the user study, in which 20 (67%) are members of the computing department of Lancaster University (e.g., lecturers, researchers, and PhD students) and 10 (33%) are their friends, spouses or other family members. 2 participants were involved in designing the system. Since participants either responded to the solicitation email voluntarily or were invited by their friends and family members as we intended, there exist a few active social groups whose members have the need and desire to locate each other. Participants were aged between 19-61, and the age distribution is illustrated in figure 6.1. Geographically, 23 participants lived in or near Lancaster, 4 in other counties of UK, and 3 in overseas countries.

All participants use a personal computer or laptop both at work and at home, and they have access to the Internet both at work and at home. All participants have used computers for many years (minimum 7 years, maximum 32 years, Mean=15.90 years, and SD=6.32), and the average self-rating for PC skills (from 1: novice to 5: expert)

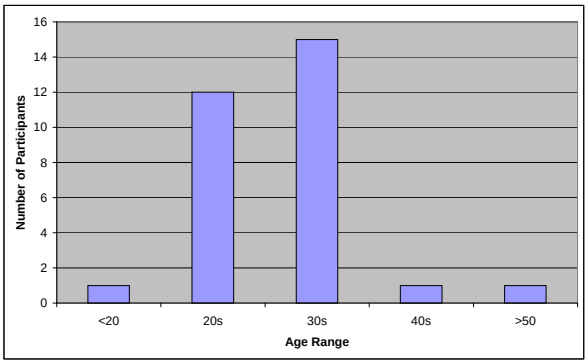


Figure 6.1: Age distribution of participants

is 4.63 (SD=0.67). All participants except one often carry a mobile phone with them while they were away from home or office. Participants have mobile usage experience from 2 to 12 years (Mean=4.47 years, SD=2.40). All 26 mobile users use their mobile phones for both voice calls and text messages, within which 12 send less than 10 SMS messages per week, 5 send 10 to 20 messages, and 9 send more than 20 messages.

Since people's perception of privacy greatly influences their decisions to disclose personal information, it is necessary to identify participants' general attitude toward privacy. The Westin/Harris Privacy Segmentation Model [Harris98] was employed in Phase 1 to categorise participants into three groups according to their different levels of privacy concerns. This methodology was developed by well-known privacy expert Alan Westin, and has been widely used by many research projects on privacy [Harris Interactive07, Smith05, Consolvo05, Joinson06]. Participants were divided into the following three categories based on their answers to three statements on a four-point scale:

- *Privacy Fundamentalists:* have "very high privacy concern" and distrust businesses on properly handling consumers' private information.
- *Privacy Pragmatists:* have a balanced attitude towards privacy. They often "ask what benefits they get as consumers in sharing their personal information to balance against risks to their privacy interests, and they usually favour a mixture of government and private solutions".
- *Privacy Unconcerned:* have "little to no concern about consumer privacy issues" and allow anyone to record and use their personal information.

The breakdown of participants based on the privacy segmentation model is shown

	HarrisPoll'01	HarrisPoll'03	Joinson'06	This Study
Privacy Fundamentalists	34%	26%	32%	33%
Privacy Pragmatists	58%	64%	56%	60%
Privacy Unconcerned	8%	10%	12%	7%

Table 6.1: Privacy segmentation of study participants

in Table 6.1. 10 participants were privacy fundamentalists, 18 were privacy pragmatists, and 2 were privacy unconcerned. The trend for privacy classification is consistent with the results in reported in US Harris polls [Harris Interactive07] and a UK privacy attitude survey [Joinson06], where the majority of the participants were privacy pragmatists, followed by privacy fundamentalists and privacy unconcerned. This showed that the sample population we chosen were comparable to the general user population in US and UK in terms of their attitude toward privacy.

6.2.2 Phase 2: Deployment of the System

We created an account on a third-party GSM-based location sensing service (i.e., FollowUs) and added participants' mobile information to our account. Each participant received an SMS message from the LBS provider informing them the account holder can locate his/her mobile. By replying to the message with a PIN number as instructed, the participant granted access for our account to track their GSM mobile phone. In addition, the service provider will generate random SMS alerts to traceable mobile holders, reminding them that their mobile phone can be tracked by our account. After a participant's mobile is added into our account on FollowUs, we created an account for the participant on our location privacy system so that they can interact with it using both web interface and formatted SMS messages. We provided a web page describing how to interact with the system and a FAQ web page answering common questions. In phase 2, participants were asked to make and receive requests for their location or the location of other registered users as real need dictates. In contrast to other user studies [Lederer03b, Smith05], participants were not given any explicit tasks to complete, and the usage of our system is totally voluntary and out of real demand. After first week of the deployment, we introduced a new type of user (i.e., 'web only user') to the system, and web only users can only request other normal users' location but cannot be located. This is to satisfy the real demand from a participant, because his family members and friends who lived outside the coverage of our location service (e.g., overseas) wanted to

know his location. Of all the 30 participants, 26 were 'mobile users' and 4 were 'web only users'. Important interactions with the system were logged with timestamps for the data analysis in phase 3.

Privacy Diary System

Logged usage data revealed what users did and how they interacted with the system, but it fails to uncover non-overt behaviour such as user intention. To cope with this problem, we designed a web-based privacy diary system that allows users to note down subjective matters about using the system. A daily email (see Appendix D for a sample email) was sent to each participant remaindering them to leave comments in their personalised privacy diary, and URLs were provided in the email to facilitate filling the diary. The reminder email also provides options for participants to describe the reason for not writing the diary entry, i.e., "no time to leave comments" or "no interesting events happened today", allowing us to differentiate intentional non-completion. A privacy diary page displays important privacy events of the day (e.g., details of received and sent location requests), and provides users web interface to type in comments related to the events happened on that day, e.g., why they sent a request, why they reject a certain request, what they thought they did, etc.

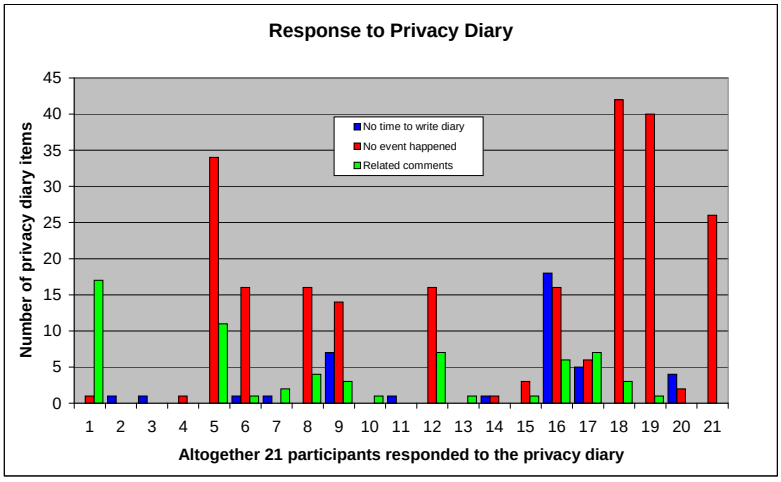


Figure 6.2: Response to privacy diary

Filling in the privacy dairy entries is voluntary, and it provides a good basis for conducting the interviews at the end of the study. During the user trial, 339 entries were added into the privacy diary system by 21 (70%) participants: 40 said 'no time to write diary today', 234 said 'no interesting event happened today', and 65 were meaningful

comments related to the system or general privacy issues. The breakdown of privacy diary entries by each participants were shown in figure 6.2.

Introducing a Stranger

To provoke a reaction, we introduced a stranger into the system just before the end-of-trial interviews when our participants were already familiar with the system. This helped us examine the effectiveness of our privacy management mechanisms and investigate how our participants would respond to location requests from unknown people. We deliberately picked a name of a real person with no association with the participants, and ensured that it had no obvious online presence.

6.2.3 Phase 3: Surveys and Interviews at the End of the Trial

In phase 3, data logs of user interactions with the location privacy system were extracted from the database, and analysed using statistical methods to reveal preliminary findings of the system. Privacy diaries of individual participants were consulted and helped us uncover hidden factors that cannot be revealed from usage data log alone. In addition, entries in privacy diary work as reminders for participants to remember the context of privacy-related interactions with the system, e.g., activities when receiving a request, reason of accepting a request, unexpected behaviour of the system, etc. Based on the usage data and saved privacy diaries, we asked participants to reflect their experience of using the system and solicited their thoughts using two different evaluation techniques. We conducted an end-of-trial interview for each mobile user and asked him/her to complete a survey questionnaire during the interview (100% responses rate). The questionnaire (Appendix C) contains Likert-style statements related to the usability of the location privacy system as well as the principles that constitute adaptive privacy management, and participants were asked to rate those statements on a 5-point scale (1 is strongly disagree and 5 is strongly agree). The questionnaire consists of both conceptual statements to evaluate the requirements of adaptive privacy management, and system-related statements to evaluate whether the prototype implementation meets the requirements. Interview questions were asked after a participant answered each section of the questionnaire to solicit the rationales why the participant made the choices. Finally, each participant was asked to classify all the other participants into different social

groups, in order for us to know the social relationships between participants. Most of the interviews were conducted face-to-face in the author's office, and the rest were conducted over the phone because of difficulties in physical presence. Data was collected in the form of audio recordings and evaluator notes, as well as materials completed by the participants. Only the author was involved in the interview process to avoid any power relationship between the interviewer and the interviewee. For the web only users, we asked them to fill a separate questionnaire mainly on the system usage (75% responses rate).

6.3 Quantitative Analysis of Usage

This section presents general findings of the user study. We analyse quantitative results of the system usage based on the analysis of logged data, and discussed interesting quotes from privacy diaries and interviews related to the findings. The following sections present the results about location requests, privacy rules and user groups, and responses to requests from the stranger respectively.

6.3.1 Location Information Requests

Figure 6.3 illustrates the number of location requests each day, where Saturdays are shown in green, Sundays are shown in red, and two UK bank holidays are shown in orange. The total number of location requests made by the participants is 297[1], where 21 were highlighted as requests sent by designers for testing purpose (most of them were sent in the first week for debugging and the rest were sent to confirm if the system is still alive). 6 mobile users' accounts were created before the trial, 18 were created during the first five days, and the other two were created on 9th and 28th day respectively. First web only user's account was created on the 8th day, the second one on 15th day, the third one on 17th day, and the fourth one on 27th day. The stranger was introduced on 40th day, and the end-of-trial interview started on the 44th day and ended on 52nd day. The maximum number of requests made per day is 30, and the minimum number of requests made per day is 0 which happened on 9 days during the trial. The average number of requests during 53 days is 5.60 (SD=6.89), the average during the first week

[1]This figure does not include the requests made by the stranger, and we discuss these further in section 6.3.4

is 17.29 (SD=10.61), the average starting from the 44th day is 1.91 (SD=2.81), and the average from 8th day to 43rd day is 6.44 (SD=7.29). There were 14 requests on the 26th because one web user was actively locating another participant, and there were 13 real requests on 9th day because two participants were actively sending requests (5 each).

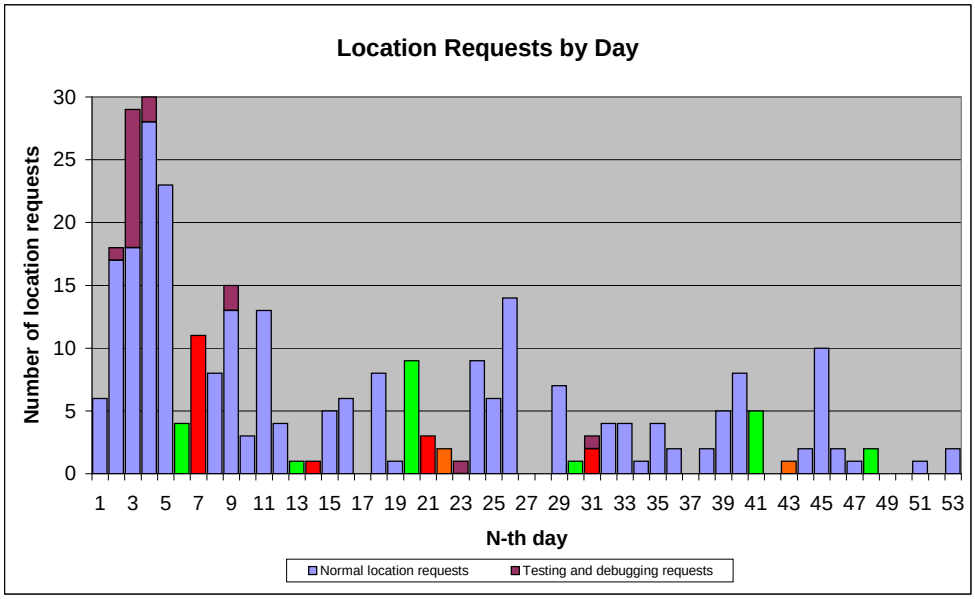


Figure 6.3: Number of location requests by day during the user trial

Experimenting with the System

Figure 6.3 indicates that the participants were most active during the first week, and we attribute this to the well-known *"novelty effect"*, i.e., the system is novel to users and they were experimenting with it. Participants' first impression of the system is quite positive, and they found the system easy to use and understand. Most participants were interested in knowing how accurate the location released by system is, and some complained that the location returned was not very accurate as some wrote in privacy diaries:

> *Quote (M9): Today my account was active and I tested out the system by finding my friend, and he tried to find me. I wasn't impressed with the accuracy, all I got was a very large general area, I would rather know more specifically if this service would ever be useful for finding my friends.*
>
> *Quote (M23): First time logging in after being out last night, interesting to see how accurate the location information is. Tried to make a request for*

> *the first time, and looks like he's got his phone turned off! I'll try again later. Later: tried the request again, easy enough. Also looked at my own location, seems a bit far out – according to this I'm on a golf course, not on campus! Campus is in the far right of the circle, but the point it suggested is about 2km away.*
>
> *Quote (M21): I had some item to test this today and it appears to be working fine. The Google map hook in is very interesting and surprisingly accurate.*

People did experiment with the system during the trial, and there was at least one participant (M22) who sent 9 testing requests to others on the 45th day, one day after we started the final interview. The participant was testing it just before her interview on the 47th day.

Breakdown by Time

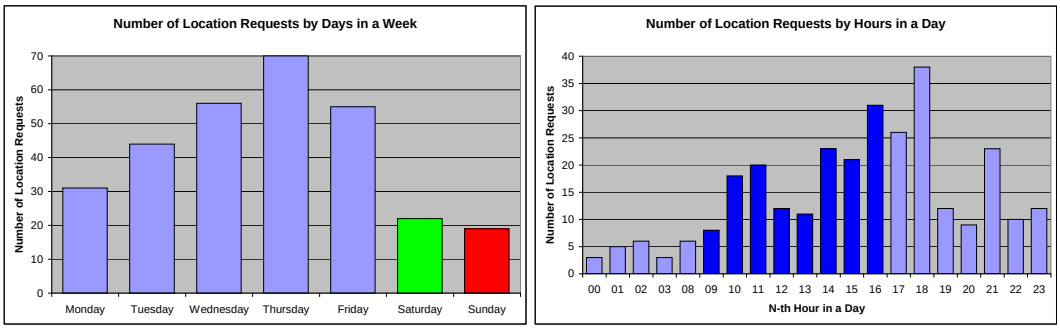


Figure 6.4: Number of location requests on days in a week and during hours in a day

Figure 6.4 illustrates the number of location requests made on days in a week and during hours in a day. It indicates that there were less location requests made on holidays (i.e., weekends and bank holidays) than those made on weekdays. This is probably because the dominant social relationships between the participants are colleagues and friends, and they tended to make location requests during working days. Relatively less location requests were made before working hours, and relatively more location request were made during 17:00–19:00 and 21:00–22:00, apart from typical working hours in UK (i.e., 9:00–17:00). This might be because that a large percentage of the participants were working in a research environment where there is no strict regulation on working hours.

Breakdown by Participant

Figure 6.5 illustrates the number of location request related to each participant, including requests sent to oneself, requests sent to other participants, and requests received from others. The first 26 participants are mobile users who can both send and receive location requests, and the last 4 participants are web only users who can only send requests. 2 mobile users never received any location requests during the trial, and 1 mobile user and 1 web only user never sent any location request. 2 mobile users sent higher number of self-requests: one (M1) was actively testing whether the location system works properly while he is on the move, and the other (M12) regularly located her own mobile that was given to her husband. The shared usage of mobile phone was not expected in designing our privacy management system, because our solution assumes every single user possesses a separate phone and does not consider the cultural difference in sharing personal devices [Chipchase07]. Therefore, we assume that our system only applies in western countries such as European countries or US. The mobile user who received highest number of requests from others (M8) is the one who asked us to introduce web only users into the system, and two of his family members were actively requesting his location during the trial.

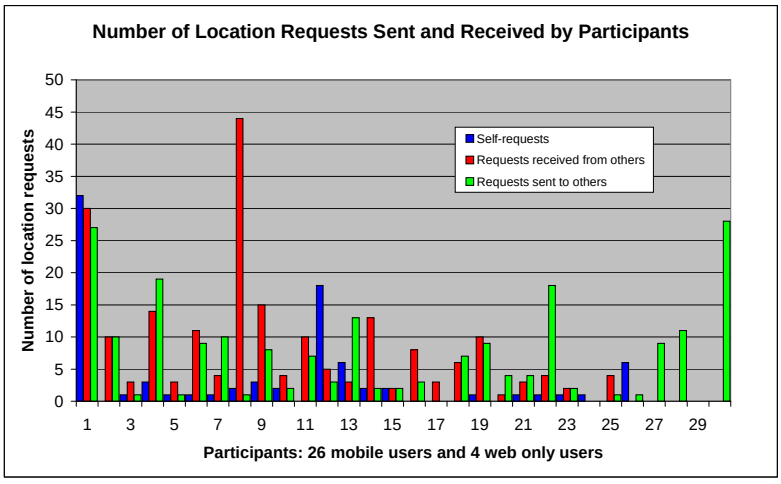


Figure 6.5: Number of location requests sent and received by participants

Breakdown by Status

Of the 297 location requests, 279 (94%) location requests were sent from web pages, and 18 (6%) were sent from mobile phone. The breakdown of location requests in different status is illustrated in figure 6.6. 58 (20%) requests were accepted manually (28 using

web interface and 30 using SMS message), and 10 (3%) were rejected manually (7 using web interface and 3 using SMS message). Participants tended to manually process received requests using their mobile phone, and this might be because they were first notified by SMS messages and reluctant to switch interaction mode to process requests. 78 (26%) were self-requests accepted by a default rule, 84 (28%) were accepted by privacy rules set by participants, and no request was rejected by any privacy rule. 5 requests were cancelled by requester, and 45 (15%) were either expired or ignored by requestee. Finally, 17 (6%) requests failed to get location information from the service provider, of which 4 were because a participant gave a wrong name for his network operator, and 13 were either because the mobiles were switched off, out of network coverage, or out of UK.

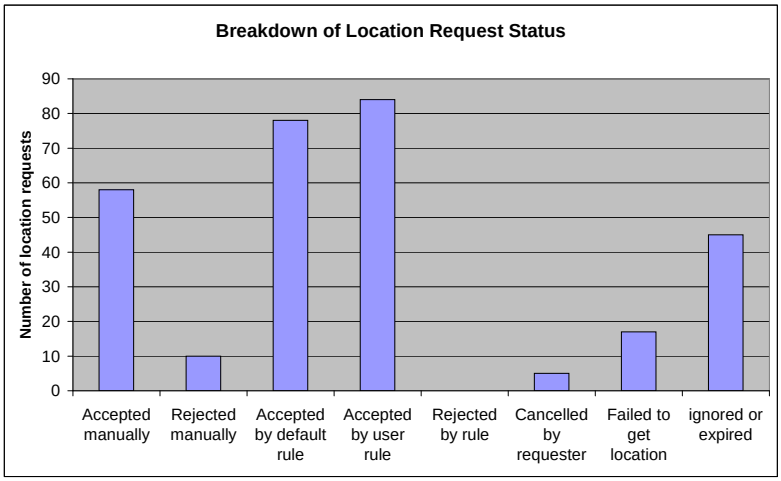


Figure 6.6: Breakdown of location requests in different final status

The system worked reliably during the trial, and a few participants mentioned that the system was very reliable and they did not experience any message loss during the final interview. One participant (M9) complained about a delayed SMS message containing location information as he written in his privacy diary: *"Yesterday I was in Manchester and I wanted to know if the friend I was meeting was already in Manchester. So I tried to find him (at around 2am), however the request took 7 hours before a reply was sent to my phone, thus the information was late (and useless)."* It was beyond control of the system, and it might be because of problems of SMS handling software or messaging centre.

During the final interview, we asked participants whether they were aware that they can set expiration time for a location request. 15 participants were aware and 11 were not. 17 people thought it was useful, and other 9 participants did not answer it. The

participants commented that it is useful because sometimes they only need to know someone's location during a certain period of time and the location returned out of that period would be useless for them. Despite people thought it was a useful feature, the usage of non-default expiration time is low: 281 (95%) location requests used the default expiration time (i.e., 1 hour), and the rest 16 (5%) requests used non-default expiration time by 4 participants. One reason is that the system only allowed people to set non-default expiration time from the web interface, and therefore the expiration times of requests sent from mobile were all set to the default. We can also speculate that people might find the 1 hour default value suitable for most of the cases and they do not need to change it.

6.3.2 Sharing of Context

The number of requests that each participant provided contextual information in request and reply were illustrated in figure 6.7. 68 (23%) location requests made by 9 different participants contained contextual information (e.g., purpose of the request), and the content of the information can be categorised as the follows: 9 was about rendezvous or trying to find someone, 13 was questioning something or starting a conversation (e.g., 'r u available for lunch?', 'quick chat to arrange a meeting', etc), 14 mentioned specific name of a place (e.g., 'how r u? r u in Lancaster?', 'r u still in UK?', etc), 5 mentioned time, 6 mentioned activity, 2 were for fun or entertainment, 15 were sent by normal users for testing (one participant once sent 9 requests with 'testing from infolab' to others just to experiment with the system), and 20 were sent by the two designers to debug and test the system. 29 (10%) manually processed requests (20 accepted and 9 rejected) contained contextual information in the reply, and they were made by 6 different participants. Among the 29 contextual information in the reply, 9 was describing more accurate location (e.g., 'yes, i am in infolab21', 1still in HALA', etc), 7 was justifying the action of accepting or rejecting (e.g., 'hello, i know you are M1's cousin', 'hi M22, I don't know you well enough for you to be tracking me :)', etc), 4 was starting or resuming a conversation, 2 mentioned about time, 2 mentioned activity of the requestee, and 5 was about testing.

