NTNU
Innovation and Creativity

Privacy handling in context dissemination

Silje Bentzen Egeland

Master of Science in Communication Technology
Submission date: June 2006
Supervisor: Per-Oddvar Osland, ITEM

Norwegian University of Science and Technology
Department of Telematics

Problem Description

As information about users becomes more available in pervasive environments, the need for privacy handling grows in importance. In provisioning of context-aware applications, a considerable amount of sensitive information is being used. A user needs to control the extent of access to his/her context information. The work carried out in this thesis should contain

- literature study / background related to privacy and dissamenation of context information
- propose and design mechanisms for privacy handling
- if possible, an implementation that demonstrates results.

Assignment given: 16. January 2006
Supervisor: Per-Oddvar Osland, ITEM

Preface

This master thesis was carried out at the Department of Telematics, at the Norwegian University of Science and Technology, in the period January 2006 until June 2006.

I would like to thank my supervisor Per-Oddvar Osland for valuable comments and suggestions during the work on this master thesis.

Trondheim, June 2006

Silje Bentzen Egeland

Abstract

Pervasive environments are characterized by ubiquitous, mobile and embedded computing devices and wireless networking. The vision is an environment where the technology resides in the background ready to be used when it is needed. An essential part of such environments is context aware applications and context information. A context aware system exploits context information to provide relevant services or information to an entity, where relevancy depends on the entity's task. As a user, the employment of such systems involves revealing a lot of personal data. Context information can divulge a lot of sensitive information which represents a threat to a person's privacy.

This master thesis looks into privacy handling in pervasive computing environments. The object is to propose a solution on how a user can control the extent of access to his or her context information. In order to identify the most important privacy concerns in the implementation of a context management system, privacy principles are looked into and privacy challenges in consequence of pervasive computing are evaluated. The different strategies for handling privacy are pointed out, such as legislation, self-regulation and technology. Among these, technology is further looked into, first through an evaluation of existing solutions and research projects, then through design an implementation of a possible solution.

The principles which are identified to be most important to handle are a user's awareness of data collection and the possibility to restrict this collection. In addition it is pointed out the importance of making the system convenient to use. The heterogeneity of different users' privacy preferences implies that some kind of personalization of the system should be present. A design and an implementation are presented which propose a solution where a user can constrain the access to his or her personal data, based on other users' identities, his or her present situation and the type of context information the other users want to receive. The system also includes functionality to abstract details away from the context information which is disseminated to other system users/entities.

Table of Contents

Preface I
Abstract II
List of figures VI
List of tables VII
Abbreviations VIII
Definitions IX
1. Introduction 1
1.1 Motivation 1
1.2 Scenario 1
1.3 Problem statement 2
1.4 Report organization 3
2. Background 5
2.1 Pervasive computing environments 5
2.2 Context 5
2.2.1 Classification 6
2.2.2 Characteristics 6
2.3 Architecture – Collection of context data 7
2.4 Context Aware Applications 8
3. Privacy 9
3.1 Definitions 9
3.2 Privacy History and Legislation 10
3.3 The Importance of Privacy 11
3.3.1 User perspectives 12
3.4 Features with pervasive environments that challenges privacy 13
3.4.1 Crossing new borders 13
3.4.2 Context aware systems 14
3.5 Privacy Strategies 15
4. Privacy system technology 16
4.1 Design Principles 16
4.2 Work done in the area of privacy protection 18
4.2.1 Privacy Enhancing Technology (PET) 18
4.2.2 Privacy policies and policy announcements 18
4.2.3 Anonymisers 19
4.2.4 Transparency and Trust tools 19
4.2.5 Data tagging – proximity, locality 20
4.3 Examples of privacy systems 20
4.3.1 Identity Management 20
4.3.2 Policy tagging of data elements 22
4.3.3 CoBrA – Context Broker Architecture 24
4.3.4 E-wallet 25
4.4 Motivation and theoretical foundation 27
4.4.1 Evaluation of existing systems 27
4.4.2 Differentiable and customizable privacy handling 28
4.4.3 Definitions 29
4.4.4 The Privacy Policy Enforcer (PPE) 30
5. Enabling technologies 33