During the final interview, 17 out of 26 (65%) participants said they were aware of providing contextual information in a location request or response. 23 people thought it was useful, and other 3 participants did not answer it. Most respondents mentioned

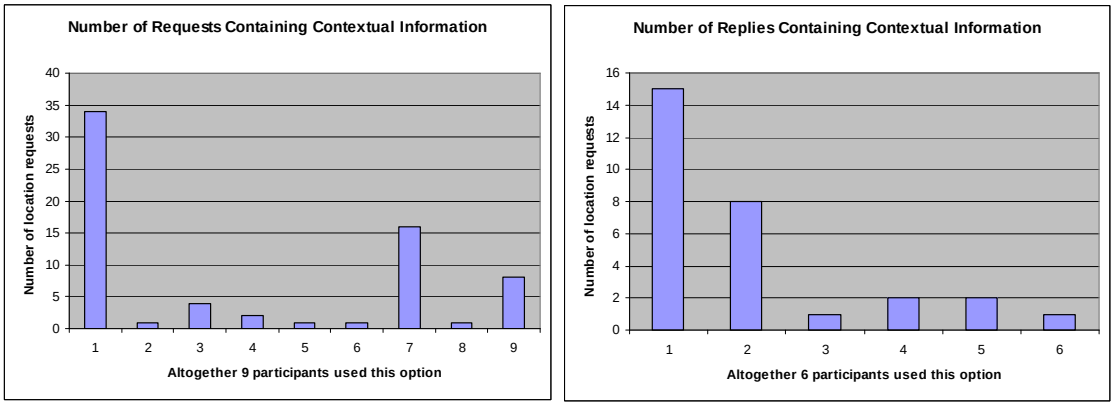


Figure 6.7: Number of location requests containing contextual information in request and reply

providing reason when sending a location request, as one of them commented:

> *Quote (M7): This is very useful and important. When sending a request, I can give some explanation or reason for requesting someone's location. Also useful for replies, to add some explanation for why I am at that location or activities in that place.*

The respondents also mentioned providing contextual information in the reply, e.g., more accurate location information, activities happened in that location, or reasons for accepting or rejecting a request. Here are some comments:

> *Quote (M9): I provided reason for rejecting someone.*
>
> *Quote (M20): I think M1 used it in a reply to me, and it adds a bit more context to the location information. It is definitely useful because the location information can be a bit vague and the accuracy.*
>
> *Quote (M17): In good practice, I may allow someone to get my location but provide something in the response, e.g., 'please avoid locate me after 18:00, because...'. I won't release my location to someone I do not know. For the people I trust, I would let them know but I also expect them to obey the 'social code' or 'gentleman's agreement'.*

Both quantitative data and qualitative comments showed that it is a very useful feature to allow users to provide contextual information with location requests and replies. It enables more effective inter-personal communication mediated by networked systems.

6.3.3 Privacy Rules and User Groups

In total, 31[2] privacy rules were created by 15 participants, of which 26 (84%) were individual rules to process requests from an individual and 5 (16%) were group rules to process request from a user group. Figure 6.8 illustrates the number of privacy rules created by each participant. The maximum number of privacy rules per participant is 6: one participant created 6 individual rules, and another one created 5 individual rules and 1 group rule. 3 individual rules were deleted by two participants: one of the participants deleted 2 individual rules and created a group rule, and the other participant deleted 1 individual rule and created a group rule. Of all the privacy rules, 30 (97%) were created to allow location information disclosure under certain conditions, and only 1 (3%) were created to disallow location information disclosure that was never applied. We concluded that participant tended to create rules to enable location information sharing for their close friends or family members, instead of disallowing requests from strangers automatically. One reason for this phenomenon is that our system by default asks the recipient for accepting or rejecting incoming requests and hence participants found no need to create 'reject' rules. To some extent, it also indicated most participants abided social norms and did not abuse the system by sending random requests to others.

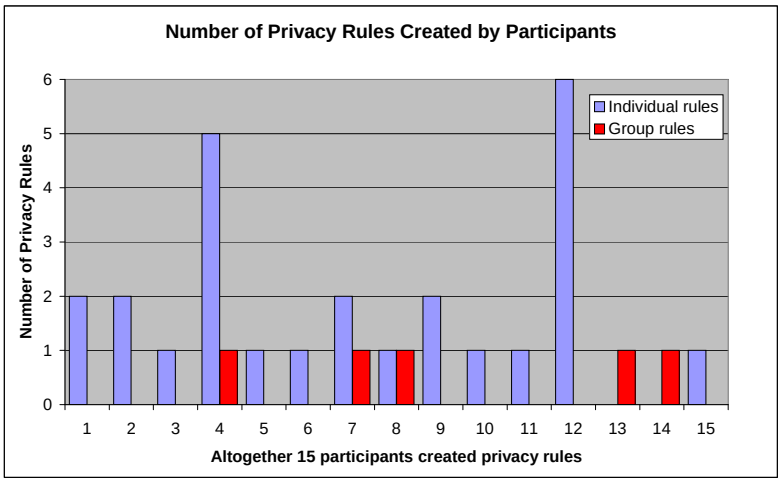


Figure 6.8: Privacy rules created by participants

Figure 6.9 illustrates the number of location requests that have been processed by each privacy rule, where red bars indicate requests processed by group rules. The maximum number of processed requests by a privacy rule is 14, 7 rules that were created did not process any requests, the average number of requests that a privacy rule processed is 2.71 (SD=3.71). Of 31 privacy rules, 13 (42%) created for friends, 8 (26%) were

[2]This figure does not include the privacy rules created for the stranger.

for colleague and friends, 5 (16%) were for family members, 1 (3%) were for a normal colleague, and 4 (13%) for previously unknown people.

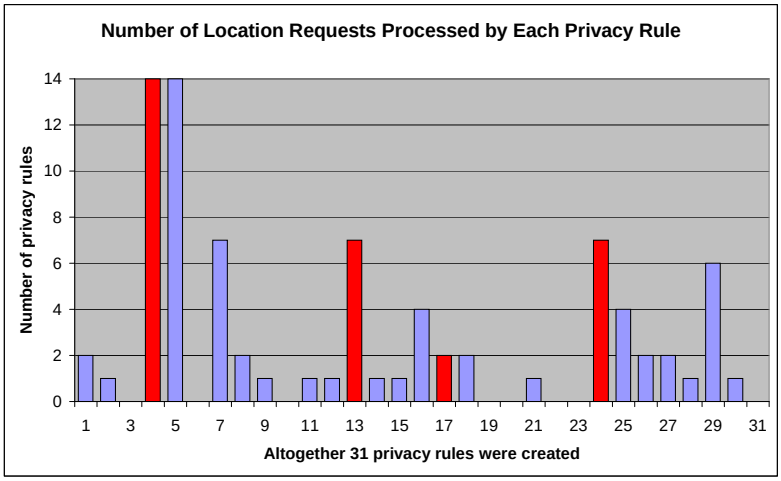


Figure 6.9: Number of location requests processed by each privacy rule

19 (61%) privacy rules were created using the web page, and 12 (39%) rules were created while a participant was using 'always' shortcut on his mobile phone. Only 1 (3%) privacy rule was created by a participant before she received any request. 15 (48%) privacy rules were created to process an incoming request (3 were created using the web page and 12 were created using SMS message). 2 out of 15 (one created using web and another created using SMS message) were created when each participant received the first request, and 13 out of 15 were created after participants received a number of requests (Mean=4.54, SD=4.61). The remaining 15 (48%) privacy rules were created after participants received and processed a certain number of requests (Mean=6.4, SD=8.65), and they were not created to process an incoming request. This supported our hypothesis that in practice people do not pre-specify privacy rules at the beginning of using a system because they tend to experience a system first and then adjust their involvement in privacy management by creating rules over time.

Figure 6.10 shows the number of individuals included in the user groups for the 5 group rules created by 5 participants, where 24 social relationships were created, i.e., adding someone to a user group. We knew that 49 social relationships were created by 7 participants, but the remaining 25 social relationships did not have any effect on handling location requests because there were not associated with any group rules. Three participants created group rules after they created individual rules, two of them deleted the created individual rules and the other one left them in the system. Two participants modified (e.g., added) members in their user group over three days, when their friends

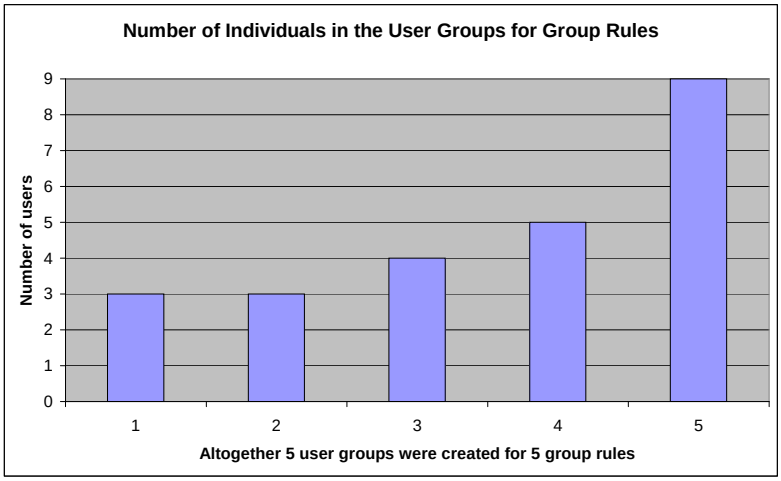


Figure 6.10: Number of individuals in the user groups for group rules

were introduced and created accounts in the system during the first 5 days. 4 group rules were created after the owner received location requests and only one was created before the owner received any request. The evolution from individual rules to group rules and the modification of members in user groups were practical evidence showing that people would modify their privacy preference and adjust the level of openness overtime.

Use of Granularity and Time Constraints

25 (81%) privacy rules allowed location disclosure at the best granularity, and 6 (19%) privacy rules used course-grained granularity, i.e., 2 group rules and 1 individual rule used 'street-level' granularity and 3 individual rules used 'city-level' granularity. During final interview, our participants thought granularity control (Mean=4.08, SD=0.76, 25 responses) in a privacy rule is useful, although only 12 participants realised that they can change granularity of location released by a privacy rule. Only 1 participant (M9) disagreed that granularity control is useful, because he thought course-granularity would not be useful for his friends as he explained: *"If I want to find someone or if someone wants to find me, I found that it is not very useful to know they are just in Lancaster, of course they are in Lancaster because all my friends are in Lancaster. Having a big granularity is not very useful. I am happy to let them know I am in this building or this room. If I am on campus, obviously I am in my office. If I am not on campus, obviously I am at home."* The low usage of granularity constraints in rules was partly due to participants' unawareness of this feature, and partly because the location information returned by the system was not accurate to set any granularity constraint (section 6.4.1).

23 (74%) privacy rules allowed location disclosure at anytime on any day, 7 (26%) rules allowed location disclosure on weekdays (4 allowed anytime and 3 specified certain time period), and 1 rule disallowed location disclosure on certain days and during certain time period. Our participants thought the date and time constraint (Mean=4.12, SD=0.83, 25 responses) in a privacy rule is useful. Only 1 participant (M17) disagreed that the date and time constraints are useful, because of his unpredictable working habit as he explained: *"My days are not predictable enough for that to be useful. To allow someone to know my location before 5:30pm isn't of much value, because I could go home between 4:00pm to 10:00pm and it is completely unpredictable. Contextual information is useful, e.g., to allow someone to find me when I am at work."* We attributed the low usage of date and time constraints to participants' unawareness of this feature, and another reason is that most created rules were for close relationships (42% for friends, 26% colleague and friends, and 16% for family members) and did not need time constraints.

6.3.4 Response to the Stranger

One the 40th day of the trial, we introduced a stranger (named 'Jessika Silversmit') as a web only user into the system, and sent 44 location requests from her account to 22 mobile users. These 22 participants received the first request between 10:00 and 11:00 in the morning, and the second one between 17:00 and 19:00 in the evening (16 between 17:00 and 18:00, and 6 between 18:00 and 19:00). The first requests were accepted by 7 (32%) participants and ignored by 15 (68%) participants. The second request were accepted by 7 (32%) participants, ignored by 11 (50%) participants, and rejected by 4 (18%) participants. Figure 6.11 shows the response to location requests from the stranger by each participant, and it indicates that 9 (41%) participants gave different responses to the two requests. 3 privacy rules were created by 3 participants to allow location disclosure to the stranger: one participant created a rule by replying 'always' to the second request from the stranger, and the other two created privacy rules from the web site after receiving the first request.

The quantitative results of responses to the requests from the stranger were pretty high, which indicated users' good awareness of receiving location requests and convenience of privacy controls. However, the acceptance rate was higher than we had expected, and we asked participants why they made certain decisions for the two re-

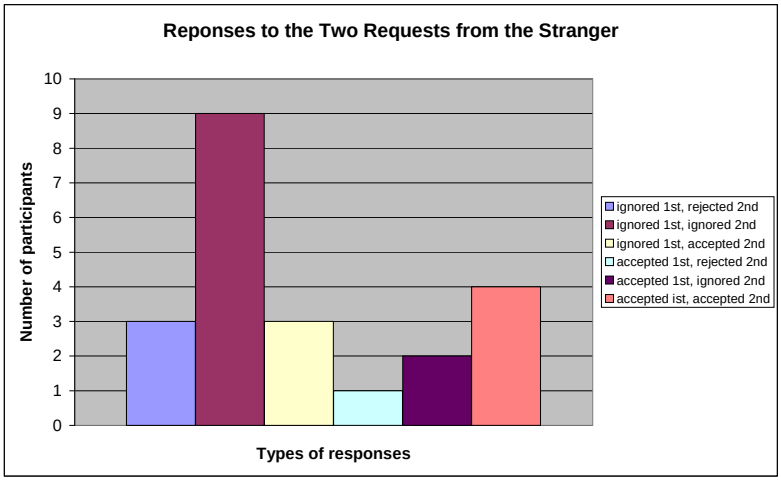


Figure 6.11: Responses to location requests from the stranger

quests from the stranger during the final interview. One participant (M26) commented: *"I ignored the first because I was too late, and reject the second one because I did not know her. I did not create a rule. But if she keeps asking, I would have created one to reject. I thought she might be someone from InfoLab, so I was trying to find out more about her on google. Of course I did not find out."* The main reason for rejecting or ignoring the requests is that participants did not know who the stranger is, and a number of participants provided this reason for rejecting the request in the response. A number of participants searched the name on the web trying to know who she is, or contacted and asked their friends in the system to get more information about her. The high acceptance rate for the requests can be explained using the comments from one participant (M20): *"I accepted the first request, and then I searched on google but could not find the person. Probably because you were doing a trial as well, I had probably been more open than I would be, if it was real commercial application."* We attributed the high acceptance rate to the fact that we were conducting a research experiment and some participants thought the stranger might be someone new to the system and she was just experimenting.

The fact that 9 participants gave different responses to the two requests illustrated our hypothesis that people change their minds about releasing private information, and we discussed it in more details in section 6.5.1. Two participants contacted each other and checked their history of location requests, and found out the stranger might be someone suspicious (see section 6.5.3).

6.4 Reflecting on User Experience

This section presents findings by reflecting on users' experience of the system. We have clustered interesting quotes from a privacy diary system and remarks participants made during the end of trial interview into the following categories: accuracy of the location information, usefulness of the system, cost of the location information, and system features that participants liked about or would like to see improved. From the results of the final survey, our participants agreed that the location disclosed by the system was sensitive information that they would not carelessly disclose to anyone (Mean=4.04, SD=0.72, 26 responses). 1 participant disagreed and 3 were neutral, and they explained that it was mainly because the location disclosed is not very accurate and they do not mind releasing their location to most people at such a level of detail. All 26 mobile users felt that their location privacy was protected by the system.

6.4.1 Accuracy of the Location Information

During the end-of-trial interview, we asked participants their perception of location accuracy. 16 participants (64%, total 25 responses) said the location returned by the system was sufficient to work out where someone was, 6 (24%) participants said it was not, and 3 (12%) participants said it sometimes was and sometimes not. Here are two comments made by participants who said not sufficient:

> *Quote (M23): Not (sufficient) for me, because I live on campus and you cannot tell if I am at office or in my room.*
>
> *Quote (M11): I was in D21 and someone was in D22. The location we got from the system is I was certain miles away from village A, and she was certain miles away from village B. Actually we are just next door.*

The main reason cited that the resolution was not sufficient is that the locatee's active locations were too close be differentiated. We conclude whether the location was sufficient or not largely depends on how far away the person's active locations are. For instance, if the person's working place and home are far apart, the location returned by the system was sufficient to work out whether he is at either location.

Moreover, the accuracy of location information returned by the system is a limitation of the existing implementation, and we expected that it had certain impact on people's

behaviour on disclosing location as one participant (M8) commented: *"One reason that I was so comfortable to disclose my location is because I knew that it was never going be good to any more than a half kilometre. That's absolutely fine. If we had a centimetre granularity location system city wide, I wouldn't necessarily want my student to know how much time I spent in the bathroom. It doesn't seem to bother me that much, but on the other hand it doesn't seem to me that they need to know either. ... I think that my creation of rule and my modification of granularity would have been much more... there would be a lot more stipulations if we had fine-grained location."* We speculate that more accurate location system would trigger more privacy issue and may boost the usage of privacy rules.

6.4.2 Usefulness of the System

Despite the moderate accuracy of the location information, 20 (77%) participants found the location tracking system was useful, and 6 (33%) participants found that it was not. There were a number of real usage scenarios when participants explained why the location tracking system was useful for them during the trial.

The primary usage of the system is not to find the exact location of a user (partly due to the accuracy), but to infer other contextual information (e.g., availability, activity, etc) from the location. Here are some comments related to this type of usage:

> *Quote X (M15) I was trying to find whether M6 is or in Lancaster, because I want to arrange a face to face chat with him. And I got a reply that he is in Nottingham. I did not call him because it was not that important and I just needed to know if he was still on campus.*
>
> *Quote (M4): I used it to see if my friends were in the same city as me. Saving a phone call to make plans if they were.*
>
> *Quote (M9): I was in Manchester and I was wondering where M4 was at about 2:00 in the morning. D was meant to be in Manchester as well. I knew he was in Lancaster with his girlfriend and he said he will be going home to Manchester and I could meet up with him. So I wanted to know where he was. It was 2:00am and he didn't answer his phone, so I thought I could use the system...*

> *Quote (M1): I used to find if M6 is available for lunch together. For me it is not just about location, I use the web site like an SMS web interface. It is free and very convenient that I do not have to call, because I am working on my machine most of the time and it is very convenient to open a browser and click the link instead of texting or calling.*

We conclude that location information returned by the system is a reliable indicator of people's availability or activity and it facilitated further interaction or cooperation between people.

Compared to traditional communication methods (e.g., phone call), most participants felt a main advantage of the system is that it provided location information without incurring much burden or distraction on the requestee. As some participants commented:

> *Quote (M17): Not disturb the people I am finding out. Life is full of distraction, and I would not want to disturb them.*
>
> *Quote (M22): The system is easier, straight forward, and requires less effort, because over the phone you have to say hello and ask where you are.*

Two participants mentioned that they liked the asynchronous mode of communication and they do not have to answer it immediately, as one (M11) commented: *"Handy, could reply during meeting or something. Flexible, I can reply in an hour, do not need to reply immediately."* The system provided a low-key method for finding someone's location that most of the conventional communication methods could not do.

Another usage of the system mentioned by two participants is to use it as a messaging system that is augmented with location information. They called it "contextual instant messaging" and "location augmented messaging", as one (M1) commented: *"The service is like a location-augmented SMS service, because you can do the ordinary texting and get extra location information."* This indicates that the main purpose of inter-personal communication is to fulfil social goals by cooperation, and automatically augmented location information establishes useful context for communication.

There were occasions that participants wanted to find exact location of another one, mainly because the locatees were travelling. Two participants commented:

> *Quote (W4): One time where the information was more useful was when we knew H was about to go to the Manchester airport, and we were able to track that he was on his way there.*
>
> *Quote (M12): This tool was useful for me to track whether my husband reached his new office safely, and on time. ... In someway, it can provide relief from worry, such as during a bad storm, or when a person is late. ... My husband has to go to work at 8:00, and sometime I caused him delayed. He has to travel to his new office in south Manchester, and he is supposed not to arrive later than 9:30. So I wanted to assure myself that he arrives on time. And also sometimes he comes to fetch me, instead of wondering how long it's going to take, I can check the web in every half an hour. Because I can't call him while he is driving, I can just know where he is so that I can estimate and arrange my tasks, and maybe wait downstairs.*

Considering the accuracy of the system, the exact location returned is only useful when locatees move between places that can be differentiated. This is actually the typical usage scenario of some commercial location tracking services that targeted to travelling people, e.g., lorry drivers, travel salesmen or engineers.

An unexpected usage scenario is to locate someone's own mobile phone when he could not find it, as one participant (M13) commented: *"I was working outside (the InfoLab21), and when I came back I found I lost my mobile. I called the number immediately, but no one answered it. Then I think of the system, and requested the location of my phone. I did three requests: the first location I received is somewhere between Bailrigg and Galgate, and the next one is toward A6, and then the third one is pretty much the same as the first one. I wish I could get more accurate location, and would be able to tell if the mobile is still there."* Clearly more accuracy would be useful in this type of scenario, and the above participant wished that he could enable more accuracy on demand. M13 also wanted to send SMS message to his mobile phone from the privacy homepage, in order to disable part of its functionalities or to notify the person who picked the mobile.

Most participants felt the system was easy to understand and use. Many participants liked the integration with Google Maps for displaying location information on a map, so that they can explore the surrounding areas to get more information. Participants felt it was a distinctive feature that they would not get from conventional communication

mechanisms such as phone calling. As unexpected usage of the map system is that that a participant (M12) used the map to find a shortcut coincidentally for her husband, as she commented during the interview: *"I located my husband because he just started working in Manchester one month ago. He used M62 to go to Manchester. I found where he was near Manchester, and at the same time I saw an alternative route (on the google map). ... We found a shortcut! You do not actually look for shortcuts, and you find it by coincidence sometimes."*

In summary, both expected and unexpected usage of the system happened during the trial, and our system has been useful for participants. The main reason that some participants found it not useful during the trial is mainly because those participants did not know many people in the system or they did not need to locate the people they knew, as one participant (M20) explained: *"A part of the problem is that I do not know many people on the list. Probably another part of the problem is that the people I do know on the list were here (on campus)."*

6.4.3 Cost for the Location Information

As mentioned in section 5.4.3, we covered the cost of the commercial service to get participants' GSM-based location information. During the final interview, we asked participants whether they would like to pay for the location information services at the price of 20 to 25 pence for each location disclosure. 13 (50%) participants said they would be willing to pay, and the other 13 (50%) said they would not. Participants who said they would not explained the price is too high because they can make a phone call to the person for similar amount of money. For the participants who said they would, they explained that they would not use the system on a daily basis but the price is acceptable for occasional use. They described situations that they would like to pay for the service, e.g., when they lost their phone, when they are travelling and do not know where they are, when someone does not answer their call, when someone is driving and could not answer the call, etc. Although not conclusive, we found that some people would like to pay for the service under certain situations that they felt useful.

6.5 Evaluation against Requirements

The quantitative results of usage data and qualitative analysis of reflecting on user experience showed that people were using the location sharing application out of real needs and they spontaneous employed the adaptive privacy management system to regulate their location privacy. This section presents the results of evaluating the requirements of adaptive privacy management from chapter 3 as well as whether the design and implementation of the system that meets these requirements.

6.5.1 R1. Adaptive Privacy Adjustment and Evolution of Privacy Preferences

Our participants thought it is important that they are able to respond to changes in circumstances by adjusting whether and how their private information is released (Mean= 4.40, SD=0.58, 25 responses, 1 were neutral, 13 agreed, and 11 strongly agreed). The neutral participant explained there are some situations that he would want to adjust and other situations that he would not. Our participants felt that the system allowed them to make different decisions on disclosing their location information depending on the situation (Mean=4.00, SD=0.59). 18 participants responded to the question (3 were neutral, 12 agreed, and 3 strongly agreed), and the remaining 8 did not answer it because they did not experience during the trial.

Our participants said they preferred to create rules to automate location request processing (Mean=3.96, SD=0.84, 25 responses) rather than interactively processing requests one-by-one (Mean=2.52, SD=1.08, 25 responses), although they wanted both modes for managing their location privacy. 6 participants strongly agreed that they preferred to automate request processing, mainly because privacy rules reduced the effort for dealing with the requests. These 6 participants created privacy rules and received higher number of requests (Mean=18.83, SD=15.33) than average (Mean=8.15, SD=9.72). 2 participants strongly agreed that they preferred to processing requests one-by-one interactively, and both of them received relatively small number of requests (Mean=3.00, SD=1.41). Most participants mentioned that they wanted both interactive and automatic methods for processing requests, because they would use both of them for different people under different conditions. Actually, four participants liked both but found it too difficult to pick a preferred one, and hence they said neutral to both

questions. Here are some comments showing that people wanted both modes:

> *Quote M11: For some person I would like to set up privacy rules. For others, I would still like to reply interactively. It totally depends on who is requesting and what time it is. I want both because I need both. I can't really say which one I prefer.*
>
> *Quote M21: The privacy rules have a certain place, but I also appreciated the flexibility. It was dependent on who you are communicating with or who you believe you are. So I was hovering in the middle there.*
>
> *Quote M12: I would like to process one-by-one, for particular environments or people. For certain people, e.g., my husband, I would like to create a rule. So I want both.*

We conclude that it is very useful to provide both interactive and automatic methods for people to manage their privacy. People tend to use automatic methods for requests from the ones in their stable social groups, and they want to make interactive decisions in situ for requests from others such as unknown people.

Changes in Privacy-related Decisions

When being asked whether they knew how and to whom they wanted to disclose their location before using the system, responses from all 26 participants are mixed (Mean=3.31, SD=1.29). Participants who agreed or strongly agreed said that they wanted their friends and family members to know their location, and participants who disagreed or strongly disagreed said that they did not know who will use the system or who will ask their location. Even for participants who agreed or strongly agreed, many of them mentioned that it was a rough and vague idea and it might not be sufficient to set up privacy rules. One participant (M11) who strongly agreed commented: "I had a basic idea how I would like people to know where I am, for example during office hour. But this is very basic idea. Probably when I need to set up a privacy rule, I might need to think about it much more carefully. It is a rough idea, and it may not be enough to set the details of a rule."

Responses to the statement that participants had changed their mind about disclosing location to an individual during the trial are also mixed (Mean=2.84, SD=1.25, 25 responses). Participants who disagreed or strongly disagreed said they did not change

their minds during the trial but they might if they had used the system for a longer period. Participants who agreed or strongly agreed remembered that they had changed their minds for someone such as the stranger. Recall from section 6.3.4, 9 participants changed their minds in responding to the two requests from the stranger. One participant rejected the first request and accepted the second wrote the follows in the privacy diary:

> *Quote (M23): Very surprised to get a location request today! I looked at the name, and it was from someone I didn't know, with no reason given, which seemed rather odd. I denied it, but somehow felt bad for doing so, and wondered why it would matter that someone who didn't know me would know (roughly!) where I was! Later though, I got another request from the same person, and this time decided to approve it. I figured that it would probably be someone trying out the system, and didn't want to deny them this chance. In a 'real world' situation I probably would have denied it, but as I knew it was likely to be someone from computing it seemed ok. It did make me think that it would be nice to be able to see some sort of details about the person who made the request. As they are a web-only user, I only have a name (which I don't recognise), and not even a number for them. ... If she had put some extra information with the request, e.g., 'I am testing the system' or 'I am new to the department', I would have accepted it in the first place.*

Another participant accepted the first request and rejected the second said the follows during the interview:

> *Quote (M22): I accepted the first one, and ignored the second. She is probably on the list. So I do not care, and I am not bothered to get her know that I am roughly on campus. But when the second comes in, I thought that is a bit strange so I ignored it. ... It is just outside of the working hour and I was on the bus to my home. I am not as happy that people know where I live as they know where I am at work. Since I was on my way home or close to my home, I thought I did not really want this stranger to know where I live.*

The first participant (M23) changed her mind in disclosing her location because of a psychological attitude change to the stranger, and the second participant (M22) her mind because of different sensitivity of location for work and non-work situation.

Although the quantitative results to the survey questions were not conclusive, we found practical evidence that people did change their minds in disclosing private information due to both subjective (i.e., attitude change) and objective reasons (i.e., change in circumstance). It also confirmed our criticism of the static-policy approach because people would change their minds in privacy decisions.

Modification of Privacy Rules

Our participants liked to be able to modify details of privacy rules (Mean=4.45, SD=0.74, total 22 responses; and Mean=4.40, SD=0.83, responses from 15 participants who created rules). All 15 respondents agreed or strongly agreed to the statement except 1 disagreed, and he explained that he did not like because he did not know that he could modify details of privacy rules. When being asked what would make them want to modify privacy rules, the responses can be roughly classified into four categories: changes in social relationship, temporary intimate relationship with someone, special personal events, and changes in one's timetable.

Changes in social relationship involves break up of an existing relationship (e.g., falling out with a friend, splitting with a girlfriend, getting divorced, etc) and establishment of a new relationship (e.g., getting to know someone well and becoming friends, a colleague turning into a friend, getting to know new colleagues, etc). Most participants mentioned that social relationship changes would trigger modification of privacy rules, as some commented:

> *Quote (M8): If I move away Lancaster, then I fall out of communication with certain people and then I may not necessarily want them to know where I am. It depends on my changing relationships with people. In that sense, it would be nice to be able to modify the privacy rules. I know my relationship with people do change over time. ... Recently, I talked to V more and got to know her better (during the trial), and I thought of adding her to my friend group.*
>
> *Quote (M5): You might want to change the details of a privacy rule if your*

> *relationship with someone changes. For example I have a friend, and then we fall out. Then he can sort of track me. ... He might want to find me and beat me up...*

Temporary intimate relationship with someone involves going for a conference with someone, doing a project together and working closely with a colleague, or going somewhere together with someone. 2 participants commented:

> *Quote (M26): If you work very closely with someone for a particular project, even though the person is colleague, if you go to a conference together to another place, I might want for the time of the conference the colleague to know exactly where I was to coordinate with each other, and change it back afterwards.*
>
> *Quote (M11): For example, I just have something to do together with some friends this weekend, but I do not know that when I started using the location system. Then I just knew that this weekend I need to go to Manchester with someone, so at least I would like her to know during the weekend. ... After the weekend, I will change the rule back. It totally depends on the situation, because you cannot predict everything.*

Special personal events that participants mentioned involves going for a holiday, going to some special party, trying to give someone a surprise, etc. Here are some comments:

> *Quote (M15): Maybe I go to some strange place, maybe some kind of party. During this time frame, no body should know my location.*
>
> *Quote (M18): The thing I was thinking of is that I was going for a holiday during the trial. So during holiday, I might want to change privacy rules for that period of time.*

2 participants mentioned changes in their timetable as one reason for modifying privacy rules, as they commented:

> *Quote (M22): For the rule I set, I might change it. Because I do not know my timetable, e.g., when I am busy or doing some thing special.*

Quote (M2): ... I may modify rules when my timetable changes, e.g., very busy this week and do not want to disturbed.

Considering the time scale of trial and rate of change of such relationships, there was only one case of social relationship change. One participant (M8) was getting to know another one (M12) well and thinking of adding her to his friend list (see M8's quote above). Two participants did go to holiday abroad, and one of them thought of disabling a group rule she created, but she did not do that because she knew the location tracking system did not work outside UK. However, there was evidence that participants did modify their privacy rules, and this happened during the first week of the trial where new users were introduced into the system. 2 participants created group rules for their friends and they kept adding their friends to their group during the first week, as one of them (M9) commented: *"The only time that I modified the rules is to add more people to the group of my friends, because they just signed up to the system and I had to add a new person (to the group)."* 2 participants deleted individual rules they created previously and created group rules instead, as one of them (M8) commented: "The only time I had modified privacy rules is when I created a group rule to allow certain people rather than creating a bunch of individual rules for each person. It was easier just to add new person to the group, and then be able to look at that group to see how can access to my location. ... Once I realised that there was some people that I always want to disclose my location, I created a group rule."

Qualitative results from the interview revealed that people want to modify privacy rules. Practical evidence of rules usage showed that people did modify rules. Although we did not see high number of rules modification during the trial, we speculate that there will be more occasions for rule modification if more people were involved in a system and more social relationship changed during the usage.

6.5.2 R2. Awareness of System Behaviour Concerning Privacy

Our participants agreed it is important to be aware when privacy information is disclosed (Mean=4.50, SD=0.58). All 26 mobile users responded to the conceptual question related to the awareness principle, 25 agreed or strongly agreed and only 1 remained neutral. As this participant (M8) explained: *"I do not need to know at the moment it is disclosed, but I do agree that I would be nice to able to go back and get a history of what*

happened." Our participants felt that they were aware of the disclosure of their location information while using the system (Mean=4.40, SD=0.70, 25 responses, 24 agreed or strongly agreed, and 1 disagreed it). The participant (M19) who disagreed to the statement only interacted with the system using his mobile phone, and his mobile phone was broken during the trial so that he can only hear the alert for receiving SMS messages but could not read them to find out who had requested his location. We conclude that the requirement of privacy awareness is important for end users and our system did meet this requirement.

Responses to the statement that "I liked to be informed every time my location was released" are mixed (Mean=3.46, SD=1.10). The participants who disagreed or chose neutral to the statement thought whether they need to be informed largely depends on who was requesting their location, and they do not need to receive SMS notification for location disclosure for the people they knew very well, especially if they had set up privacy rules for their friends or family members. One participant (M13) who said neutral to the statement commented: *"The thing is that it can be overwhelmed when people are trying to track you. As long as you've decided when and by whom you want to be tracked (by setting up privacy rules), then it is fine because you have chosen the criteria. Or you can have a summary at the end of each day, otherwise you will keep getting SMS messages. ... From the psychological point of view, when you are getting (SMS notifications) 5 or 10 times a day, at the end you don't check them. You keep ignoring them, not because you don't care, but because you can't be bothered."* One participant (M9) who disagreed with the statement commented: *"I set up rules for the people I trusted, and I do not really care when they knew I was. I don't find it useful and I just get annoyed because my phone can just hold 10 SMS messages. I set these rules for my friends, unless I fall out with a friend I do not think I would care."* The participant (M8) who actually received highest number of location requests strongly disagreed with the statement: *"(being informed) every single time is annoying. As long as I gave people permission to know where I am, I don't mind them knowing where I am or checking, I don't really want to know. It's more like that I don't want to be disturbed. So the SMS feature that informed me every time my location was released actually bothered me. ... But what I said only applies to the person I know and I had allowed him to access my location. It's a different case if I haven't allowed. For example, for the person that I don't know, I do like to know it on an one-by-one basis."* We concluded that participants do not necessarily want to be informed every time their

information was released although people thought it is important to be aware of private information disclosure.

Multiple Notification Mechanisms

Most participants found it was useful to have multi-modal privacy notifications to their mobile phone, email and web browser (Mean=3.92, SD=0.95, responses, 6 were neutral and 1 was strongly disagreed, and the rest were agreed or strongly agreed). One participant (M21) who strongly agreed with the statement commented: *"When I was at my desk, I would like to receive notification via email or web. But when I was away from office, the main communication is obviously SMS. But I think I appreciated having them, rather than not having them."* The participant (M19) who strongly disagreed to the statement only interacted with the system using his mobile and his mobile was broken. The neutral participants worried about information overload, i.e., receiving too many SMS messages and emails. It is actually the amount of notification (or interruption) that concerns them, not the multiple models of privacy notification. One neutral participant (M16) commented: *"It is nice to access to all, but not send notification all the time."* Another one (M9) suggested notifying him using the most appropriate modality for him instead of three different modals at the same time: *"It is useful for some of them, but it is annoying to have all of them. I generally carry my phone with me. If I made the request via my phone, I want the result back to my phone, I do not need to get the results in my email. If someone was requesting where I am, I do not need to see it both on my phone and my email. I generally see it on my phone first. I interact with email and phone. But my phone goes 'beep beep', and my email is only checked every 5 minutes. So it is annoy to see it twice."*

We asked participants to rank the three notification mechanisms in the system, 20 (out of 25 responses) participants felt SMS message is the most effective way for notification, mainly because they normally carry their mobile phone with them all the time and SMS messages reached them first before the other two mechanisms. As one of them (M20) commented: *"It is because generally my phone follows me everywhere. Although I check my email and browse the web from my mobile as well, SMS is still best because it is available all the time, I do not have to check my email and browse the web in order to receive notifications. ... I did not notice the popup windows. I think that's the problem with the web page popups, because you have to be at the web page. I generally did not*

spend much time on the web page." 6 participants ranked email as the most effective for notification: 3 of them have low-usage of SMS, 2 have medium usage, and 1 has high usage. The main reason for choosing email as the most effective is because they preferred to interact with the system from the email and some of them mentioned that they understand the word 'effective' as 'preferred' here. One of them (M8) commented: *"SMS made me most aware but I didn't like it. Email was my preferred and effective, and email is what I acted upon. When I receive an SMS notifying me that someone is trying to know my location, I almost immediately deleted it or disregarded it. When I saw an email saying there was activity, I would go to the web site and have a look in general what had been happening. As like a macro-view, a bit like having a newsgroup, do like a weekly digest of everything that went on."* The participant (M4) who chose email and had high SMS usage commented: *"It is not as interrupting as SMS when you receive an email."* No participant thought the popup window on their privacy homepage was the most effective way for privacy notifications. The reason cited is the user has to be at the web page in order to see the popup window, and most of the participants did not spend much time on the web page. Another possible reason mentioned by a participant is the settings of the web browser that may block the popup windows.

From the above discussion, we conclude that it is useful to have privacy notifications using multiple modalities. SMS messages were the most effective way for notification, email were second effective and less interruptive, popup windows in web pages were not very effective because it only works when users were on the web page.

6.5.3 R3. Convenient and Timely Access to Privacy Controls

Our participants agreed it is important to have control whenever their privacy information is disclosed (Mean=4.38, SD=0.70). All 26 mobile users responded to the conceptual question related to this principle, 25 agreed or strongly agreed and 1 disagreed. Actually, the participant who disagreed slightly misunderstood the question, because he did not regard privacy rules he set as a kind of control. He (M19) commented as the follows: *"First time when I use the system, yes, (I want to have control). But after I set up privacy rules, I do not care. I do not like to control interactively, because I have privacy rules to control. ... I do not see rule as a means of control here."* From the quantitative results, we conclude that the principle of privacy control is important for end users.

Multimodal Privacy Controls

Our participants felt strongly it was very useful to have privacy controls on both the phone and the web (Mean=4.56, SD=0.51, 25 responses either agreed or strongly agreed), or more generally to have multi-modal privacy controls on multiple devices. Here are some commented from the participants:

> *Quote (M21): When I was in front the machine, I personally prefer to use the web. But when I was away from my machine, SMS would be a replacement of that.*
>
> *Quote (M8): It is nice to have the privacy control on my phone and the web interface. I tended to use the one on the computer because it is easy to use. But if I was on the move or away from my computer for a while, then it is handy to be able to do it with my phone. (Having privacy control) available on the phone is good.*
>
> *Quote (M2): I always sent requests using the web, and accept or reject them using SMS.*

A number of participants mentioned that multimodal interaction is a distinctive feature of the system they liked about, as one of them (M13) commented: "like the fact that I can use both web and SMS interface. If it was just SMS interface, I would not use it very much because I am not a very mobile phone typing person. Since I spent lots of time with my computer, I tend to use the web interface a lot. But for some person who do not use internet very much, probably the SMS interface is more useful for them."

Participants who used the web interface for privacy control found it was very easy to use (Mean=4.55, SD=0.51, 22 responses either agreed or strongly agreed), and many participants mentioned they tended to use the web interface if they were near the computer. Our participants found it was easy to use SMS messages to accept or reject location request, but not as easy as the web interface (Mean=4.00, SD=0.90, 23 responses, disagreed and 3 were neutral). Some of the participants mentioned that they were not used to typing text messages, and therefore SMS interface is not as convenient as the web interface.

However, we found a usability problem with the SMS reply to the location request, especially when the requestee wants to append extra information to the reply. One

participant (M5) who disagreed to the statement commented as the follows: *"I was on the train and I got a request from M1 saying 'are you lunching?' I wanted to reply 'yes' but I said 'no'. I was still thinking of the question in the request. I should have text 'yes' to release my location and append 'no' to answer the actual question. That confused me a bit."* Another participant (M6) left two privacy diary entries for the same issue: *"Got the location request during the exam — should've had it on silent! Replied yes, etc., but realised I needed a yes for the location part and to the purpose question, i.e. yes, yes invigilating etc. was uncertain as to whether yes, yes, invigilating would work..."* One participant (M8) suggested other controls on the mobile phone instead of SMS: *"It is nice to have the privacy control on my phone, but the SMS isn't the ideal interface. It would be better to have some application running where you set a slider bar or tick a box with the phone interface."*

From the above discussion, we conclude that providing multimodal interaction supported the principle of convenient and timely access to privacy control. Both the web and SMS interface of our system were easy to use, despite of a small usability problem of SMS interface that could be further improved.

6.5.4 R4. Balance between Privacy and User Involvement

Our participants agreed it is important adjust user involvement in privacy decisions to reduce effort and intrusiveness (Mean=4.48, SD=0.51, 25 responses). They also agreed that the system allowed them to find an agreeable balance between the effort and interruption (Mean=4.05, SD=0.67 for total 21 responses; Mean=3.93, SD=0.79 for 15 participants who created privacy rules). One participant disagreed and one said neutral, and they both thought the system reduced the effort but not necessarily interruption because of the SMS notifications. One of them (M8) commented: *"I was fine with the effort, because the cognitive load of processing a request was reduced. But there was always interruption. Actually during the trail of the system, I have put my phone into silent mode, so that I wasn't interrupted when someone made a request. ... But for the same token, I kept it with me and on and charged it all the time. ... So it changed the way of my behaviour with the phone in many ways."* It indicates that the usefulness of the system out-valued the interruption incurred on him. From the responses to the survey questions and evident form the spontaneous usage of privacy rules, we conclude that people want to balance privacy management and user effort and intrusiveness. Our

system allowed participants to do that by using privacy rules, and our participants felt the system allowed them to find a good balance between effort and intrusiveness.

Usefulness of Privacy Rules

Our participants thought privacy rules were useful for them (Mean=4.42, SD=0.61 for total 19 responses; Mean=4.47, SD=0.65 for 15 participants who created privacy rules). All 15 participants who created privacy rules either agreed or strongly agreed that privacy rules reduced the amount of interaction involved in privacy management (Mean=4.47, SD=0.50). Most of them agreed that privacy rules reduced the amount of interruption (Mean=3.53, SD=1.35). Within 15 participants, 1 participant (the one who made the above comments) strongly disagreed, 3 disagreed, and 1 said neutral, and that's mainly because they still received privacy notifications via SMS messages even though they created privacy rules.

All 15 participants found it was very easy to create privacy rules via the web pages (Mean=4.47, SD=0.47). 8 participants had created rules via SMS messages, and they found it was easy to do as well (Mean=4.13, SD=0.84), although 2 of them remained neutral mainly because they were not used to typing text messages on their mobile phone. 13 participants responded to the statement whether they found user groups and group rules useful, and all of them agreed or strongly agreed except one said neutral (Mean=4.38, SD=0.65). For the 5 participants who did create group rules, they all agreed or strongly agreed to the statement (Mean=4.40, SD=0.55), because the system allowed them to easily manage group members. For the 15 participants who created privacy rules, they had a good awareness of their privacy rules, and 12 of them remembered the exact number of rules they had created. One participant thought he had created a rule but he actually did not. Another participant thought he did not created any rule but actually he replied 'always' to two different requests and hence created two rules. He explained that he thought privacy rules were not just 'allowing' or 'disallowing' but more fine-grained controls.

The main reason that some participants did not create any privacy rule or any group rule is that they felt there were not enough location requests for them or not enough people in the system they knew to justify the creation of rules or groups, and most of them said they would have done if they received more requests or introduced their close friends or family members into the system. 15 participants who created privacy rules

received an average of 11.93 requests from others (SD=11.26), and the remaining 11 participants received an average of 3.00 requests (SD=2.79). It indicates that privacy rules were more useful for participants who received higher number of requests.

From the above discussion, we conclude that privacy rules were useful for reducing user effort and intrusiveness of privacy management. Our system provided usable methods that facilitated users to create rules and allowed users to have a good awareness of their created rules.

Creation of Privacy Rules

All 26 participants responded to the question to rank when they prefer to create privacy rules, 16 (61%) participants preferred to create rules after they received and processed a few requests, 8 (31%) participants preferred to create rules before receiving any requests, and 2 (8%) participants preferred to create rules when they are receiving and processing a request. Participants preferred to create rules beforehand mainly because they wanted to automatically disclose their location to their friends or family members, as one participant (M26) explained: *"I tend to know in advance how I want the system to be used. I know who I want to disclose... But by default, I would ask any request sent to me basically (to manually process). For only special people, e.g., for my wife, it is ok that she should know where I am at any time, for friends I want them know if I am on campus, but no more, for my family, I want them to know if I am in UK or not. So it is not only white or black whether they are allowed or not, or when. There is also how precise the location information is that they can obtain."* One participant (M3) preferred to create rules afterwards commented: *"I have to get familiar with the system first, on how it releases location information for example. And after I received more requests, I would think of creating privacy rules to improve efficiency."* Another participant (M7) said: *"I do not know beforehand who is on the user list, and the list is growing during the trial. I do not know beforehand what rules to create. I only created rules when I knew someone was interested in knowing my location."* We conclude that the main reasons that participants preferred to create rules afterwards are: they have to be familiar with the system on how it would disclose their location, they do not know beforehand who they should release location to and what rules to create, and they only create rules when there is a need to, i.e., someone was interested in knowing their location and sent them requests.