5.1 Akogrimo 33
5.1.1 The context management architecture 33
5.1.2 Context model 35
5.1.3 Collecting context data 36
5.1.4 Using and inferring context data 36
6. Design and Implementation 38
6.1 Method 38
6.1.1 Rational Unified Process (RUP) 38
6.2 Overall Description and Requirements 40
6.3 Use Cases 42
6.3.1 Create Privacy Policy 43
6.3.2 Handle request from consumer 45
6.4 Design 46
6.4.1 The Context Model 46
6.4.2 The Privacy Model 47
6.4.3 Information model 52
6.4.4 Message Sequence Charts (MSC) 53
6.4.5 System behaviour 56
6.4.6 Further design which will not be implemented 59
6.5 Implementation 60
6.5.1 Overview 60
6.5.2 contextmanger.entities 60
6.5.3 The contextManager package 61
6.5.4 The privacyManager package 63
6.5.5 The userinterface package 65
6.6 Testing 66
6.6.1 Test environment 66
6.6.2 Test results 66
6.6.3 Comment on test results 67
7. Discussion 68
7.1 Achievements 68
7.2 Coverage of the privacy design principles 70
7.2.1 Notice 70
7.2.2 Choice and consent 70
7.3 User perspectives 70
7.4 Employment of the PPE system 71
7.4.1 Student scenario 72
7.4.2 Using the PPE system 72
7.4.3 Discussion – Added value compared to solutions used today 72
8. Conclusion 74
8.1 Future work 74
References 76
Appendix A: Delivery details 78
A-1 Setup 78
A-2 Execution 79
A-2-1 Execution in order to test the existing privacy policy: 79
A-2-2 Execution in order to create a new privacy policy and test these: 81
Appendix B: Test details 83
B-1 Input 83

B-2 Tests 84

List of figures

Figure 2-1: Classification of context types 6
Figure 2-2: Layered conceptual framework for context-aware systems 7
Figure 3-1: The 1970s' "Horror vision" 11
Figure 4-1: P3P example 19
Figure 4-2: Identity management 21
Figure 4-3: Policy tagging 22
Figure 4-4: Model of the intelligent context broker 24
Figure 4-5: An overview of the Semantic Web environment 25
Figure 4-6: The main steps involved in processing a query submitted to an e-Wallet 26
Figure 4-7: The Privacy Policy Enforcer 31
Figure 4-8: Communication of context information to a 3rd part via a management system 32
Figure 5-1: Overview of the Akogrimo layers 33
Figure 5-2: Generic flow of context data 34
Figure 5-3: Context Manager architecture 34
Figure 5-4: User-oriented context 35
Figure 5-5: Collecting context data 36
Figure 5-6: Using and inferring context data 37
Figure 6-1: The RUP workflow 38
Figure 6-2: Actors and Use Cases relevant for context 42
Figure 6-3: Use Case: Create Privacy Policy 43
Figure 6-4: Use Case: Handle request from consumer 45
Figure 6-5: Overview of the system architecture 46
Figure 6-6: The Communication flow in Privacy handling 47
Figure 6-7: Process in the progress of privacy handling 51
Figure 6-8: Information model 52
Figure 6-9: MSC – Creating the Privacy Policy 53
Figure 6-10: MSC – Request 54
Figure 6-11: MSC – Subscription 55
Figure 6-12: Processes in the context handler 56
Figure 6-13: Procedure – send context update 57
Figure 6-14: Processes in privacy handler 58
Figure 6-15: Package overview 60
Figure 6-16: The contextManager package 61
Figure 6-17: The privacyManager package 63
Figure A-0-1: Screen shot of Eclipse Workspace 78
Figure A-0-2: Screen shot: Import of existing project into Eclipse Workspace 78
Figure A-3: Run the program in Eclipse 79

List of tables

Table 6-1: The system requirements 41
Table 6-2: The objects and parameters of the privacy policy 47
Table 6-3: Definitions of Detail level 48
Table 6-4: Definition of Current context values of the situation “Work” 49
Table 6-5: Example of information contained in a privacy policy 50
Table 6-6: Test description 66
Table 6-7: Test summary 67
Table 7-1: Fulfillment of the system requirements 69
Table A-1: The enclosed Privacy Policy 79