For the 15 participants who created privacy rules, 10 (67%) preferred to create rules after they received requests, 4 (27%) preferred to create rules before they received any request, and 1 (4%) preferred to create rules while processing a request. Accompanying these responses to the quantitative results in section 6.3.3, we found that what participants said is not completely consistent with what they actually did. The logged data revealed that 10 participants created rules after they had processed some requests, 8 participants created rules while processing an incoming request, and only 1 participant created a rule before receiving any request (note: 4 participants were in the first two categories). No participant who preferred to create rules before receiving any request or while processing an incoming request actually did what they said. 10 participants who preferred to create rules after they received requests acted most consistently: 7 of them did what they said, and 3 of them created rules while they were receiving and processing a request. The fact that what people thought they will do is often inconsistent with what they actually did can be illustrated by an incident happened during the interview, where a participant (M20) changed his mind in ranking the options (from 'before' to 'after') for this question: *"I'd like to have the rule in place before requests are coming in, so that I can decide how I could manage requests in advance. I could probably define what sorts of rules I'd need. I have not just because I did not use as much as I thought. ... If the requests become more frequent, maybe I would have set rules. ... Yes, you might be right. I think my natural response is to say that (before received any request), but in practice it is probably that (after received and processed a few requests)."* Although some participants thought they wanted to create rules beforehand, they were actually 'lazy' in creating rules before receiving any request. A possible explanation of this discrepancy is: people tend to take *"path of the least resistance"* in privacy management that is not the primary purpose of using the service, and hence in practice they only create rules when the benefits of creating it (i.e., reduced effort and interruption) overweight the cost of doing it (i.e., effort of creating rules).

From the quantitative usage data and qualitative results about privacy rules, we conclude that in practice people do not pre-specify privacy rules at the beginning of using a system. The results confirmed our criticism of static-policy approach in chapter 2 and supported our hypothesis of adaptive privacy management: they tend to experience a system first and then adjust their involvement in privacy management by creating rules over time.

6.5.5 R5. Accountability for Privacy-related Behaviour

Our participants agreed it is important to make the behaviour of private information disclosure accountable, i.e., to maintain a record of what information people found about them (Mean=4.2, SD=0.82, 25 responses, 3 neutral and 1 disagreed). Three participants mentioned the record might be important and useful for legal reasons, as one of them (M9) commented: *"If there is a record of where I am, it might be useful for legal reasons. I meant to commit a crime in Manchester but I was in Lancaster. ... Or maybe if my friend wasn't here, I can say why you were late I know you are at such as such location."* 2 participants who said neutral thought it was not important because they did not care for most of the people, and another one said it was not necessary because she had already known the location disclosure by accepting the requests. The participant (M17) who disagreed commented: *"I do not think that's important (to record the fact you used it), because that's yours. I gave you permission to access it before you took it, then it is yours. So I would expect travel log before you took my location."* Since the fact that someone had accessed another one's private information is a piece of shared information, it is an interesting debate on whose privacy should the system protect. It is more of a legal and public policy issue, and it is out of the scope of the thesis. However, we can conclude that the principle of accountability for privacy-related behaviour is important for end users.

Usefulness of Privacy History

24 out of 26 participants were aware that the system maintained a history of every location request they made or received, and most of them felt more comfortable knowing the fact (Mean=3.57, SD=0.73, 23 responses, 1 strongly agreed, 13 agreed, 7 said neutral, and 2 disagreed). One participant who disagreed had worried about someone else might have access to the history, as he (M21) commented: *"I was assuming it meant long-term history. It would be nice to know that it (the history) was limited to a certain amount of time, e.g., one month or three month. Who have the access to that information? I do not particularly want my boss to know where I was having lunch."* A main reason for the other participant who disagreed and some of the participants said neutral is that they were aware when their location information was disclosed and therefore the history did not make them feel more comfortable. Another reason mentioned by three participants who said neutral is that they knew we were conducting a research experiment and

they were informed of the history at the beginning of the trial. They commented as the follows:

> *Quote (M4): Same for me, have it or not. Because my location information has already been kept by you (for the experiment).*
>
> *Quote (M19): I knew the system is logging location requests for the trial. If I was bothered that the system logged the requests, I won't have used it in the first place. Because I knew it was doing it, whether it does or not kind of influences the question. So I know it does, whether I am comfortable is not the point. If I wasn't comfortable, I won't have used the system in the first place.*
>
> *Quote (M16): I guess I trust you that you are doing an experiment and it is just for your studies. If it is a company I might be more worried. If the data is kept on my machine, it might make me feel better because I know no one has access to it.*

It is important to acknowledge a limitation of the study is that people knew there were conducting a research experiment and it may influence some of their reactions to the system such as the responses to the history.

13 out of 26 participants said that they did check the history of location request and disclosure. Participants who did not check said they did not think there was a need to do that because they had already known the location disclosure and they do not care any more. Some of them said it was mainly because there were small number of location request and disclosure for them, and they would have checked if they were frequent user or they have used the system for longer. For the participants who did check the history, most of them said they were just out of curiosity and interest, and some of them did it more frequently said it is useful to review the history periodically to know who had requested their location. Here are some comments:

> *Quote M8: I would look at the web site like every two or three days and see how much activity there has been and how many requests have been made to know my location. Just a glance, like the way I looked at my credit card bill, I just glance it just to make sure that there is nothing unordinary.*
>
> *Quote M7: I can review the location requests and disclosure, to see who had requested my location and who is interested in my location. If there are*

lots of people I knew such as relatives, friends, and colleague, it would help me to understanding the trends in social relationships.

Quote M20: It is useful to go back and see what requests have been made. Or maybe useful is a wrong word. It is interesting to look back over times to see who made requests to me, when, possibly why. I looked back this morning.

However, one participant did use the history of request to detect unusual behaviour happened in the system, i.e., finding out that there was a stranger sending requests to everyone. The participant (M9) commented: *"I knew you introduced a fake person 'Jessika'. My friends D got a message from her in the morning and I got a message as well. He sent me a message from the Internet saying 'hi, did you get a message from so as so? Do you know who she is?' I said I didn't know, and I checked my history then to see if I also got a message from her at the same time. And we realised that we both got a message from her at exactly the same time. We thought there might be something suspicious there."* The above evidence showed that history of privacy behaviour is useful for detecting usual behaviour related privacy, although it was not particularly useful for privacy purpose during the experiment. We speculate that it would be more useful for extended use of real-world applications.

6.6 Discussion

6.6.1 Key Findings

Section 6.3 presented general finding of the user study and section 6.4 evaluated requirements of adaptive privacy management. The key findings of the evaluation can be summarised as follows:

- R1: Our participants thought it is important that they are able to respond to changes in circumstances by adjusting whether and how their private information is released. Our system provided both interactive and automatic methods for people to process private information requests from different person under different situations, and our participants found it was very useful to have both methods. We found practical evidence that people did change their minds in disclosing private

information due to both subjective (i.e., attitude change) and objective reasons (i.e., change in circumstance), and it confirmed our criticism of static-policy approach. Our participants liked to be able to change details of privacy rules when social relationship changes, special events happen, or personal timetable changes. Our system supported creating privacy rules using both web pages and SMS messages, and supported modifying privacy rules using web pages. We also found evidence that participants changed privacy rules when their friends were introduced into the system and participants created group rules to replace individual rules, and it supported our hypothesis of adaptive privacy management and privacy preference evolution.

- R2: Our participants thought it is important to be aware when privacy information is disclosed. The system promoted users' awareness of privacy by notifying them using three different mechanisms, e.g., SMS messages, email, about popup alert windows within web pages. Our participants felt it was useful to have privacy notifications using multiple modalities, and most participants thought SMS messages were the most effective. Our participants felt they were aware of the disclosure of their location information while using the system. However, they do not necessarily want to be informed every time their information is released, which motivates us to propose incorporating configuration of awareness mechanisms into privacy rules discussed below.

- R3: Our participants thought it is important to have control whenever their privacy information is disclosed, and they felt it was very useful to have multi-modal privacy controls on multiple devices. The system provided convenient and timely access to privacy control both on both the web and the phone, and our participants found both privacy controls of our system were easy to use. Some participants mentioned that the privacy control on the mobile phone can be improved and extended beyond basic SMS messages.

- R4: Our participants agreed it is important adjust user involvement in privacy decisions to reduce effort and intrusiveness, and they felt that the system allowed them to balance between the effort and interruption. The system allowed creation of individual and group privacy rules to adjust user involvement in privacy decisions. Our participants found privacy rules were useful to reduce the amount of user effort and intrusiveness, although some participants felt they were still

interrupted by SMS privacy notifications. Participants who created rules in the system found it was easy to create privacy rules both via the web pages and via SMS messages, and participants had good awareness of privacy rules they created. Both quantitative and qualitative analysis showed that in practice people do not pre-specify privacy rules at the beginning of using a system and they tend to experience a system first and then adjust their involvement in privacy management by creating rules over time. The results confirmed our criticism of static-policy approach and supported our hypothesis of adaptive privacy management.

- R5: Our participants thought it is important to make the behaviour of private information disclosure accountable, i.e., to maintain a record of what information people found about them. The system maintained a history of location request and disclosure centrally in the middleware platform, and the history can be used by multiple applications. Most of the participants felt more comfortable knowing that the system maintained a history of location disclosure and request for them, although we have to take into account the fact that the research experiment environment may affect their responses. The history was not particularly useful for privacy purpose during the experiment, although some participants regularly scanned though the location request and disclosure to be aware of what had happened to them. However, two participants have used the privacy history to detecting unusual detecting usual behaviour in the system, and hence we speculate that it would be more useful in long-term usage of real-world applications.

6.6.2 Limitations

When looking into the results from the study, we have to take into account the following limitations of the system and the study:

- Our system provided location information at an accuracy based on the GSM cell tower density (approximately 0.93km – 5.59km around Lancaster area), and this may affect participants' attitude toward privacy and the usage of granularity constraints in privacy rules. In addition, the system can only provide location information of mobile phones in UK, and it may affect participants' behaviour of disclosing their location when they were abroad, e.g., during holiday or conference.

- The fact that people knew they were involved in a research experiment may have influenced their behaviour. For example, they tended to be more open than normal when disclosing location to others, e.g., unknown people. Since they knew the system recorded their interactions for research purpose, some participants' attitude toward privacy history may have been influenced.

- The user study involved 30 participants and various typical social relationships, and the system has been used a period of $7^1/_2$ weeks. Although the user population is not very large, their privacy segmentation was quite typical and consistent with the ones in large-scale studies in US and UK.

6.6.3 Reflecting on Developer's Experience

Reflecting on our experience of developing and deploying the adaptive privacy aware location sharing application and extending the functionality of the platform to better support the user trial (e.g. changing service providers, introducing web users), we found that there is limited evidence that the platform is both flexible and extensible. We provide the following samples as evidence for it:

- Our system had only one asynchronous notification mechanism, i.e., email, for privacy notification when the NotifyHandler (P5 in figure 5.5) plug-in interface was specified. To enable users to receive privacy notifications while they are mobile, we implemented the SMS notifier plug-in using the above interface and the change took 2 days. The implementation of the plug-in is mainly a TCP client that communicates with the SMS gateway using a proprietary protocol to instruct it to forward a privacy notification to a user's mobile phone. The plug-in mechanism allowed us to concentrate our main effort on the specification of the proprietary TCP protocol.

- Originally our system did not support negative rules (i.e., rules to reject requests under certain conditions): requests that do not conform to the conditions of a positive rule will be rejected, and requests that cannot be processed by any rule will be sent to the recipient. We added the support for negative rules after we examined the expressive power of similar systems for configuring firewalls and file system access control permissions in NTFS. This change mainly affected on

our implementation of the FindPrefHandler plug-in (P1 in figure 5.10) for selecting an applicable privacy rule for processing a request. The new plug-in requires resolving conflicts between negative and positive rules (the detailed description was presented in section 5.3.4). This change took 3 days to complete.

- During the first week of deploying the location sharing application, we added support for web only users to meet the real demand from one of the trial subjects (see section 6.2.2). The introduction of web only users does not affect most of the functionality of the system; web only users only have a subset of the functionality of the system available to them (they are able to track users, but not be tracked or consequently set rules for incoming privacy requests). The only change needed was to have different privacy notification messages for web only users, because some of the options in the original messages did not apply to them. We created two new message templates for notification messages (see Appendix E) when requests from web only users were accepted or rejected, and the template mechanism allowed us to change messages without having to modify much of the source code.

- The first third-party location service (i.e., world-tracker) was out of service as we entered the user trial phase, claiming that they were waiting for verification from UK network operators. After delaying our trial for nearly two months, we reluctantly decided to switch service provider and selected FollowUs. The architecture allowed us to integrate the new location service with our system in less than 3 days, and the main work was concentrated on connecting to the service web site and converting location information in HTML format to the XML format we specified in section 5.4.3.

We are clearly unable to infer from this evidence that the system is indeed flexible and extensible in the general case (e.g. when applied to new problem domains or applications by 3rd party developers). However, we hypothesise that given the number of reasonably significant changes we made to the system during the late development and early deployment stages outlined above, and the ease with which they were integrated, does at least intimate that the system promotes a reasonable degree of flexibility and extensibility as we intended in our design.

6.6.4 Suggestions for Improvement

From the experience of deploying the location privacy system and feedbacks from participants, we have summarised the following areas that can be used to improve the system. Some of the suggestions were not specific to the system but more general to the implementation of adaptive privacy management.

- The existing implementation of privacy rules enabled automation of private information request processing, but it does not control the privacy notification generated after information disclosure. By incorporating elements for configuring privacy notification mechanisms (e.g., what type of notification, frequency of notification, etc) into privacy rules, the implemented system would better support R2 and R4.
- The usability problem of SMS messages can be improved by redesigning the syntax and semantics of commands in replying requests. Moreover, other interaction methods on phone (e.g., GUI) can be incorporated to improve the usability of phone interfaces and hence provide better support for R3.
- To better support R4, the system could allow additional conditions to be incorporated into privacy rules, e.g., location, activity, personal calendar, etc.
- Additional functionalities and commands can be introduced to facilitate creation and modification privacy rules on mobile phone, e.g., creating a rule without having to respond to a request, reminding user to create rule when next online, enabling or disabling all rules created. In addition to enabling people to modify rules, the system could facilitate people to switching between privacy rules they created. One suggestion is that users could create privacy rules for different situations or modes, e.g., invisible to all, visible to all, visible to friends, etc, and be able to switch between these modes very easily. Both suggestions can be employed to improve R1.
- Instead of maintaining history of information requests and disclosure, the system should also make the history more useful and usable, e.g., providing summary of recent requests and disclosure instead of a one-by-one list, automatically detecting unusable requests and disclosure, providing search facilities, etc. This does not directly support R5, but it is related and aims to make use of the accountability mechanisms to improve users' privacy.

- The final suggestion is to improve "plausible deniability" support in the system. The existing system supported "plausible deniability" by allowing users to ignore requests instead of explicitly rejecting them. However, ignoring requests might lead the requester to think that you are deliberately ignoring his request. If people do not want to release location information but they do not want to offend the requester (by either rejecting or ignoring), the system could provide options for them to choose different return information, e.g., the mobile is switched off, the mobile is out of the network, etc.

6.7 Summary

This chapter discussed the user study of the adaptive privacy aware system that allows users to preserve privacy while sharing GSM-based location information. The user trial was conducted during April to May in 2007 over a period of $7^1/_2$ weeks and involved 30 participants. The chapter started presenting experimenting methodology of the three phased user study, followed by general findings related to the location sharing application, including participants profiles, quantitative results of system usage, qualitative results reflecting on experience of the system, and both quantitative and qualitative analysis of responses to the location requests sent from the stranger we introduced just before the end of the trial. We found the principles of adaptive privacy management are important for end users to manage their privacy management, and the design and implementation of the existing system did meet all the requirements. Finally, the chapter provided an objective discussion reflecting on the strengths and limitations of the implemented system, and provided suggestions to improve future implementation.

CHAPTER VII

Conclusions

Contents

7.1 Overview

This thesis has presented an investigation into the issues concerning prevention of privacy intrusion through accidental or negligent sharing of personal information in applications that enable the intentional sharing of private information in networked computing environments.Following Palen and Dourish's observation that privacy management is *a dialectic and dynamic boundary regulation process* [Palen03], we have argued that no set of pre-specified privacy rules or policies can meet users' changing requirements for privacy in networked environments due to changes in context and setting. In response we proposed *adaptive privacy management* where the user and/or a system continuously adjusts the disclosure of personal information according to the user's changing desire for openness. We identify the requirements for adaptive privacy management, and propose a design of a corresponding middleware platform to support them. We report on a prototype implementation that demonstrates that the proposed requirements can actually be met and do support the adaptive approach. Both the principles of adaptive privacy management and the prototype implementation were evaluated based on a 53 day user study using a location sharing application built using the adaptive privacy management system. More specifically:

Chapter 1 introduced the concept of privacy and established the target domain and scope of the thesis. The chapter provided Westin's definition of information privacy, and motivated the need for privacy in networked computing environments. Next, the chapter explored the technological impact on privacy and how privacy is impacted as we strive for the Ubicomp vision. The chapter then defines 'adaptive privacy management' and presents the aims and objectives of the research.

Chapter 2 provided important context for the thesis by exploring the issue of privacy from historical, social, legal and technical perspectives. The chapter presented an in-depth investigation of existing technical mechanisms for privacy support. The result of this investigation motivated the need for adaptive privacy management, by which a user and/or a system continuously adjusts the system's disclosure of personal information according to the user's changing desire for openness under different circumstances.

Chapter 3 presented an analysis of the possible limitations of existing technical approaches. The chapter reviewed different design strategies for information privacy solutions, and explained our rationale for selecting specific strategies for adaptive privacy

management based on a critical analysis of technical approaches in each category. The chapter concluded by identifying the set of requirements that should be satisfied in order to develop adaptive privacy management for personal information sharing applications.

To meet the requirements we proposed, chapter 4 presented the design of a middleware platform to simplify the construction of adaptive privacy aware applications. We started by identifying the important design features for incorporating adaptive privacy management into private information sharing applications. The chapter motivated the need for a flexible middleware platform to support the development of adaptive privacy management, and presented an architectural design for application interactions with such a platform. The flexibility of the platform enables developers to customise its behaviour by developing plug-ins with different policies or algorithms, in order to meet the needs of different problem domains.

Chapter 5 presented the prototype implementation of an adaptive privacy management system, i.e., an adaptive privacy manager, that was developed using our platform. The chapter contributes the core API methods exposed by the platform that are required for adaptive privacy management. We also identify the plug-in interfaces for customising and extending its basic functionality, as well as the internals and operations of the platform and algorithms employed for developing plug-ins. Finally, the chapter discussed a proof-of-concept location sharing application integrated with the adaptive privacy manager, in order to demonstrate the feasibility of the architecture and illustrate the workings of the platform.

We presented the evaluation of the principles of adaptive privacy management as well as the design and implementation of the prototype system in Chapter 6. The evaluation was based on the experiences gained from a deployment and end user trial of the location sharing application with 30 participants over 53 days. Quantitative results from logged usage data and Likert-style survey questionnaire were analysed, and qualitative results from the interviews and daily on-line privacy diaries were discussed. The evaluation concluded that all five requirements for adaptive privacy management *are* important for end users, and the implemented system *did provide support* for all these characteristics. The chapter provided an objective discussion reflecting on the strengths and limitations of the implemented system, and provided suggestions to improve future implementation.

The remainder of this chapter presents our conclusions by highlighting the major

and minor results of the thesis, discusses potential future research directions related to this work, and presents our concluding remarks.

7.2 Major Results

This section reviews the major results of the work presented in the thesis. The sequence of the results presented in the following sections is based on the order they appeared in the thesis and does not imply any ranking of importance.

7.2.1 Identification of Adaptive Privacy Management

An important contribution of the thesis is the identification of the limitations of existing systems in supporting users to achieve better privacy in networked computing environments and our proposal of adaptive privacy management. In particular, the thesis presented the following results concerning the identification of adaptive privacy management:

- Investigated a number of projects that have taken the static-policy approach for user-transparent privacy negotiation with networked applications, and provided pragmatic evidence showing that the static-policy approach failed to enable users to efficiently and effectively adjust the level of openness according to their changing desire for privacy in different situations.
- Studied work on theory of privacy and provided theoretical evidence showing that privacy management is "not about setting rules and enforce them" but rather "a dialectic and dynamic boundary regulation process" and "the continual management of boundaries between different spheres of actions and degrees of disclosure within those spheres" in a networked world.
- Combined pragmatic and theoretical evidence to demonstrate that the desired end result of information privacy management is not about keeping personal information hidden but rather selectively disclosing personal information to fulfil our social goals,and proposed our own definition of better privacy as *"enabling personal information disclosure at a level of openness that is as close to a user's desired level to assist him/her in accomplishing useful tasks."*

- Motivated the need for adaptive privacy management and defined it as *"the process that a user and/or a system continuously adjusts the system behaviour of disclosing personal information according to the user's changing desire for openness under different circumstances in dynamic environments."*

7.2.2 Requirements for Adaptive Privacy Management

In this thesis we analysed different design strategies for information privacy, including control at point of collection, anonymity and pseudonymity, awareness and accountability, and control at the point of use, and identified rationales for selecting specific technical mechanisms to support the development of adaptive privacy management. Followed this analysis, the thesis presented the following requirements for adaptive privacy management:

- **Adaptive Privacy Balance and Evolution of Privacy Preference (R1):** to enable users or/and the system to adjust the balance between openness and closedness depending on situations in dynamic networked environments; and to allow evolution of users' privacy preferences specified in the system over time as a result of on-going interactions between the user and the system.
- **Awareness of System Behaviour Concerning Privacy (R2):** to promote users' awareness of system's behaviours concerning privacy, e.g., what the system can potentially and/or actually do with users' personal information.
- **Convenient and Timely Access to Privacy Controls (R3):** to provide end users with convenient and timely access to privacy controls, in order to encourage them to adjust the system's behaviour regarding their personal information disclosure, in response to changes of circumstance.
- **Balance between Privacy and User Involvement (R4):** to balance end users' need for information privacy with the level of effort and intrusiveness incurred by privacy-related interactions.
- **Accountability for Privacy-related Behaviour (R5):** to maintain audit trails for privacy-related behaviours (e.g., information disclosed either explicitly by the user or automatically by the system) to increase accountability and traceability of the system.

7.2.3 Feasibility of Adaptive Privacy Solution

The thesis presented evidence demonstrating the feasibility of adaptive privacy management for the target domain:Specifically, the creation and evaluation of a prototype implementation based on our architectural design for supporting adaptive privacy management, and the results of the evaluation of this prototype with end users. The detailed results concerning feasibility of adaptive privacy solution are:

- Presented a prototype implementation of an adaptive privacy management system providing multi-modal interaction via the web/SMS. The system was built using a middleware platform that supports constructing adaptive privacy aware applications. This demonstrated that adaptive privacy solutions can be built using the platform.
- Integrated the adaptive privacy management system with a location service and developed a proof-of-concept application that enables end users to share GSM-based location information and preserve location privacy using the adaptive approach. This demonstrated that adaptive privacy solutions can be employed to create privacy aware applications.
- Deployed the location sharing applications integrated with the adaptive privacy management system and conducted a three-phased user study based on the deployment. This demonstrated that adaptive privacy solutions can be employed by end users to manage their private information.
- Evaluated the principles of adaptive privacy management and the prototype implementation based on the findings from the user study. This demonstrated that the principles are important for people in managing their privacy, and the prototype implementation did support these principles and hence helped people to achieve better privacy.

7.2.4 End User Study and Evaluation

The final major contribution of the thesis is the deployment and the three-phased user study of the location privacy system, which has been used by 30 participants in their everyday lives over a period of 53 days. Based on quantitative results and qualitative findings, we evaluated the principles of the adaptive privacy management as well as the

design and implementation of the prototype system. The results concerning the end user study and evaluation (listed here by requirement number) consist of:

- R1: Our participants would like to make privacy decisions in response to changes in circumstances and our system provided both interactive and automatic methods to support it. We found practical evidence of people adjusting privacy balance and it confirmed our criticism of static-policy approaches. We also found evidence that participants modified privacy rules, supporting our hypothesis of privacy preference evolution.
- R2: Our participants thought it is important to be aware when privacy information is disclosed, and they felt that three different notification mechanisms offered by the system did promote their awareness of private information disclosure.
- R3: Our participants thought it is important to have control whenever their privacy information is disclosed, and multi-modal interactions on multiple devices allowed them to get convenient and timely access to privacy controls.
- R4: Our participants agreed it is important adjust user involvement in privacy decisions to reduce effort and intrusiveness, and they felt privacy rules were useful to support this. Both quantitative and qualitative analysis showed that in practice, people do not pre-specify privacy rules at the beginning of using a system, rather they tend to gain some experience with the system first, and then adjust their involvement in privacy management by creating rules over time.
- R5: Our participants thought it is important to make the behaviour of private information disclosure accountable, i.e., to maintain a record of what information people found out about them. Two participants used the privacy history to detecting unusual behaviour during the trial, and we speculate that it would become more useful in long-term usage of real-world applications.

7.3 Other Significant Results

7.3.1 Investigation of the Problem of Privacy

The concept of privacy is complex (i.e., difficult to comprehend) and dynamic (i.e., evolving over time), and the problem space of privacy is vast and spans across multiple

disciplines including psychology, sociology, economics, jurisprudence and computer science. This thesis contributed an in-depth investigation into the problem of privacy from historical, social, legal and technical perspectives, and presented an analysis of existing technical mechanisms for privacy. The thesis provided the following results based on this investigation:

- Examined technological impact on information privacy and identified key challenges that make privacy hard to achieve throughout each stage of the information lifecycle.
- Classified privacy threats in networked computing environments into two categories (i.e., malicious or covert privacy attacks, and accidental or negligent privacy intrusion) and identified the focus of the research as preventing accidental privacy violations in personal information sharing applications.
- Conducted a review of the literature, bringing together the historical, social, legal and technical perspectives in one place. Background on privacy from historical, social, and legal perspectives provided important context for understanding the technical mechanisms that are operating under existing social and legal frameworks.
- Investigated the technical mechanisms for achieving information privacy in networked computing environments, including early research in access control and encryption, anonymity and pseudonymity, recent development in privacy transparency and awareness, privacy enforcement, and work in system support for building privacy aware applications.

7.3.2 An Architecture for Adaptive Privacy Management

To satisfy the requirements for adaptive privacy management, this thesis presented the design of an overall architecture and platform that provide support for incorporating adaptive privacy management into distributed applications. In more detail, the results concerning the architecture and platform support are:

- Presented a set of key design decisions for adaptive privacy aware applications in our target domain. These design decisions were derived from the aforementioned set of requirements.

- Identified the limitations of existing design principles and frameworks to provide support for developing privacy aware applications. Motivated the need for platform support for developing adaptive privacy aware applications and argued that the platform should be flexible so that it can be configured or reconfigured to meet the requirements in different problem domains.
- Presented the design of an overall architecture and a middleware platform that supports for incorporating adaptive privacy management into distributed applications.
- Identified the need for transforming synchronous operation of information sharing into a number of asynchronous interactions, in order to support the coordination between users and the system during the process of adaptive privacy management.

7.3.3 An Instance of Middleware Platform

Followed the key design decisions and proposed architecture, the thesis presented the implementation of an instance of a middleware platform supporting rapid construction of privacy aware applications. The detailed results concerning implementing the middleware platform are:

- Identified a set of application programming interfaces that support construction of adaptive privacy aware applications, as well as a number of plug-in interfaces that customise the functionality offered by the platform and hence the behaviour of adaptive privacy management.
- Instantiated a number of plug-ins for the platform using different algorithms, to extend its basic functionality for privacy management, e.g., plug-ins for generating privacy notification using SMS and email, plug-ins for determining priority and resolving conflicts of privacy rules, and a plug-in for maintaining privacy related events in the underlying database.
- Developed a web portal and an SMS gateway as end user interfaces for an adaptive privacy manager.
- Integrated the adaptive privacy manager with a location service and developed a proof-of-concept application that enables end users to share GSM-based location information and to preserve location privacy using the adaptive approach.

7.4 Future Work

There are a number of issues related to this work that is worth further exploration in future research. We discuss some of the most significant elements in the following sections.

7.4.1 Improving the Location Sharing Application

One of the areas for future work is about extending and improving the current implementation of the privacy aware location sharing application. There are a few issues that arise from considering how this system can be extended to improve its usefulness:

- **Employ other location sensing technologies:** The current implementation of the application exploited GSM-based location sensing for providing location information with an accuracy that is slightly better than the size of a cell. To improve the location accuracy, other location sensing technologies can be employed, such as those that based on GPS, 802.11, Bluetooth, RFID, or a combination of the above. Providing location with increased accuracy may introduce further privacy concerns that would be interesting for further investigation.

- **Enable users to tag location with meaningful names:** Harrison and Dourish highlighted the critical distinction between "space" and "place" by arguing that the notion of place includes *"the dimensions of lived experience, interaction and use of a space by its inhabitants"* [Harrison96]. Rather than just providing users' addresses with geometric coordinates, the future system could enable users to tag physical location with semantically meaningful names, e.g., naming an address as someone's home or workplace. This would potentially provide more meaningful context for computer-mediated social interactions.

- **Disclose location actively or proactively:** The existing application only allows passive location disclosure, i.e., a user can release his location in response to an information requests, and this is based on the assumption that people want to know one anothers' location. However, there are occasions when people want others to know their location, and the future system could enable users to disclose their own location actively (e.g., publishing location to friends as a meeting

place) or proactively (e.g., automatically disclosing location when entering a certain area). This would potentially make the application more useful and introduce new privacy concerns at the same time.

- **Augment location into legacy services:** The usage of the location sharing application is not very high for average users, and this is partly because it is a stand-alone application. It has not been integrated into legacy services that people regularly use in established social practices. It would be interesting to augment location into legacy services (e.g., SMS, instant messaging, social networking, etc.), and we believe it would make location information more useful and interactions more convenient.

7.4.2 Using the Platform as a Testbed for Privacy Solutions

The design and implementation of our middleware platform is intended to support incorporating adaptive privacy management into multiple distributed applications, so that end users will manage multiple types of personal information with a greater degree of privacy. In the scope of the thesis, one specific instance of the platform was implemented, and the implemented system was incorporated into a single application for end users to manage one type of private information (i.e. location). To further explore privacy solutions for networked computing environments, the implemented platform can be employed as a testbed for developing various privacy aware applications involving more dimensions of private information, and those applications in turn can be used to evaluate the applicability and the adaptability of the platform. Possible areas of future research along this direction involve:

- **Develop applications concerning more dimensions of private information:** With the help of the middleware platform, we can start developing adaptive privacy aware applications that involve private information other than location, e.g., shared calendar information, 'status' in instant messengers, activities sensed by intelligent environments. In addition, we can put the platform in the public domain and encourage other application developers to employ it. By developing more applications involving more dimensions of private information, it would provide stronger evidence on the usefulness of the platform and hence of adaptive privacy management.

- **Customise and extend the platform:** The platform accomplishes flexibility and extensibility by defining a set of plug-in interfaces for developing custom algorithms and extensions to the platform. We could implement new behaviours by creating different plug-ins for the platform, and use the customised platform for different problem domains, allowing cross-validation of our results. For example, plug-ins for privacy preference learning and suggestion can be implemented to enable the system to automatically generate or suggest new privacy preferences over time. This specific platform may be employed for UbiComp applications such as intelligent environments where user involvement requires being kept at the minimum level.

- **Evaluate the applicability and adaptability of the platform:** As identified by Edwards et al. [Edwards03], it is a challenging problem to evaluate middleware platform that supports the development of interactive applications. By developing more applications using the platform and creating various plug-ins to customise it, we would gain better insight on the applicability and adaptability of the platform. We hope that our own experience of using the platform and feedback from other application developers would help us to evaluate its design and implementation, as well as refine and improve it further.

7.4.3 Extending Adaptive Privacy Management

In this thesis, we have concentrated on designing and implementing adaptive privacy management mechanisms in the scope of individual-to-individual interactions mediated by distributed applications, as opposed to interactions between individuals and organisations. With the flourishing of networked services and the advance of UbiComp systems and applications, people are increasingly concerned that their private information is being collected and exploited while they are interacting with these services and applications. This is an intricate social-technical problem that can only be achieved through a combination of technologies, legislation, social norms, and market forces. An interesting future research area would be to extend adaptive privacy approach to individual-to-organisation settings, more particularly:

- **Establish new threat model for privacy:** For the interactions between individuals and organisations, the threats to personal information privacy would be dif-

ferent from the individual-to-individual interactions. Therefore, it is necessary to establish a new threat model for privacy before we could further investigate the applicability of adaptive privacy management.

- **Investigate applicability of adaptive privacy management principles:** It would be necessary to assess how well the five core principles of the adaptive approach apply in individual-to-organisation settings. This could be done theoretically initially, by analysing a few case studies of networked services and UbiComp applications. It may require extending existing core principles or incorporating new ones.

- **Incorporate other design strategies:** The design and implementation of our system employed three design strategies, i.e., control at collection, awareness and accountability. For the individual-to-organisation settings, other design strategies (e.g., anonymity or pseudonymity, control at the point of use, etc.) may be required to be incorporated to cope with new privacy threats in the problem domain, e.g., secondary usage of private information, data mining, etc.

7.5 Concluding Remarks

Privacy has become a growing concern in networked computing environments and future UbiComp systems. People selectively share private information using networked applications to improve inter-personal communication, while at the same time they want to remain in control of their privacy. Privacy management in networked environments is not about hiding as much private information as possible, but should be a dialectic and dynamic process to adjust the level of openness for different circumstances.

Following the above insight on privacy, this thesis has investigated a technical approach, i.e., adaptive privacy management, to support the dynamic process that a user and/or a system continuously adjusts the system behaviour of disclosing personal information. The thesis demonstrated the feasibility of adaptive privacy management and evaluated with end users that it would lead to better privacy. The author hopes that the principles advocated throughout the thesis will, in long term, contribute to the development of technical solutions that allow people to achieve better privacy in networked computing environments.

References

[Ackerman99] Ackerman, M. S., Cranor, L. F. and Reagle, J. Privacy in e-commerce: examining user scenarios and privacy preferences. In *EC '99: Proceedings of the 1st ACM conference on Electronic commerce*, pp. 1–8. ACM Press, New York, NY, USA. 1999.

[Ackerman04] Ackerman, M. S. Privacy in pervasive environments: next generation labeling protocols. *Personal Ubiquitous Comput.*, 8(6):pp. 430–439. 2004.

[Acquisti03] Acquisti, A., Dingledine, R. and Syverson, P. On the Economics of Anonymity. In Wright, R. N. (ed.), *Proceedings of Financial Cryptography (FC '03)*. Springer-Verlag, LNCS 2742. 2003.

[Acquisti04] Acquisti, A. Security of Personal Information and Privacy: Technological Solutions and Economic Incentives. In Camp, J. and Lewis, R. (eds.), *The Economics of Information Security*. Kluwer. 2004.

[Acquisti05a] Acquisti, A. and Grossklags, J. Privacy and Rationality in Individual Decision Making. *IEEE Security and Privacy*, 3(1):pp. 26–33. 2005.

[Acquisti05b] Acquisti, A. and Grossklags, J. Uncertainty, Ambiguity and Privacy. In *WEIS'05: Proceedings of the Fourth Workshop on the Economics of Information Security*. 2005. Available at http://infosecon.net/workshop/pdf/64.pdf.

[Adams01a] Adams, A. *Users' Perceptions of Privacy In Multimedia Communications*. Ph.D. thesis, Departments of Psychology and Computer Science, University College London. 2001.

[Adams01b] Adams, A. and Sasse, M. A. Privacy in Multimedia Communications: Protecting Users, Not Just Data. In *Joint proceedings of the HCI2001 and ICM2001*, pp. 49–64. Springer. 2001.

[Allen99] Allen, J., Guinn, C. and Horvtz, E. Mixed-Initiative Interaction. *IEEE Intelligent Systems*, 14(5):pp. 14–23. 1999.

[Altman77] Altman, I. Privacy Regulation: Culturally Universal or Culturally Specific? *Journal of Social Issues*, 33(3):pp. 66–84. 1977.

[Anderson01] Anderson, R. *Security Engineering: A Guide to Building Dependable Distributed Systems*. John Wiley & Sons Inc. 2001.

[Anderson03a] Anderson, R. Cryptography and competition policy: issues with 'trusted computing'. In *PODC '03: Proceedings of the twenty-second annual symposium on Principles of distributed computing*, pp. 3–10. ACM Press, New York, NY, USA. 2003.

[Anderson03b] Anderson, R. Trusted Computing Frequently Asked Questions. http://www.cl.cam.ac.uk/ rja14/tcpa-faq.html. 2003.

[Anderson04] Anderson, R. Personal Information, Privacy and Ubicomp. In *2nd UK-UbiNet Workshop: Security, trust, privacy and theory for ubiquitous computing*. University of Cambridge. 2004.

[Anderson06] Anderson, R., Bond, M., Clulow, J. and Skorobogatov, S. Cryptographic Processors-a survey. *Proceedings of the IEEE*, 94(2):pp. 357–369. 2006.

[Anonymizer Inc.07] Anonymizer Inc. Anonymizer.com, Anonymous proxy servers [online]. 2007. Available from: `http://www.anonymizer.com` [cited 29 June 2007].

[Anonymous03] Anonymous. London Congestion Charge CCTV privacy concerns [online]. 2003. Available from: `http://www.spy.org.uk/cgi-bin/cclondon.pl` [cited 29 June 2007].

[Answers Corp.07] Answers Corp. "encryption." Computer Desktop Encyclopedia. [online]. 2007. Available from: `http://www.answers.com/topic/encryption` [cited 29 June 2007].

[Aoki03] Aoki, K. and Downes, E. J. An analysis of young people's use of and attitudes toward cell phones. *Telemat. Inf.*, 20(4):pp. 349–364. 2003.

[Aoki05] Aoki, P. M. and Woodruff, A. Making space for stories: ambiguity in the design of personal communication systems. In *CHI '05: Proceedings of the SIGCHI*

conference on Human factors in computing systems, pp. 181–190. ACM Press, New York, NY, USA. 2005.

[Arminen03] Arminen, I. Location: a socially dynamic property — a study of location telling in mobile phone calls. In Haddon, L., Mante-Meijer, E., Sapio, K. H. K. B., Fortunati, L. and Kant, A. (eds.), *Proceedings of The Good, the Bad and the Irrelevant: The User and the Future of Information and Communication Technologies*. 2003. Available from: http://goodbad.uiah.fi/files/public/P025/P025.doc .

[Ashley02] Ashley, P., Powers, C. and Schunter, M. From privacy promises to privacy management: a new approach for enforcing privacy throughout an enterprise. In *NSPW '02: Proceedings of the 2002 workshop on New security paradigms*, pp. 43–50. ACM Press, New York, NY, USA. 2002.

[Ashley03] Ashley, P., Hada, S., Karjoth, G., Powers, C. and Schunter, M. Enterprise Privacy Authorization Language (EPAL 1.2) [online]. 2003. Available from: http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/ [cited 29 June 2007].

[AT&T Labs01] AT&T Labs. AT&T's Sentient Computing [online]. 2001. Available from: http://www.cl.cam.ac.uk/Research/DTG/attarchive/spirit/ [cited 29 June 2007].

[Bacard] Bacard, A. Anonymous Remailer FAQ. http://www.andrebacard.com/remail.html.

[Bacon95] Bacon, J., Bates, J., Hayton, R. and Moody, K. Using Events to Build Distributed Applications. In *SDNE '95: Proceedings of the 2nd International Workshop on Services in Distributed and Networked Environments*, p. 148. IEEE Computer Society, Washington, DC, USA. 1995.

[Bacon00] Bacon, J., Moody, K., Bates, J., Hayton, R., Ma, C., McNeil, A., Seidel, O. and Spiteri, M. Generic Support for Distributed Applications. *Computer*, 33(3):pp. 68–76. 2000.

[Bardram04] Bardram, J. E. and Hansen, T. R. The AWARE architecture: supporting context-mediated social awareness in mobile cooperation. In *CSCW '04: Pro-*

ceedings of the 2004 ACM conference on Computer supported cooperative work, pp. 192–201. ACM Press, New York, NY, USA. 2004.

[Barkuus03] Barkuus, L. and Dey, A. Location-Based Services for Mobile Telephony: a study of users' privacy concerns. In *Interact'03: Ninth IFIP TC13 International Conference on Human-Computer Interaction*, pp. 709–712. IOS Press. 2003.

[Bass01] Bass, L., John, B. E. and Kates, J. Achieving Usability Through Software Architecture. Tech. rep., Carnegie Mellon University, Software Engineering Institute. 2001. Available from: `http://www.sei.cmu.edu/publications/documents/01.reports/01tr005.html` .

[Baus05] Baus, J., Cheverst, K. and Kray, C. A Survey of Mobile Guides. In *A Survey of Map-based Mobile Guides in: Map-based mobile services - Theories, Methods and Implementations. Chapter 13.* Springer-Verlag. 2005.

[BBC04] BBC. Google's Gmail sparks privacy row [online]. 2004. Available from: `http://news.bbc.co.uk/1/hi/business/3602745.stm` [cited 29 June 2007].

[BBC06] BBC. Google defies US over search data [online]. 2006. Available from: `http://news.bbc.co.uk/1/hi/technology/4630694.stm` [cited 29 June 2007].

[Beck04] Beck, K. and Andres, C. *Extreme Programming Explained: Embrace Change (2nd Edition).* Addison-Wesley Professional. 2004.

[Beckwith03] Beckwith, R. Designing for Ubiquity: The Perception of Privacy. *IEEE Pervasive Computing*, 2(2):pp. 40–46. 2003.

[Bell06] Bell, D. G. The Age of the Thumb: An Ethnographic Account of Cell Phones in Asia. In *UCSC Digital Arts and New Media MFA Program (DANM) Colloquium Series*. University of California Santa Cruz. 2006.

[Bellotti93] Bellotti, V. and Sellen, A. Design for Privacy in Ubiquitous Computing Environments. In *Proceedings of the Third European Conference on Computer Supported Cooperative Work (ECSCW'93)*, pp. 77–92. Kluwer. 1993.

[Bellotti97] Bellotti, V. *Design for privacy in multimedia computing and communications environments*, pp. 63–98. MIT Press, Cambridge, MA, USA. 1997.

[Beresford03] Beresford, A. R. and Stajano, F. Location Privacy in Pervasive Computing. *IEEE Pervasive Computing*, 2(1):pp. 46–55. 2003.

[Beresford05] Beresford, A. R. Location privacy in ubiquitous computing. Tech. Rep. UCAM-CL-TR-612, University of Cambridge, Computer Laboratory. 2005. Available from: http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-612.pdf .

[Birrell84] Birrell, A. D. and Nelson, B. J. Implementing remote procedure calls. *ACM Trans. Comput. Syst.*, 2(1):pp. 39–59. 1984.

[Booth04] Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C. and Orchard, D. Web Services Architecture W3C Working Group Note 11 February 2004 [online]. 2004. Available from: http://www.w3.org/TR/ws-arch/ [cited 29 June 2007].

[Boyle05] Boyle, M. and Greenberg, S. The language of privacy: Learning from video media space analysis and design. *ACM Trans. Comput.-Hum. Interact.*, 12(2):pp. 328–370. 2005.

[CBS News04] CBS News. GPS Keeping Tabs On Car Rentals. http://www.cbsnews.com/stories/2004/03/06/eveningnews/main604461.shtml. 2004.

[CDT95] CDT. European Union's Directive 95/46/EC [online]. 1995. Available from: http://www.cdt.org/privacy/eudirective/EU_Directive_.html [cited 29 June 2007].

[Chatfield04] Chatfield, C. and Häkkilä, J. Designing intelligent environments - user perceptions on information sharing. In *In Proceedings of 6th Asia-Pacific Conference on Computer-Human Interaction*. 2004.

[Chatfield05] Chatfield, C., Carmichael, D., Hexel, R., Kay, J. and Kummerfeld, B. Personalisation in intelligent environments: managing the information flow. In *OZCHI '05: Proceedings of the 19th conference of the computer-human interaction special interest group (CHISIG) of Australia on Computer-human interaction*, pp. 1–10. Computer-Human Interaction Special Interest Group (CHISIG) of Australia, Narrabundah, Australia, Australia. 2005.

[Chaum81] Chaum, D. L. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):pp. 84–90. 1981.

[Cheverst01] Cheverst, K. and Smith, G. Exploring the notion of information push and pull with respect to the user intention and disruption. 2001. Available from: citeseer.ist.psu.edu/cheverst01exploring.html .