Abbreviations

A4A	-	Authentication Authorization Authentication Accounting
API	-	Application Programming Interface
CCPP	-	Composite Capabilities/Preference Profile
CoBrA	-	Context Broker Architecture
GPS	-	Global Positioning System
GSM	-	Global System for Mobile communications
HTML	-	Hyper Text Markup Language
HTTP	-	Hyper Text Transfer Protocol
ID	-	Identification
IM	-	Instant Messaging
MSC	-	Message Sequence Chart
OWL	-	Ontology Web Language
PawS	-	Privacy Awareness System
PPE	-	Privacy Policy Enforcer
P3P	-	Privacy Preferences Protocol
RDF	-	Resource Description Framework
RFID	-	Radio Frequency Identification
RUP	-	Rational Unified Process
SIP	-	Session Initiation Protocol|
SLP	-	Service Location Protocol
SOUPA	-	Standard Ontology for Ubiquitous and Pervasive Applications
XML	-	Extensible Markup Language
UML	-	Unified Modelling Language
URI	-	Uniform Resource Identifier
URL	-	Uniform Resource Locator
UTM	-	Universal Transverse Mercator
W3C	-	World Wide Web Consortium
WLAN	-	Wireless Local Area Network

Definitions

Privacy policy	Rules for how to enforce privacy
Context consumer	A user/entity that receives context information about other users/entities
Context provider	An user/entity that has a context and communicates this information to a management system
Context owner	The user/entity that has the legal right to the given context
Privacy policy	This refers to a context owner's preferences on how his or her privacy should be maintained.
Access rights	An access right is the right to receive context information about another user or system entity.
Access to context information	Access to context information is to be able to receive context information.
Request	A request is a querying from a context consumer to receive context information
Subscription	A subscription is a request which a context consumer subscribes to. Each time the context changes for a particular entity the consumer will be noticed.

1. Introduction

1.1 Motivation

New types of mobile and embedded computing devices make the vision of a computing environment where devices, software agents, and services integrate and cooperate in support of human objectives a possible reality. Characteristics of this system are the use of wireless networking, sensor-rich environments, mobile and wearable computing devices, and intelligent human-computer interfaces. A public WLAN makes it possible to get an online connection wherever you go. In situations where your mobile user equipment is not able to meet your needs numerous publicly available devices are ready to be used. This scenario is part of what we call a pervasive or ubiquitous computing environment.

A pervasive computing environment introduces new challenges for security and privacy systems. Conventional computer use focuses on one single device that might interconnect with a few other devices across a network. A pervasive environment gives a whole new situation where a collection of different devices collaborate to perform a service. Context-awareness is a software design approach that is particularly appropriate for pervasive computing. This software relies on context information that is being collected by sensors and embedded devices in a pervasive computing environment. The information is being processed in order to make decisions about how to adapt dynamically to meet user requirements [14] [19].

The collected information is often of personal matter. In everyday life we make decisions all the time about giving out personal information or not. Normally when we agree to give out the requested information it is because we trust the party that is asking or because we consider the information to be insignificant. What most people do not realize is that small pieces of information can divulge a lot of information when put together. In a pervasive environment we often deal with machines that can reason about the collected information on a semantic level, i.e. the machine can understand the data. In addition, the amount of collected information has increased. Combined with the possibility to process large quantities of information limitless applications might be created [8].
Due to this change in computer use privacy has risen to be of great concern. It is important to deal with these privacy issues beforehand. Once distrust has been established among users it might be difficult to get anyone to use such a system. No one will trust a system that subjects them to unexpected assaults on their privacy, although it enables them to do lots of other useful things.

1.2 Scenario

In order to illustrate a possible service in a pervasive environment and give a picture of the necessity of privacy handling in such systems, a scenario will be outlined in this section. The scenario gives an example of a computing system which exploits the use of context information to offer a service.

Job scenario: Tom has a job where he is moving around a lot. He is working both from his home office and the company's office in addition to attending meetings with business associates quite frequently. In this kind of job it is convenient with a system that keeps track of available services and devices in addition to the location and occupation of all the employees. In Tom's office this is accomplished by the use of a context management system. The system collects information about system entities (e.g. devices, services, users) and makes it available to the users of the system. All the employees can send requests to the system or subscribe to information about their colleagues. They can also get information about all available services and devices in a given situation.

The context management system is not turned off when working hours are over. The system is still useful since Tom still might want to use some of the services or devices available to him or maybe get in contact with a colleague. As the system also includes Tom's family and friends it is nice to be able to get information about them after working hours as well.