[Chinnici06] Chinnici, R., Moreau, J.-J., Ryman, A. and Weerawarana, S. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language [online]. 2006. Available from: http://www.w3.org/TR/wsdl20/ [cited 29 June 2007].

[Chipchase07] Chipchase, J. Shared Phone Use [online]. 2007. Available from: http://www.janchipchase.com/sharedphoneuse [cited 29 June 2007].

[CMU SEI00] CMU SEI. Three Tier Software Architectures [online]. 2000. Available from: http://www.sei.cmu.edu/str/descriptions/threetier.html [cited 29 June 2007].

[Cohen03] Cohen, J. E. DRM and privacy. *Commun. ACM*, 46(4):pp. 46–49. 2003.

[Communities02] Communities, E. Directive 2002/58/EC (Directive on priacy and electronic communications. http://europa.eu.int/eur-lex/pri/en/oj/dat/2002/l_201/l_20120020731en00370047.pdf. 2002.

[Consolvo05] Consolvo, S., Smith, I. E., Matthews, T., LaMarca, A., Tabert, J. and Powledge, P. Location disclosure to social relations: why, when, & what people want to share. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 81–90. ACM Press, New York, NY, USA. 2005.

[Coulouris01] Coulouris, G., Dollimore, J. and Kindberg, T. *Distributed systems (3rd ed.): concepts and design*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. 2001.

[Covington01] Covington, M. J., Long, W., Srinivasan, S., Dev, A. K., Ahamad, M. and Abowd, G. D. Securing context-aware applications using environment roles. In *SACMAT '01: Proceedings of the sixth ACM symposium on Access control models and technologies*, pp. 10–20. ACM Press, New York, NY, USA. 2001.

[Cranor06] Cranor, L. and Wenning, R. Platform for Privacy Preferences Project (P3P) — finalised 13 November 2006 [online]. 2006. Available from: http://www.w3.org/P3P/ [cited 29 June 2007].

[Cuellar02] Cuellar, J. R. Location Information Privacy. In *In Sarikaya, Behcet (Ed.) Geographic Location in the Internet*. 2002.

[Cuellar04] Cuellar, J., Morris, J., Mulligan, D., Peterson, J. and Polk, J. *RFC3693: Geopriv Requirements*. IETF's Geopriv Working Group. 2004. Available from: `http://www.ietf.org/rfc/rfc3693.txt` .

[Dai07] Dai, W. PipeNet v1.1 White Paper [online]. 2007. Available from: `http://www.eskimo.com/~weidai/pipenet.txt` [cited 29 June 2007].

[Danezis05] Danezis, G., Lewis, S. and Anderson, R. How much is location privacy worth? In *WEIS'05: Proceedings of the Fourth Workshop on the Economics of Information Security*. 2005. Available at http://infosecon.net/workshop/pdf/location-privacy.pdf.

[Davies98] Davies, N., Friday, A., Wade, S. P. and Blair, G. S. An asynchronous distributed systems platform for heterogeneous environments. In *EW 8: Proceedings of the 8th ACM SIGOPS European workshop on Support for composing distributed applications*, pp. 66–73. ACM Press, New York, NY, USA. 1998.

[Davies02] Davies, N. and Gellersen, H.-W. Beyond Prototypes: Challenges in Deploying Ubiquitous Systems. *IEEE Pervasive Computing*, 1(1):pp. 26–35. 2002.

[Dhamija06] Dhamija, R., Tygar, J. D. and Hearst, M. Why phishing works. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 581–590. ACM Press, New York, NY, USA. 2006.

[Diffie76] Diffie, W. and Hellman, M. E. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT-22(6):pp. 644–654. 1976.

[Dix98] Dix, A., Finley, J., Abowd, G. and Beale, R. *Human-computer interaction (2nd ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA. 1998.

[Dourish92] Dourish, P. and Bellotti, V. Awareness and coordination in shared workspaces. In *CSCW '92: Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, pp. 107–114. ACM Press, New York, NY, USA. 1992.

[Dragovic05a] Dragovic, B. and Crowcroft, J. Information exposure control through data manipulation for ubiquitous computing. In *NSPW '04: Proceedings of the*

2004 workshop on New security paradigms, pp. 57–64. ACM Press, New York, NY, USA. 2005.

[Dragovic05b] Dragovic, B. and Policroniades, C. Information SeeSaw: Availability vs. Security Managment in the UbiComp World. In *SDM'05: Proceedings of Secure Data Management: Second VLDB Workshop*, pp. 200–216. 2005.

[Dunlop03] Dunlop, N., Indulska, J. and Raymond, K. Methods for Conflict Resolution in Policy-Based Management Systems. *edoc*, 00:p. 98. 2003.

[Edwards03] Edwards, W. K., Bellotti, V., Dey, A. K. and Newman, M. W. The challenges of user-centered design and evaluation for infrastructure. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 297–304. ACM Press, New York, NY, USA. 2003.

[Egenhofer99] Egenhofer, M. J. and Rodríguez, M. A. Relation algebras over containers and surfaces: An ontological study of a room space. *Spatial Cognition and Computation*, 1(2):pp. 155–180. 1999.

[Eldin04] Eldin, A. A., van den Berg, J. and Wagenaar, R. A fuzzy reasoning scheme for context sharing decision making. In *ICEC '04: Proceedings of the 6th international conference on Electronic commerce*, pp. 371–375. ACM Press, New York, NY, USA. 2004.

[Escofet03] Escofet, G. What the operators are doing. *Mobile Location Analyst*, pp. 14–18. 2003.

[Esposito02] Esposito, D. .NET Remoting—Design and Develop Seamless Distributed Applications for the Common Language Runtime. http://msdn.microsoft.com/msdnmag/issues/02/10/NETRemoting/default.aspx. 2002.

[Felten06] Felten, J. A. H. E. W. Digital Rights Management, Spyware, and Security. *IEEE Security and Privacy*, 4(1):pp. 18–23. 2006.

[Ferraiolo92] Ferraiolo, D. and Kuhn, R. Role-Based Access Controls. In *15th National Computer Security Conference*. 1992.

[Ferraiolo01] Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R. and Chandramouli, R. Proposed NIST standard for role-based access control. *ACM Trans. Inf. Syst. Secur.*, 4(3):pp. 224–274. 2001.

[FTC98] FTC. Privacy Online: A Report to Congress [online]. 1998. Available from: http://www.ftc.gov/reports/privacy3/ [cited 29 June 2007].

[Gamma95] Gamma, E., Helm, R., Johnson, R. and Vlissides, J. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. 1995.

[Gelernter85] Gelernter, D., Carriero, N., Chandran, S. and Chang, S. Parallel Programming in Linda. In *ICPP*, pp. 255–263. 1985.

[Goldberg97] Goldberg, I., Wagner, D. and Brewer, E. Privacy-enhancing technologies for the Internet. In *COMPCON '97: Proceedings of the 42nd IEEE International Computer Conference*, p. 103. IEEE Computer Society, Washington, DC, USA. 1997.

[Goldberg02] Goldberg, I. Privacy-Enhancing Technologies for the Internet, II: Five Years Later. In *Privacy Enhancing Technologies*, pp. 1–12. 2002.

[Goldschlag99] Goldschlag, D., Reed, M. and Syverson, P. Onion routing. *Communication of the ACM*, 42(2):pp. 39–41. 1999.

[Google Inc.06] Google Inc. Google Maps [online]. 2006. Available from: http://maps.google.com/ [cited 29 June 2007].

[Google Inc.07a] Google Inc. DodgeBall [online]. 2007. Available from: http://www.dodgeball.com/ [cited 29 June 2007].

[Google Inc.07b] Google Inc. Google Groups [online]. 2007. Available from: http://groups.google.com/ [cited 29 June 2007].

[Graham72] Graham, G. S. and Denning, P. Protection – Principles and Practice. In *AFIPS Spring Join Comp. Conf.*, pp. 417–429. 1972.

[Greenfield02] Greenfield, A. *Everyware: The Dawning Age of Ubiquitous Computing*. Peachpit Press Publications,U.S., 1st edn. 2002.

[Grinter01] Grinter, R. E. and Eldridge, M. y do tngrs luv 2 txt msg? In *ECSCW*, pp. 219–238. 2001.

[Group] Group, A.-P. W. [online].

[Grudin01] Grudin, J. Desituating Action:Digital represenation of Context. *Human-Computer Interaction*, 16(2/4):pp. 269–286. 2001.

[Grudin03] Grudin, J. and Horvitz, E. Presenting choices in context: approaches to information sharing. In *Workshop on Ubicomp communities: Privacy as Boundary Negotiation*. 2003. Available from: http://guir.berkeley.edu/pubs/ubicomp2003/privacyworkshop/papers.htm .

[Gudgin03] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J.-J., Nielsen, H. F., Karmarkar, A. and Lafon, Y. SOAP Version 1.2 Part 1: Messaging Framework [online]. 2003. Available from: http://www.w3.org/TR/soap12-part1/ [cited 29 June 2007].

[Gurteen02] Gurteen, D. Knowledge, Awareness and Understanding. 2002. Available from: http://www.gurteen.com/gurteen/gurteen.nsf/0/8A9D48887C22104280256BD2002F66DF/ .

[Hadzilacos94] Hadzilacos, V. and Toueg, S. A Modular Approach to Fault-Tolerant Broadcasts and Related Problems. Tech. Rep. TR94-1425, Cornell University, Dept. of Computer Science. 1994. Available from: http://citeseer.ist.psu.edu/hadzilacos94modular.html .

[Hallam-Baker06] Hallam-Baker, P. Achieving Email Security Usability. 2006. 5th Annual PKI R&D Workshop. Available from: http://middleware.internet2.edu/pki06/proceedings/index.html .

[Harper96] Harper, R. H. R. Why people do and don't wear active badges: a case study. *Comput. Supported Coop. Work*, 4(4):pp. 297–318. 1996.

[Harris Interactive07] Harris Interactive. Harris Poll Online [online]. 2007. Available from: http://www.harrispollonline.com/ [cited 29 June 2007].

[Harris98] Harris, L., Associates and Westin, A. F. E-Commerce and Privacy: What Net Users Want. 1998.

[Harrison75] Harrison, M. A., Ruzzo, W. L. and Ullman, J. D. On protection in operating systems. In *SOSP '75: Proceedings of the fifth ACM symposium on Operating systems principles*, pp. 14–24. ACM Press, New York, NY, USA. 1975.

[Harrison88] Harrison, S., Minneman, S. and Stults, B. The Media Space - experience with video support of design activity. In *International Workshop on Engineering Design and Manufacturing Management*, pp. 114–126. 1988.

[Harrison96] Harrison, S. and Dourish, P. Re-place-ing space: the roles of place and space in collaborative systems. In *CSCW '96: Proceedings of the 1996 ACM conference on Computer supported cooperative work*, pp. 67–76. ACM Press, New York, NY, USA. 1996.

[Hazas04] Hazas, M., Scott, J. and Krumm, J. Location-Aware Computing Comes of Age. *Computer*, 37(2):pp. 95–97. 2004.

[Heath91] Heath, C. and Luff, P. Disembodied Conduct: Communication through Video in a Multi-Media Office Environment. In *CHI'91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 99–103. ACM Press, New York, NY, USA. 1991.

[Helmers97] Helmers, S. A Brief History of anon.penet.fi - The Legendary Anonymous Remailer. *Computer-Mediated Communication Magazine*, 4(9). 1997. Available at http://www.december.com/cmc/mag/1997/sep/helmers.html.

[Hendricks01] Hendricks, E. Wireless Location Technology: The Ultimate Challenge to Privacy. http://www.paris-conference-2001.org/eng/contribution/hendricks_contrib.pdf. 2001.

[Henricksen05] Henricksen, K., Wishart, R., McFadden, T. and Indulska, J. Extending Context Models for Privacy in Pervasive Computing Environments. In *PerCom Workshops*, pp. 20–24. 2005.

[Hindus01] Hindus, D., Mainwaring, S. D., Leduc, N., Hagström, A. E. and Bayley, O. Casablanca: designing social communication devices for the home. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 325–332. ACM Press, New York, NY, USA. 2001.

[Hochheiser02] Hochheiser, H. The platform for privacy preference as a social protocol: An examination within the U.S. policy context. *ACM Transactions on Internet Technology (TOIT)*, 2(4):pp. 276–306. 2002.

[Hong04a] Hong, J. I. and Landay, J. A. An architecture for privacy-sensitive ubiquitous computing. In *MobiSys '04: Proceedings of the 2nd international conference*

on Mobile systems, applications, and services, pp. 177–189. ACM Press, New York, NY, USA. 2004.

[Hong04b] Hong, J. I., Ng, J. D., Lederer, S. and Landay, J. A. Privacy risk models for designing privacy-sensitive ubiquitous computing systems. In *DIS '04: Proceedings of the 2004 conference on Designing interactive systems*, pp. 91–100. ACM Press, New York, NY, USA. 2004.

[Hong05] Hong, J. *An Architecture for Privacy-Sensitive Ubiquitous Computing*. Ph.D. thesis, Departments of Computer Science, University of California, Berbeley. 2005.

[HP Corp.01] HP Corp. HP's Cooltown [online]. 2001. Available from: http://www.hpl.hp.com/archive/cooltown/ [cited 29 June 2007 (content under review)].

[Huberman05] Huberman, B. A., Adar, E. and Fine, L. R. Valuating Privacy. *IEEE Security and Privacy*, 3(5):pp. 22–25. 2005.

[Hull03] Hull, R., Kumar, B., Lieuwen, D. F., Patel-Schneider, P. F., Sahuguet, A., Varadarajan, S. and Vyas, A. A policy-based system for personalized and privacy-conscious user data sharing. Tech. rep., Bell Labs. 2003. Available from: http://db.bell-labs.com/project/e-services-customization/personal-data-sharing-2003-TM.pdf .

[Hull04] Hull, R., Kumar, B., Lieuwen, D. F., Patel-Schneider, P. F., Sahuguet, A., Varadarajan, S. and Vyas, A. Enabling Context-Aware and Privacy-Conscious User Data Sharing. In *Mobile Data Management*, pp. 187–198. 2004.

[Iachello05] Iachello, G. and Abowd, G. D. Privacy and proportionality: adapting legal evaluation techniques to inform design in ubiquitous computing. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 91–100. ACM Press, New York, NY, USA. 2005.

[Iannella01] Iannella, R. Digital Rights Management (DRM) Architectures. *D-Lib Magazine*, 7(6). 2001. Available from: http://www.dlib.org/dlib/june01/iannella/06iannella.html .

[IBM Corp.06] IBM Corp. IBM's Enterprise Privacy Technologies [online]. 2006. Available from: http://www.zurich.ibm.com/csc/cryptography/epa.html [cited 29 June 2007].

[Internet Archive07] Internet Archive. WayBack Machine [online]. 2007. Available from: `http://www.archive.org/` [cited 29 June 2007].

[Jacobson97] Jacobson, I., Griss, M. and Jonsson, P. *Software reuse: architecture, process and organization for business success*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA. 1997.

[James05] James, L. *Phishing Exposed*. Syngress Publishing. 2005.

[Jensen04] Jensen, C. and Potts, C. Privacy policies as decision-making tools: an evaluation of online privacy notices. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 471–478. ACM Press, New York, NY, USA. 2004.

[Jensen05] Jensen, C., Tullio, J., Potts, C. and Mynatt, E. D. STRAP: A Structured Analysis Framework for Privacy. Tech. Rep. GIT-GVU-05-02, Graphics, Visualization, and Usability Center (GVU Center), Georgia Institute of Technology. 2005.

[Jiang02a] Jiang, X., Hong, J. I. and Landay, J. A. Approximate Information Flows: Socially-Based Modeling of Privacy in Ubiquitous Computing. In *Ubicomp*, pp. 176–193. 2002.

[Jiang02b] Jiang, X. and Landay, J. A. Modeling Privacy Control in Context-Aware Systems. *IEEE Pervasive Computing*, 1(3):pp. 59–63. 2002.

[Johanson02] Johanson, B. and Fox, A. The Event Heap: A Coordination Infrastructure for Interactive Workspaces. In *WMCSA '02: Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications*, p. 83. IEEE Computer Society, Washington, DC, USA. 2002.

[Joinson06] Joinson, A. N., Paine, C., Buchanan, T. and Reips, U.-D. Watching me, watching you: privacy attitudes and reactions to identity card implementation scenarios in the United Kingdom. *Journal of Information Science*, 32(4):pp. 334–343. 2006.

[Jones03] Jones, K. S. Privacy: what's different now? *Interdisciplinary Science Reviews*, 28(4). 2003.

[Kaasinen03] Kaasinen, E. User needs for location-aware mobile services. *Personal Ubiquitous Computing*, 7(1):pp. 70–79. 2003.

[Kapadia07] Kapadia, A., Henderson, T., Fielding, J. and Kotz, D. Virtual Walls: Protecting Digital Privacy in Pervasive Environments. In *Proceedings of the Fifth International Conference on Pervasive Computing (Pervasive)*, Lecture Notes in Computer Science. Springer-Verlag. 2007. Available from: http://www.cs.dartmouth.edu/~dfk/papers/kapadia:walls.pdf .

[Karjoth02] Karjoth, G. and Schunter, M. A Privacy Policy Model for Enterprises. In *CSFW '02: Proceedings of the 15th IEEE Computer Security Foundations Workshop (CSFW'02)*, p. 271. IEEE Computer Society, Washington, DC, USA. 2002.

[Kim00] Kim, A. J. *Community Building on the Web: Secret Strategies for Successful Online Communities*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. 2000.

[Ku04] Ku, W. and Chi, C.-H. Survey on the Technological Aspects of Digital Rights Management. In *Proceeding of the 7th Information Security Conference*, pp. 391–403. 2004.

[Kupper05] Kupper, A. *Location-based Services: Fundamentals and Operation*. John Wiley & Sons. 2005.

[Lampson74] Lampson, B. W. Protection. *SIGOPS Oper. Syst. Rev.*, 8(1):pp. 18–24. 1974.

[Langheinrich01] Langheinrich, M. Privacy by Design - Principles of Privacy-Aware Ubiquitous Systems. In *UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing*, pp. 273–291. Springer-Verlag, London, UK. 2001.

[Langheinrich02a] Langheinrich, M. A P3P Preference Exchange Language 1.0 (APPEL1.0) [online]. 2002. Available from: http://www.w3.org/TR/P3P-preferences/ [cited 29 June 2007].

[Langheinrich02b] Langheinrich, M. A Privacy Awareness System for Ubiquitous Computing Environments. In *UbiComp '02: Proceedings of the 4th interna-*

tional conference on Ubiquitous Computing, pp. 237–245. Springer-Verlag, London, UK. 2002.

[Laurant03] Laurant, C. (ed.). *Privacy and Human Rights 2003*. EPIC and Privacy International, London, UK. 2003. Available at http://www.privacyinternational.org/survey/phr2003/.

[Lederer03a] Lederer, S., Mankoff, J. and Dey, A. K. Who wants to know what when? privacy preference determinants in ubiquitous computing. In *CHI '03: CHI '03 extended abstracts on Human factors in computing systems*, pp. 724–725. ACM Press, New York, NY, USA. 2003.

[Lederer03b] Lederer, S., Mankoff, J., Dey, A. K. and Beckmann, C. Managing Personal Information Disclosure in Ubiquitous Computing Environments. Tech. Rep. UCB/CSD-03-1257, EECS Department, University of California, Berkeley. 2003.

[Lederer04] Lederer, S., Hong, I., Dey, K. and Landay, A. Personal privacy through understanding and action: five pitfalls for designers. *Personal Ubiquitous Comput.*, 8(6):pp. 440–454. 2004.

[Lemos01] Lemos, R. Rental-car firm exceeding the privacy limit? http://news.com.com/2100-1040-268747.html. 2001.

[Lessig98] Lessig, L. The Architecture of Privacy. In *CTaiwan NET'98*. 1998.

[Lessig99] Lessig, L. *Code and Other Laws of Cyberspace*. Basic Books, New York, NY. 1999.

[Lester05] Lester, J., Choudhury, T., Borriello, G., Consolvo, S., Landay, J., Everitt, K. and Smith, I. Sensing and Modeling Activities to Support Physical Fitness. In *In the Workshop Proceedings of Ubicomp (Workshop: Monitoring, Measuring and Motivating Exercise: Ubiquitous Computing to Support Fitness)*. Tokyo, Japan. 2005. Available from: `http://seattleweb.intel-research.net/projects/ubifit/papers/w10-p11-rev.pdf` .

[Lucent Technologies98] Lucent Technologies. The Lucent Personalized Web Assistant [online]. 1998. Available from: `http://www.bell-labs.com/project/lpwa/` [cited 29 June 2007].

[Mabley00] Mabley, K. Privacy vs. Personalization. http://www.egov.vic.gov.au/pdfs/wp-2000-privacy3.pdf. 2000.

[Mackay91a] Mackay, W. E. Ethical issues in the use of video: is it time to establish guidelines? In *CHI'91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 403–405. ACM Press, New York, NY, USA. 1991.

[Mackay91b] Mackay, W. E. Triggers and barriers to customizing software. In *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 153–160. ACM Press, New York, NY, USA. 1991.

[Mazières98] Mazières, D. and Kaashoek, M. F. The design, implementation and operation of an email pseudonym server. In *CCS '98: Proceedings of the 5th ACM conference on Computer and communications security*, pp. 27–36. ACM Press, New York, NY, USA. 1998.

[MBG06] MBG. Industry Code of Practice for the use of mobile phone technology to provide passive location services in the UK [online]. 2006. Available from: `http://www.mobilebroadbandgroup.com/` [cited 29 June 2007].

[Meyer88] Meyer, B. *Object-Oriented Software Construction*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA. 1988.

[Michaelis06] Michaelis, M. *Essential C# 2.0 (Microsoft .Net Development Series)*. Addison-Wesley Professional. 2006.

[Microsoft Corp.01] Microsoft Corp. Microsoft's EasyLiving [online]. 2001. Available from: `http://research.microsoft.com/easyliving/` [cited 29 June 2007].

[MIT04] MIT. MIT's Oxygen [online]. 2004. Available from: `http://oxygen.lcs.mit.edu/` [cited 29 June 2007].

[MobileLocate Ltd.07] MobileLocate Ltd. MobileLocate.co.uk [online]. 2007. Available from: `http://www.mobilelocate.co.uk/` .

[Myles03] Myles, G., Friday, A. and Davies, N. Preserving Privacy in Environments with Location-Based Applications. *IEEE Pervasive Computing*, 2(1):pp. 56–64. 2003.

[Nardi00] Nardi, B. A., Whittaker, S. and Bradner, E. Interaction and outeraction: instant messaging in action. In *CSCW '00: Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pp. 79–88. ACM Press, New York, NY, USA. 2000.

[Nardi04] Nardi, B. A., Schiano, D. J., Gumbrecht, M. and Swartz, L. Why we blog. *Commun. ACM*, 47(12):pp. 41–46. 2004.

[Netcetera Limited07] Netcetera Limited. World-Tracker.com [online]. 2007. Available from: http://www.world-tracker.com/ [cited 29 June 2007].

[Nielsen94] Nielsen, J. Enhancing the explanatory power of usability heuristics. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 152–158. ACM Press, New York, NY, USA. 1994.

[NY Lawyer07] NY Lawyer. Lewan v. Sharman, U.S.Dist. Ct., N.D. Ill 06-cv-6736. 2007. Available from: http://yro.slashdot.org/article.pl?sid=06/12/07/1756200 [cited 22 June 2007].

[Oates06] Oates, J. Parliament committee hears DRM rights and wrongs [online]. 2006. Available from: http://www.theregister.co.uk/2006/02/02/apig_hears_evidence/ [cited 29 June 2007].

[Odlyzko03] Odlyzko, A. Privacy, economics, and price discrimination on the Internet. In *ICEC '03: Proceedings of the 5th international conference on Electronic commerce*, pp. 355–366. ACM Press, New York, NY, USA. 2003.

[OECD80] OECD. *OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data*. Organization for Economic Cooperation and Development. 1980. Available from: Availableathttp://www.oecd.org [cited 29 June 2007].

[Olson05] Olson, J. S., Grudin, J. and Horvitz, E. A study of preferences for sharing and privacy. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pp. 1985–1988. ACM Press, New York, NY, USA. 2005.

[OMG04a] OMG. Common Object Request Broker Architecture: Core Specification [online]. 2004. Available from: http://www.omg.org/docs/formal/04-03-12.pdf [cited 29 June 2007].

[OMG04b] OMG. Event Service Specification [online]. 2004. Available from: http://www.omg.org/docs/formal/04-10-02.pdf [cited 29 June 2007].

[OMG04c] OMG. Notification Service Specification [online]. 2004. Available from: http://www.omg.org/docs/formal/04-10-11.pdf [cited 29 June 2007].

[Open Sources07a] Open Sources. Data mining from Wikipedia [online]. 2007. Available from: http://en.wikipedia.org/wiki/Data_mining [cited 29 June 2007].

[Open Sources07b] Open Sources. Digital Rights Management (Wikipedia) [online]. 2007. Available from: http://en.wikipedia.org/wiki/Digital_rights_management [cited 29 June 2007].

[Open Sources07c] Open Sources. EU Directive 95/46/EC on the protection of personal data from Wikipedia [online]. 2007. Available from: http://en.wikipedia.org/wiki/Directive_95/46/EC_on_the_protection_of_personal_data [cited 29 June 2007].

[Open Sources07d] Open Sources. Grounded Theory Methodology (Wikipedia) [online]. 2007. Available from: http://en.wikipedia.org/wiki/Grounded_theory [cited 29 June 2007].

[Open Sources07e] Open Sources. Trusted Computer System Evaluation Criteria (Wikipedia) [online]. 2007. Available from: http://en.wikipedia.org/wiki/TCSEC [cited 29 June 2007].

[Oppermann97] Oppermann, R., Rashev, R. and Kinshuk. Adaptability and Adaptivity in Learning Systems. In Behrooz, A. (ed.), *Knowledge Transfer (volume II)*, pp. 173–179. 1997. Available from: citeseer.ist.psu.edu/oppermann97adaptability.html .

[Osbakk04] Osbakk, P. and Ryan, N. The development of a privacy-enhancing infrastructure: Some interesting findings. 2004. Position Paper for the Ubicomp Privacy: Current Status and Future Directions workshop at UbiComp 2004, Nottingham, UK. Available from: http://www.cs.kent.ac.uk/pubs/2004/1977 .

[Osborn00] Osborn, S., Sandhu, R. and Munawer, Q. Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Trans. Inf. Syst. Secur.*, 3(2):pp. 85–106. 2000.

[Oulasvirta05] Oulasvirta, A., Raento, M. and Tiitta, S. ContextContacts: re-designing SmartPhone's contact book to support mobile awareness and collaboration. In *MobileHCI '05: Proceedings of the 7th international conference on Human computer interaction with mobile devices & services*, pp. 167–174. ACM Press, New York, NY, USA. 2005.

[Palen99] Palen, L. Social, individual and technological issues for groupware calendar systems. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 17–24. ACM Press, New York, NY, USA. 1999.

[Palen03] Palen, L. and Dourish, P. Unpacking "privacy" for a networked world. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 129–136. ACM Press, New York, NY, USA. 2003.

[Patel06] Patel, S., Kientz, J., Hayes, G., Bhat, S. and Abowd, G. D. Farther Than You May Think: An Empirical Investigation of the Proximity of Users to Their Mobile Phones. *UbiComp 2006: Ubiquitous Computing*, pp. 123–140. 2006. Available from: `http://dx.doi.org/10.1007/11853565_8` .

[Patil05] Patil, S. and Lai, J. Who gets to know what when: configuring privacy permissions in an awareness application. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 101–110. ACM Press, New York, NY, USA. 2005.

[Pfitzmann87] Pfitzmann, A. and Waidner, M. Networks without user observability. *Computers and Security*, 6(2):pp. 158–166. 1987.

[Pfitzmann01] Pfitzmann, A. and Köhntopp, M. M. Anonymity, unobservability, and pseudeonymity—a proposal for terminology. In *International workshop on Designing privacy enhancing technologies*, pp. 1–9. Springer-Verlag New York, Inc., New York, NY, USA. 2001.

[Pfleeger02] Pfleeger, C. P. and Pfleeger, S. L. *Security in Computing*. Prentice Hall Professional Technical Reference. 2002.

[PGP Corp.07] PGP Corp. Pretty Good Privacy (PGP) website [online]. 2007. Available from: `http://www.pgp.com/` [cited 29 June 2007].

[Pitt78] Pitt, W. Quotation, William Pitt the elder, British Statesman, 1st Earl of Chatham, Viscount Pitt of Burton-Pynsent, The Great Commoner, 1708-1778 [online]. 1708–1778. Available from: http://en.thinkexist.com/quotation/the-poorest-man-may-in-his-cottage-bid-defiance/537178.html [cited 29 June 2007].

[Pitt01] Pitt, E. and McNiff, K. *Java.rmi: The Remote Method Invocation Guide*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. 2001.

[Pruitt04] Pruitt, D. A Text Template Class for C#. http://www.codeproject.com/csharp/nettemplate.asp. 2004.

[Pu06] Pu, P., Viappiani, P. and Faltings, B. Increasing user decision accuracy using suggestions. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 121–130. ACM Press, New York, NY, USA. 2006.

[Rabitti91] Rabitti, F., Bertino, E., Kim, W. and Woelk, D. A model of authorization for next-generation database systems. *ACM Trans. Database Syst.*, 16(1):pp. 88–131. 1991.

[Radialpoint Inc.06] Radialpoint Inc. Freedom Network by Zero-Knowledge Systems (no longer available as of 1st September 2006) [online]. 2006. Available from: http://www.freedom.net/ [cited 29 June 2007].

[Raento05] Raento, M., Oulasvirta, A., Petit, R. and Toivonen, H. ContextPhone: A Prototyping Platform for Context-Aware Mobile Applications. *IEEE Pervasive Computing*, 4(2):pp. 51–59. 2005.

[Rasch] Rasch, M. Google's data minefield. http://www.securityfocus.com/columnists/383.

[Reimers07] Reimers. Google Maps .NET Control [online]. 2007. Available from: http://www.reimers.dk/ [cited 29 June 2007].

[Reiter98] Reiter, M. K. and Rubin, A. D. Crowds: anonymity for Web transactions. *ACM Transactions on Information and System Security*, 1(1):pp. 66–92. 1998.

[Resnick00] Resnick, P., Kuwabara, K., Zeckhauser, R. and Friedman, E. Reputation systems. *Commun. ACM*, 43(12):pp. 45–48. 2000.

[Rheingold00] Rheingold, H. *The Virtual Community: Homesteading on the Electronic Frontier*. MIT Press, Cambridge, MA, USA. 2000.

[Russinovich05] Russinovich, M. Sony, Rootkits and Digital Rights Management Gone Too Far. http://www.sysinternals.com/blog/2005/10/sony-rootkits-and-digital-rights.html. 2005.

[Saliba01] Saliba, C. Study: Web Privacy Policies Should Be Shorter, Clearer for Consumers. http://www.ecommercetimes.com/story/15084.html. 2001.

[Sandhu94] Sandhu, R. S. and Samarati, P. Access Control: Principles and Practice. *IEEE Communications Magazine*, 32(9):pp. 40–48. 1994. Available from: citeseer.ist.psu.edu/sandhu94access.html .

[Sandhu97] Sandhu, R. and Samarati, P. Authentication, Access Control, and Intrusion Detection. In Tucker, A. B. (ed.), *The Computer Science and Engineering Handbook*, pp. 1929–1948. CRC Press. 1997.

[Satyanarayanan03] Satyanarayanan, M. Privacy: The Achilles Heel of Pervasive Computing? *IEEE Pervasive Computing*, 2(1). 2003.

[Schneier00] Schneier, B. Inside risks: semantic network attacks. *Commun. ACM*, 43(12):p. 168. 2000.

[Segall97] Segall, B. and Arnold, D. Elvin Has Left the Building: A Publish/Subscribe Notification Service with Quenching. In *AUUG97: Proceedings of the 1997 Australian UNIX Users Group*. 1997.

[Shields00] Shields, C. and Levine, B. N. A protocol for anonymous communication over the Internet. In *CCS '00: Proceedings of the 7th ACM conference on Computer and communications security*, pp. 33–42. ACM Press, New York, NY, USA. 2000.

[Shneiderman97] Shneiderman, B. and Maes, P. Direct manipulation vs. interface agents. *interactions*, 4(6):pp. 42–61. 1997.

[Silverman05] Silverman, D. N0 privacy 4 u, LOL!!!!! (AOL IM users sign away right to privacy). *Houston Chronicle Online*. 2005. Available from: http://www.chron.com/disp/story.mpl/business/3081870.html .

[Simon96] Simon, H. A. *The sciences of the artificial (3rd ed.)*. MIT Press, Cambridge, MA, USA. 1996.

[Smale05] Smale, S. and Greenberg, S. Broadcasting information via display names in instant messaging. In *GROUP '05: Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, pp. 89–98. ACM Press, New York, NY, USA. 2005.

[Smith99] Smith, M. and Kollock, P. (eds.). *Communities in Cyberspace*. Routledge, New York, NY, 10001. 1999.

[Smith04] Smith, E. H. S. S. W. Grand Challenges in Information Security: Process and Output. *IEEE Security and Privacy*, 2(1):pp. 67–71. 2004.

[Smith05] Smith, I., Consolvo, S., Hightower, J., Iachello, G., LaMarca, A., Scott, J., Sohn, T. and Abowd, G. Social Disclosure Of Place: From Location Technology to Communications Practices. In *Proceedings of the Third International Conference on Pervasive Computing*, Lecture Notes in Computer Science. Springer-Verlag. 2005.

[Srinivasan95] Srinivasan, R. RPC: Remote Procedure Call Protocol Specification Version 2. 1995.

[Stajano02] Stajano, F. *Security for Ubiquitous Computing*. John Wiley and Sons. 2002. Available from: `http://www.cl.cam.ac.uk/~fms27/secubicomp/` .

[Stajano03] Stajano, F. and Crowcroft, J. *The butt of the iceberg: hidden security problems of ubiquitous systems*, pp. 91–101. Kluwer Academic Publishers, Norwell, MA, USA. 2003.

[Stajano05] Stajano, F. RFID is x-ray vision. *Commun. ACM*, 48(9):pp. 31–33. 2005.

[Stevens90] Stevens, W. R. *UNIX Network Programming*. Prentice-Hall, Upper Saddle River, NJ 07458, USA. 1990. Available from: `citeseer.ist.psu.edu/stevens90unix.html` .

[Swinth02] Swinth, K. R., Farnham, S. D. and Davis, J. P. Sharing Personal Information in Online Community Member Profiles. http://research.microsoft.com/scg/papers/sharing personal information in online community member profiles - with names.pdf. 2002.

[Symantec Corp.07] Symantec Corp. SafeWeb (no longer available) [online]. 2007. Available from: http://www.safeweb.com [cited 29 June 2007].

[Tanenbaum06] Tanenbaum, A. S. and Steen, M. V. *Distributed Systems: Principles and Paradigms*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edn. 2006.

[Trace a Mobile.com07] Trace a Mobile.com. Trace A Mobile [online]. 2007. Available from: http://www.traceamobile.co.uk/ [cited 29 June 2007].

[Tsandilas04] Tsandilas, T. and Schraefel, M. C. Usable adaptive hypermedia systems. *Hypermedia*, 10(1):pp. 5–29. 2004.

[UMTS Forum00] UMTS Forum. Enabling UMTS / Third Generation Services and Applications [online]. 2000. Available from: http://www.umts-forum.org/component/option,com_docman/task,cat_view/gid,171/Itemid,12/ [cited 29 June 2007].

[US Dept. of Health73] US Dept. of Health. Records, Computers and the Rights of Citizens [online]. 1973. Available from: http://aspe.hhs.gov/datacncl/1973privacy/Summary.htm [cited 29 June 2007].

[Viappiani02] Viappiani, P., Pu, P. and Faltings, B. Acquiring User Preferences for Personal Agents. In *AAAI Fall Symposium*. AAAI Press, North Falmouth, MA. 2002. Available from: http://hci.epfl.ch/website/publication-doc/a3iSymposium.pdf .

[W3C98] W3C. P3P Guiding Principles [online]. 1998. Available from: http://www.w3.org/TR/NOTE-P3P10-principles [cited 29 June 2007].

[W3C02] W3C. The Platform for Privacy Preferences 1.0 (P3P1.0) Specification [online]. 2002. Available from: http://www.w3.org/TR/P3P/ [cited 29 June 2007].

[W3C03a] W3C. P3P 1.0: A New Standard in Online Privacy [online]. 2003. Available from: http://www.w3.org/P3P/brochure.html [cited 29 June 2007].

[W3C03b] W3C. Platform for Internet Content Selection (PICS) [online]. 2003. Available from: http://www.w3.org/PICS/ [cited 29 June 2007].

[Warren90] Warren, S. and Brandeis, L. The Right to Privacy. *Harvard Law Review*, 4(5):pp. 193–220. 1890. Available at http://www.lawrence.edu/fast/boardmaw/Privacy_brand_warr2.html.

[Weilenmann04] Weilenmann, A. H. and Leuchovius, P. "I'm waiting where we met last time": exploring everyday positioning practices to inform design. In *NordiCHI '04: Proceedings of the third Nordic conference on Human-computer interaction*, pp. 33–42. ACM Press, New York, NY, USA. 2004.

[Weiser91] Weiser, M. The Computer for the Twenty-First Century. *Scientific American*, 265(3). 1991. Available from: `http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html` .

[Westin67] Westin, A. F. *Privacy and freedom*. New York: Atheneum. 1967.

[Whitten99] Whitten, A. and Tygar, J. D. Why Johnny can't encrypt: A usability evaluation of PGP 5.0. In *8th USENIX Security Symposium*. 1999.

[Winett71] Winett, J. M. Definition of a socket. http://www.ietf.org/rfc/rfc147.txt. 1971.

[Wu06] Wu, M., Miller, R. C. and Garfinkel, S. L. Do security toolbars actually prevent phishing attacks? In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 601–610. ACM Press, New York, NY, USA. 2006.

[Zimmermann91] Zimmermann, P. Why I Wrote PGP. http://www.philzimmermann.com/EN/essays/WhyIWrotePGP.html. 1991.

Appendix A

Research Protocol Form

LANCASTER UNIVERSITY DEPARTMENT OF COMPUTING

RESEARCH PROTOCOL FORM

TITLE OF RESEARCH:
Exploring location privacy management.

PRINCIPAL INVESTIGATOR:
Maomao Wu (supervised by Dr. Adrian Friday)
Department of Computing, Lancaster University, Lancaster. LA1 4YR
Tel: 01524- 510375
E-mail: maomao@comp.lancs.ac.uk

We are conducting a research study on user management of location privacy. We invite you to participate in this study which will involve
a) signing up your mobile phone number to be tracked by a third-party location tracking service;
b) making and receiving requests for your location or the location of other registered users as need dictates;
c) filling in a daily privacy event diary online (or signifying that nothing of interest occurred each day) as appropriate; and,
d) completing an entry questionnaire and exit interview at the start and end of the study respectively.

It is important that you read and understand several principles that apply to all who take part in our studies;
a) taking part in the study is entirely voluntary;
b) personal benefit may not result from taking part in the study, but knowledge may be gained that will benefit others;
c) any significant findings will be discussed with you if you desire;
d) you may withdraw from the study at any time.

The nature of the study, the risks, inconveniences, discomforts, and other pertinent information about the study are discussed below. You are urged to discuss any questions you have about this study with the investigators before you sign this consent. We will also be happy to answer any questions as they arise during the course of our research.

In accord with all of our research protocols, anonymity and confidentiality will be maintained at all times.

BACKGROUND & PURPOSE:

This project will inform our understanding of how people wish to manage information they consider private online. The primary emphasis in this study is on the controlled release of personal location as tracked by locating an individual's mobile phone. We will be interested in when information is and isn't released, frequency and patterns of behaviour, and whether individuals choose to delegate some decisions to the system (set privacy rules).

STUDY PROCEDURE:

The main study procedure for this project is by logging the requests and responses you make online and via SMS for individuals' location, or responding to requests for your location. The diary system is designed to allow you to log your thoughts while using the system for later review. You are being asked to participate in a study that will require the following:

- Signing up to the system (offering your phone number and form of declaration to meet the world-tracker's legal constraints)
- Attending a brief introduction to the system
- Completing a short questionnaire to elicit your previous experience with online systems and mobile phones
- Using the system for a period of approximately 2 months (at will)
- Responding to the daily privacy diary requests (an option is provided to skip a day's entry if nothing significant occurred or you have no time)
- Completing an exit interview to review interesting uses of the system during the study

Note that when writing the data into a project report or any other form of documentation, steps are taken to ensure anonymity for all those involved in the study. Confidentiality

will be maintained at all times. Any recordings that are made or any materials collected are the property of the researcher, will be kept in a secure environment and will be destroyed at the conclusion of the research.

Note also that your phone number and details of how to use our system will only be made available to anyone participating in the experiment. The details in the world-tracker phone location service will not be given out at any time. The world-tracker account will be closed at the conclusion of the research.

RISKS OF PARTICIPATION IN THE STUDY:

The risks of participating in this study are minimal.

It is the investigators' intention to anonymise any research findings or reports and thereby ensure that your identity in these studies will remain confidential at all times.

However, there is a small risk of inadvertent disclosure. In addition, your identity and the study findings may be disclosed through legal action - when, for example, non-disclosure would constitute contempt of court. However, as far as possible, we will ensure that any such disclosure is unlikely to have an adverse effect on you, on your family members, and on your family relationships.

BENEFITS:
There may be no personal benefit to you from participating in this project. The benefits of this research may include learning more about management of private information online.

The research should provide more sophisticated, empirically-based understandings of how individuals choose to manage such information and how to construct better privacy aware applications and supporting software. The project will provide an opportunity to examine and report on our findings, if desired.

COSTS AND COMPENSATION:

You will not be paid for participating in this study. However, we intend to offer a gift certificate as an incentive to participate, and also to offset the cost of any text messages incurred during the experiment.

There is unlikely to be any significant cost - financial or other - to you for participation in the study. No additional charges are made to you as an individual or your mobile phone account other than charges associated with any text messages you may choose to send.

CONFIDENTIALITY:

All information collected in this study belongs to the fieldworker and will be maintained in a confidential manner at Lancaster University. Nobody, other than the fieldwork researcher and the research team, will have access to the data. Any tape recordings will be destroyed at the end of the project. Although rare, it is possible that disclosure may be required by law. Otherwise, the information will not be disclosed to third parties without your permission. If the study is published, your name and institution will be kept confidential.

PEOPLE TO CONTACT:

If you have further questions related to this research study, you may call the Principal Investigator, Maomao Wu at 01524-510375.

If you are not satisfied with the manner in which this study is being conducted, you may report (anonymously if you so choose) any complaints to Yvonne Fox, Secretary to the Ethical Committee, Lancaster University by calling 01524-592068 , emailing y.fox@lancaster.ac.uk; or addressing a letter to Y.Fox, Ethical Committee, Lancaster University, LA1 4YR.

SUBJECT'S CONSENT:

I understand that I am free to refuse to participate in this research project or to withdraw my consent and discontinue participation in the project at any time without prejudice.
I understand that I will not be paid to participate in this study.
I have had the opportunity to fully discuss this investigation and the procedure(s) with a study investigator.
All my questions regarding this project have been answered.

I agree to participate in the project as described above.

Subject's signature

Date signed

Subject's printed name

A COPY OF THIS FORM HAS BEEN GIVEN TO ME

Subject's initials

I have discussed with the subject, (and, if required, the subject's guardian) the procedure(s) described above and the risks involved; I believe he/she understands the contents of the consent form, and is competent to give a legally effective and informed consent.

Signature of Investigator

Date signed

Appendix B

Opening Questionnaire

1. About You

Full Name		
Gender	☐ Male	☐ Female
Age or Age Group (Age = your exact age, Age Group = a range of 5 years, e.g., 25~30.)		

2. Computing and Mobile Phone Experience

Do you use a personal computer or laptop?	☐ at work	☐ at home
Do you use the Internet?	☐ at work	☐ at home
How many years have you used a computer?		
What kind of computer experience do you have on a scale 1~5? (1 = Beginner and 5 = Expert)	☐1 ☐2 ☐3 ☐4 ☐5	
How many years have you used a mobile phone?		
Do you often carry a mobile phone with you while away from the home or office?	☐ Yes	☐ No

What do you use your mobile for?	☐ Voice Call ☐ Text messages ☐ Multimedia messages ☐ Email ☐ Internet Surfing ☐ Other
Approximately how many SMS messages do you send each week? (you can give a range)	

3. Your Attitude to Personal Information

Tick the most appropriate one	Strongly disagree	Somewhat Disagree	Somewhat Agree	Strongly agree
Consumers have lost all control over how personal information is collected and used by companies	☐	☐	☐	☐
Most businesses handle the personal information they collect about consumers in a proper and confidential way	☐	☐	☐	☐
Existing laws and organizational practices provide a reasonable level of protection for consumer privacy today	☐	☐	☐	☐

Do you think you would ever reject or ignore requests for your location from friends, family, colleagues or strangers?

☐ Yes ☐ No

If Yes, please describe a scenario.

Do you think you would ever intentionally alter the precision of information given out about your location?

☐ Yes ☐ No

If Yes, please describe a scenario.

Do you think that you'd like to be involved in ALL decisions regarding disclosing your location?

☐ Yes ☐ No

Do you think you'd be able to create a rule that would enable you to manage requests for your location automatically?

☐ Yes ☐ No

If Yes, please imagine creating a rule for one of the following people who wants to know your location: partner, parents, boss, coworker, close friend, student, or roommate.

Appendix C

End-of-trial Survey Form

Section 1

Please tick the most appropriate choice	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
It is important that I am aware when my private information is disclosed	☐	☐	☐	☐	☐
I liked to be informed every time my location was released	☐	☐	☐	☐	☐
I was aware of the disclosure of my location while using the system	☐	☐	☐	☐	☐
It was useful to have notifications to my phone, email and web browser	☐	☐	☐	☐	☐
Please rank how effectively you found the following methods of notification (1 is most effective, 3 is lest effective)	___ SMS ___ Email ___ Popup windows in web pages				

Section 2

Please tick the most appropriate choice	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
It is important that I have control whenever my private information is being disclosed	☐	☐	☐	☐	☐
I found it easy to accept/reject an incoming request using the web pages	☐	☐	☐	☐	☐
I found it easy to accept/reject an incoming request using SMS messages	☐	☐	☐	☐	☐
It is useful to have privacy controls on both my phone and on the web	☐	☐	☐	☐	☐
Did you realise that you can set expiration time for a request?		☐ Yes		☐ No	
Do you think they are useful?		☐ Yes		☐ No	
Did you realise that you can provide extra information in a request or response?		☐ Yes		☐ No	
Do you think it is useful?		☐ Yes		☐ No	

Section 3

Please tick the most appropriate choice	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
It is important that there is a record of what information people found out about me	☐	☐	☐	☐	☐
Are you aware that the system maintained a history of every location request you made or received?	☐ Yes ☐ No				
I felt more comfortable knowing that the system kept a history of my location disclosure	☐	☐	☐	☐	☐
Did you check the history of requests and disclosures?	☐ Yes ☐ No				

Section 4

Please tick the most appropriate choice	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
It is important that I can adjust my involvement in privacy decisions to reduce the number of times I'm interrupted or the effort needed to manage my privacy	☐	☐	☐	☐	☐
I found that the system did allow me to find an agreeable balance between the effort and interruption	☐	☐	☐	☐	☐
Privacy rules were useful for me	☐	☐	☐	☐	☐
Privacy rules reduced the amount of interaction involved in managing my privacy	☐	☐	☐	☐	☐
Privacy rules reduced the amount I was interrupted	☐	☐	☐	☐	☐
It was easy to create privacy rules via web pages	☐	☐	☐	☐	☐
It was easy to create privacy rules via SMS messages	☐	☐	☐	☐	☐
I found user groups and group rules useful	☐	☐	☐	☐	☐
I prefer to create privacy rules (1 is most preferred, 3 is lest preferred)	___ before receiving any requests ___ when I am receiving a request ___ after I am familiar with the system (i.e., received and processed a few location requests)				
How many privacy rules have you created? (please give a number)					

Section 5

Please tick the most appropriate choice	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
It is important that I am able to respond to changes in circumstance by adjusting whether and how my private information is released	☐	☐	☐	☐	☐
I knew before I used the system how and to whom I wanted my private information to be released	☐	☐	☐	☐	☐
I changed my mind about releasing information to an individual during the trial	☐	☐	☐	☐	☐
The system allowed me to make different decisions on disclosing my location depending on the situation	☐	☐	☐	☐	☐
I prefer to process location requests one-by-one interactively	☐	☐	☐	☐	☐
I prefer to create privacy rules to automate location request processing	☐	☐	☐	☐	☐
I liked to be able to modify details of my privacy rules	☐	☐	☐	☐	☐
Did you realise you could change the granularity of location released by a rule	☐ Yes ☐ No				
It is useful to have granularity control in a privacy rule	☐	☐	☐	☐	☐
It is useful to have date and time constraints in a privacy rule	☐	☐	☐	☐	☐
What triggered modification of a privacy rule?					

Section 6

Please tick the most appropriate choice	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
The location disclosed by the system is sensitive information that I would not carelessly release to anyone	☐	☐	☐	☐	☐
Do you feel your location privacy was protected by the system?		☐ Yes		☐ No	
Did you find the location returned by the system sufficient to work out where someone was?		☐ Yes		☐ No	
Do you find the location tracking application useful for you? (why?)		☐ Yes		☐ No	
The location tracking service is commercially available and it costs around 20~25 pence for each location disclosure. Would you be willing to pay the service at that price? (why?)		☐ Yes		☐ No	
Why did you make location requests instead of using traditional form of communication, e.g. a phone call?					
How did you respond to two requests from 'Jessika Silversmit'? Why?					
What did you like about the system?					
How do you feel that system could have been improved?					

Appendix D

Sample Email Messages

Email reminding a user to write their privacy diary entry

Please write today's privacy diary (Mon, 28 May 2007)

pm@comp.lancs.ac.uk

To: maomao@comp.lancs.ac.uk

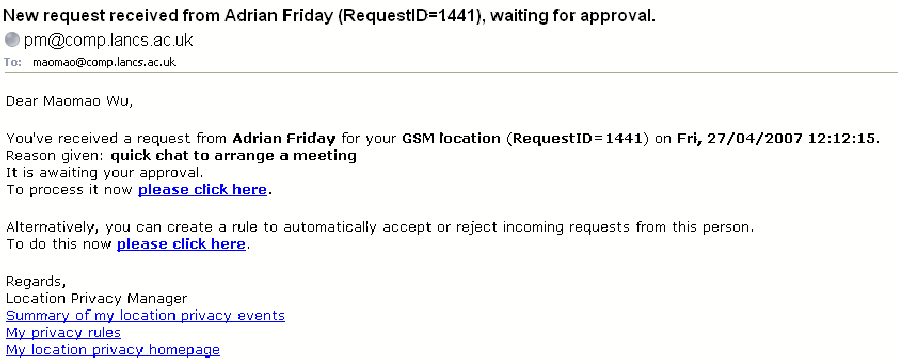
Dear Maomao Wu,

Have you written your privacy diary today?
If not, please edit your privacy diary today by selecting one of the following options:

- I have no time to leave any comments today.
- I have no comments because nothing interesting happened today.
- I'd like to leave a comment about the privacy system or an event today.

Regards,
Location Privacy Manager
Mon, 28/05/2007 18:00:00
Summary of my location privacy events
My privacy rules
My location privacy homepage

Email notifications that a request has been received and is waiting for processing



New request received from Adrian Friday (RequestID=1441), waiting for approval.

pm@comp.lancs.ac.uk

To: maomao@comp.lancs.ac.uk

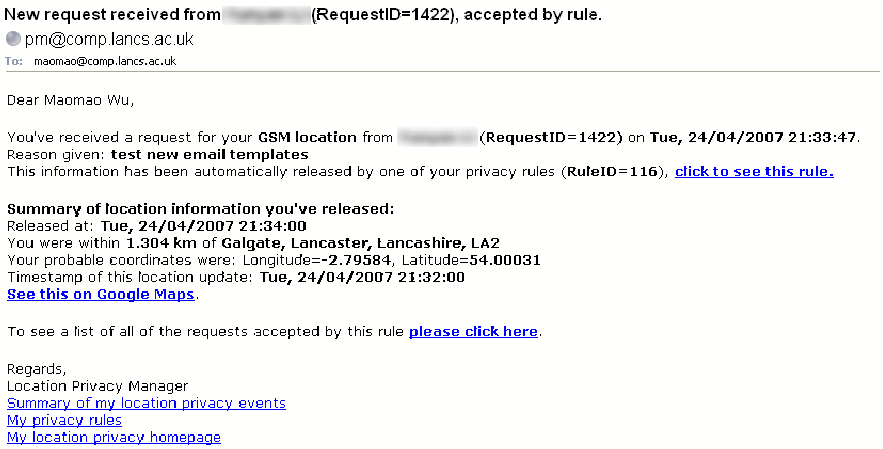
Dear Maomao Wu,

You've received a request from **Adrian Friday** for your **GSM location** (**RequestID=1441**) on **Fri, 27/04/2007 12:12:15.**
Reason given: **quick chat to arrange a meeting**
It is awaiting your approval.
To process it now **please click here**.

Alternatively, you can create a rule to automatically accept or reject incoming requests from this person.
To do this now **please click here**.

Regards,
Location Privacy Manager
Summary of my location privacy events
My privacy rules
My location privacy homepage

Email notifying the user that a rule accepted a request on their behalf



New request received from [redacted] **(RequestID=1422), accepted by rule.**

pm@comp.lancs.ac.uk

To: maomao@comp.lancs.ac.uk

Dear Maomao Wu,

You've received a request for your **GSM location** from [redacted] **(RequestID=1422)** on **Tue, 24/04/2007 21:33:47**.
Reason given: **test new email templates**
This information has been automatically released by one of your privacy rules (**RuleID=116**), **click to see this rule.**

Summary of location information you've released:
Released at: **Tue, 24/04/2007 21:34:00**
You were within **1.304 km** of **Galgate, Lancaster, Lancashire, LA2**
Your probable coordinates were: Longitude=**-2.79584**, Latitude=**54.00031**
Timestamp of this location update: **Tue, 24/04/2007 21:32:00**
See this on Google Maps.

To see a list of all of the requests accepted by this rule **please click here**.

Regards,
Location Privacy Manager
Summary of my location privacy events
My privacy rules
My location privacy homepage

Email response containing the user's location when a request is accepted

Previously sent request (RequestID=1536), accepted by [redacted].

pm@comp.lancs.ac.uk

To: maomao@comp.lancs.ac.uk

Dear Maomao Wu,

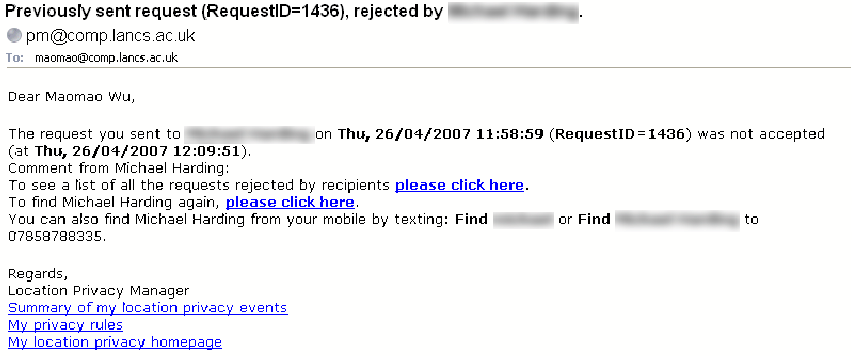
Your request for [redacted]'s location (**RequestID=1536**) has been accepted (at **Wed, 23/05/2007 12:13:26**).

Information released by Weiou Wu:
Released at: **Wed, 23/05/2007 12:13:51**
Weiou Wu was within **0.05 km** of **Boarhills, St Andrews, Fife, KY16**
Probable coordinates were: (Longitude=**-2.72953**,Latitude=**56.32047**)
Timestamp of this location update: **Wed, 23/05/2007 12:13:00**
See this on Google Maps
Comment from Weiou Wu:

To see a list of the previous requests accepted by recipients **please click here**.
To find Weiou Wu again, **please click here**.
You can also find Weiou Wu from your mobile by texting: **Find** [redacted] or **Find** [redacted] to 07858788335.

Regards,
Location Privacy Manager
Summary of my location privacy events
My privacy rules
My location privacy homepage

Email reply when the user's location request has been rejected



Previously sent request (RequestID=1436), rejected by [redacted].

pm@comp.lancs.ac.uk

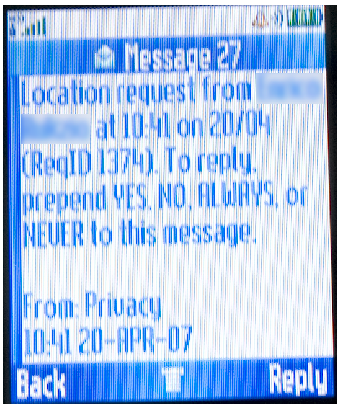
To: maomao@comp.lancs.ac.uk

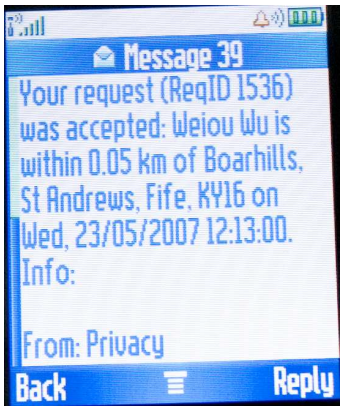
Dear Maomao Wu,

The request you sent to [redacted] on **Thu, 26/04/2007 11:58:59** (**RequestID=1436**) was not accepted (at **Thu, 26/04/2007 12:09:51**).
Comment from Michael Harding:
To see a list of all the requests rejected by recipients **please click here**.
To find Michael Harding again, **please click here**.
You can also find Michael Harding from your mobile by texting: **Find** [redacted] or **Find** [redacted] to 07858788335.

Regards,
Location Privacy Manager
Summary of my location privacy events
My privacy rules
My location privacy homepage

Sample SMS messages





Appendix E

Sample Message Templates

Email template for privacy diary reminders

```
<html><body>
<p><font face=Verdana size=2>
<p>Dear [%FirstName%] [%LastName%],
<p>
Please edit your privacy diary today by selecting one of the following options:
<ul>
<li><a target=_blank href="[%VirtualDir%]/
Welcome.aspx?Token=[%Token%]&ReturnUrl=%2FMyPrivacyWeb%2FPersonal%2FEditPrivacyDiary.aspx
%3Fdate%3D[%Today%]%26option%3D0">I have no time to leave any comments today.</a><br/>
<li><a target=_blank href="[%VirtualDir%]/
Welcome.aspx?Token=[%Token%]&ReturnUrl=%2FMyPrivacyWeb%2FPersonal%2FEditPrivacyDiary.aspx
%3Fdate%3D[%Today%]%26option%3D1">I have no comments because nothing interesting happened
today.</a><br/>
<li><a target=_blank href="[%VirtualDir%]/
Welcome.aspx?Token=[%Token%]&ReturnUrl=%2FMyPrivacyWeb%2FPersonal%2FEditPrivacyDiary.aspx
%3Fdate%3D[%Today%]%26option%3D2">I'd like to leave a comment about the privacy system or
an event today.</a><br/>
</ul>
<p>
Regards,<br/>
Location Privacy Manager<br/>
[%DateTimeToday%]<br/>
<a target=_blank href="[%VirtualDir%]/
Welcome.aspx?Token=[%Token%]&ReturnUrl=%2FMyPrivacyWeb%2FPersonal%2FMyPrivacyManager.aspx
">Summary of my location privacy events</a><br/>
<a target=_blank href="[%VirtualDir%]/
Welcome.aspx?Token=[%Token%]&ReturnUrl=%2FMyPrivacyWeb%2FPersonal%2FGridViewMyRules.aspx"
>My privacy rules</a><br/>
<a target=_blank href="[%VirtualDir%]/Welcome.aspx">My location privacy homepage</a><br/>
</body></html>
```

Email template for notifying users of waiting requests

```
<html><body>
<p><font face=Verdana size=2>
<p>Dear [%FirstName%] [%LastName%],
<p>
You've received a request from <b>[%Requester%]</b> for your <b>[%InfoName%]</b>
(<b>RequestID=[%RequestID%]</b>) on <b>[%ReqDateTime%]</b>.<br/>
Reason given: <b>[%Purpose%]</b><br/>
It is awaiting your approval.<br/>
To process it now
<b><a target=_blank href="[%VirtualDir%]/
Welcome.aspx?Token=[%Token%]&ReturnUrl=%2FMyPrivacyWeb%2FPersonal%2FGridViewReceivedReque
sts.aspx%3Fid%3D[%RequestID%]">please click here</a></b>.
<p>Alternatively, you can create a rule to automatically accept or reject incoming
requests from this person. <br/>
To do this now <b><a target=_blank href="[%VirtualDir%]/
Welcome.aspx?Token=[%Token%]&ReturnUrl=%2FMyPrivacyWeb%2FPersonal%2FSuggestRule.aspx%3Fre
qID%3D[%RequestID%]">please click here</a></b>.<br/>
<p>
Regards,<br/>
Location Privacy Manager<br/>
<a target=_blank href="[%VirtualDir%]/
Welcome.aspx?Token=[%Token%]&ReturnUrl=%2FMyPrivacyWeb%2FPersonal%2FMyPrivacyManager.aspx
">Summary of my location privacy events</a><br/>
<a target=_blank href="[%VirtualDir%]/
Welcome.aspx?Token=[%Token%]&ReturnUrl=%2FMyPrivacyWeb%2FPersonal%2FGridViewMyRules.aspx"
>My privacy rules</a><br/>
<a target=_blank href="[%VirtualDir%]/Welcome.aspx">My location privacy homepage</a><br/>
</body></html>
```

Email template for notifying a user that their request was accepted

```
<html><body>
<p><font face=Verdana size=2>
<p>Dear [%FirstName%] [%LastName%],
<p>
Your request for <b>[%Requestee%]</b>'s location (<b>RequestID=[%RequestID%]</b>) has
been accepted (at <b>[%ReqDateTime%]</b>).
<p><b>Information released by [%Requestee%]:</b><br>
Released at: <b>[%ReleaseDT%]</b><br>
[%Requestee%] was within <b>[%Radius%] km</b> of <b>[%Address%]</b><br>
Probable coordinates were: (Longitude=<b>[%Longitude%]</b>,Latitude=<b>[%Latitude%]</
b>)<br>
Timestamp of this location update: <b>[%InfoDateTime%]</b><br>
<b><a target=_blank href="[%VirtualDir%]/
Welcome.aspx?Token=[%Token%]&ReturnUrl=%2FMyPrivacyWeb%2FPersonal%2FShowMap.aspx%3Fid%3D[
%ReleaseID%]">See this on Google Maps</a></b><br>
Comment from [%Requestee%]: <b>[%RecipientComment%]</b>
<p>To see a list of the previous requests accepted by recipients <b><a target=_blank
href="[%VirtualDir%]/
Welcome.aspx?Token=[%Token%]&ReturnUrl=%2FMyPrivacyWeb%2FPersonal%2FGridViewSentRequests.
aspx%3Fshow%3DAccepted%26id%3D[%RequestID%]">please click here</a></b>.<br>
To find [%Requestee%] again, <b><a target=_blank href="[%VirtualDir%]/
Welcome.aspx?Token=[%Token%]&ReturnUrl=%2FMyPrivacyWeb%2FPersonal%2FGsmLocationRequest.as
px">please click here</a></b>.<br>
You can also find [%Requestee%] from your mobile by texting: <b>Find
[%RequesteeNickName%]</b> or <b>Find [%Requestee%]</b> to 07858788335.<br>
<p>
Regards,<br/>
Location Privacy Manager<br/>
<a target=_blank href="[%VirtualDir%]/
Welcome.aspx?Token=[%Token%]&ReturnUrl=%2FMyPrivacyWeb%2FPersonal%2FMyPrivacyManager.aspx
">Summary of my location privacy events</a><br/>
<a target=_blank href="[%VirtualDir%]/
Welcome.aspx?Token=[%Token%]&ReturnUrl=%2FMyPrivacyWeb%2FPersonal%2FGridViewMyRules.aspx"
>My privacy rules</a><br/>
<a target=_blank href="[%VirtualDir%]/Welcome.aspx">My location privacy homepage</a><br/>
</body></html>
```

Email template notifying a web only user that their request was accepted

```
<html><body>
<p><font face=Verdana size=2>
<p>Dear [%FirstName%] [%LastName%],
<p>
Your request for <b>[%Requestee%]</b>'s location (<b>RequestID=[%RequestID%]</b>) has
been accepted (at <b>[%ReqDateTime%]</b>).
<p><b>Information released by [%Requestee%]:</b><br>
Released at: <b>[%ReleaseDT%]</b><br>
[%Requestee%] was within <b>[%Radius%] km</b> of <b>[%Address%]</b><br>
Probable coordinates were: (Longitude=<b>[%Longitude%]</b>,Latitude=<b>[%Latitude%]</
b>)<br>
Timestamp of this location update: <b>[%InfoDateTime%]</b><br>
<b><a target=_blank href="[%VirtualDir%]/
Welcome.aspx?Token=[%Token%]&ReturnUrl=%2FMyPrivacyWeb%2FPersonal%2FShowMap.aspx%3Fid%3D[
%ReleaseID%]">See this on Google Maps</a></b><br>
Comment from [%Requestee%]: <b>[%RecipientComment%]</b>
<p>To see a list of the previous requests accepted by recipients <b><a target=_blank
href="[%VirtualDir%]/
Welcome.aspx?Token=[%Token%]&ReturnUrl=%2FMyPrivacyWeb%2FPersonal%2FGridViewSentRequests.
aspx%3Fshow%3DAccepted%26id%3D[%RequestID%]">please click here</a></b>.<br>
To find [%Requestee%] again, <b><a target=_blank href="[%VirtualDir%]/
Welcome.aspx?Token=[%Token%]&ReturnUrl=%2FMyPrivacyWeb%2FPersonal%2FGsmLocationRequest.as
px">please click here</a></b>.<br>
<p>
Regards,<br/>
Location Privacy Manager<br/>
<a target=_blank href="[%VirtualDir%]/
Welcome.aspx?Token=[%Token%]&ReturnUrl=%2FMyPrivacyWeb%2FPersonal%2FMyPrivacyManager.aspx
">Summary of my location privacy events</a><br/>
<a target=_blank href="[%VirtualDir%]/
Welcome.aspx?Token=[%Token%]&ReturnUrl=%2FMyPrivacyWeb%2FPersonal%2FGridViewMyRules.aspx"
>My privacy rules</a><br/>
<a target=_blank href="[%VirtualDir%]/Welcome.aspx">My location privacy homepage</a><br/>
</body></html>
```

Sample Templates of SMS messages

Template of SMS message notifying that a received request is waiting for processing:
Location request from [%Requester%] at [%ReqDateTime%] (ReqID [%RequestID%]). To reply, prepend [%ControlMessage%] to this message. [%Purpose%]

Template of SMS message notifying that a previously sent request was accepted:
Your request (ReqID [%RequestID%]) was accepted: [%Requestee%] is within [%Radius%] km of [%Address%] on [%InfoDateTime%]. [%RecipientComment%]

Template of SMS message notifying that a previously sent request was not accepted:
Your request to [%Requestee%] (ReqID [%RequestID%]) was not accepted on [%ReleaseDT%]. [%RecipientComment%]