However, in some situations Tom wants to keep his privacy. The context management system therefore uses a privacy enforcement mechanism that enables all the users to define their personal privacy preferences. By using an application on his cellular phone or PC Tom can give different persons access to his context information. The access is based on the person's identity, his own situation and the type of information that is requested. The system checks if Tom is in a situation where he has allowed this particular person to receive information about him, and if the request matches the type of information which should be shared in this situation.

1.3 Problem statement

Today's web technology already makes it quite easy to collect and process personal information of individuals. Continuing progress in the development of electronic devices might make the vision of a pervasive computing environment a reality. In addition to the use of Internet, a pervasive system introduces collection of context information through ubiquitous sensing and invisible embedded computing devices. Today a user of the Internet has some degree of control of where and when he is giving out personal information, while in a pervasive system data collection might happen without the user's approval, sometimes even without his knowledge. This raises significant questions about how to handle privacy in such environments.

The following questions and problem areas will be investigated throughout this report:

1. **Which privacy issues should be handled in a pervasive computing environment?**
2. **How can access to a user's context information be controlled, in a flexible and differentiable manner?**
3. **Design and implementation of a privacy enforcement mechanism in a given context aware system.**

The implementation will demonstrate a system which enforces privacy by restricting access to context information based on different parameters. The design principles will be in focus

while other aspects such as quality of service, standardisation and performance will be left to future work. Information security is also considered as out of scope for this assignment and will not be dealt with to any extent.

The focus will be on privacy concerns related to context dissemination. The system will perform access control to assure that only authorized entities receive a user's context information. It is assumed that the context management system is authorized to collect context information from the user and to store this information according to a set of predefined rules.

The context management system is the mediation system between context providers and context consumers. The administrator of the context management system is a trusted party and engaged in a business relationship with the user and other service providers. How privacy is secured in these relationships is regarded as out of scope for this assignment. Based on these assumptions the implementation aims to demonstrate how privacy can be enforced from the moment context information has been collected by the context management system and until it is communicated to other entities. The object is to prevent a third party from getting access to personal information without permission from the user.

It is further assumed that the technology needed to collect and communicate context information is implemented. The context management system is assumed to be implemented as described in Chapter 5. The management of context information is handled by this system. The handling of privacy enforcement is the task in this thesis. The design of the privacy enforcement system is further development of the project assignment which was carried out as part of the master degree [15].

1.4 Report organization

Chapter 2 introduce important terms and definitions. Section 2.1 describes pervasive environments which are the type of environments in which a context aware system will be applied. The rest of the chapter introduces the terms context and context awareness. There exist various definitions of these terms. The most important definitions and those most relevant for this assignment are mentioned. Principles for organizing context information and architecture will also be briefly looked into.

Chapter 3 looks into the concept of privacy. The different definitions and legislative principles are outlined. In order to identify the actual needs of the user the importance of privacy is discussed. User concerns related to new technological features introduced by pervasive computing environments are looked into. In Section 3.5 three different strategies on how to handle privacy is presented.

Chapter 4 presents different technologies to handle privacy enforcement. In Section 4.1 design principles and important tasks to consider when designing a privacy enforcement system are discussed. Section 4.2 and 4.3 presents work done in the area of privacy handling. This work is further evaluated and a new privacy handling system is presented in Section 4.4.

Chapter 5 presents the technology used to collect and manage context information. The context management system which is the mediation system between context providers and context consumers is described.

Chapter 6 presents the design and implementation. In Section 6.1 the method followed during the work on the design and implementation, RUP (the Rational Unified Process), is

presented. The different phases of the work process and the activities in each phase are identified. Section 6.2 presents the overall descriptions and requirements. In Section 6.3 the Use Cases are found and in Section 6.4 the design is presented with an overview, Message Sequence Charts and Process graphs. Section 6.4.6 presents part of the design which is not implemented. The details concerning the implementation are presented in Section 6.5 and the testing is documented in Section 6.6.

Chapter 7 covers the discussion of this report. In Section 7.1 the achievements resulting from the design and implementation are discussed. In Section 7.2 to what the degree the design principles outlined in Chapter 4 is fulfilled is looked into. In Section 7.3 the value added to the use of a context management system through introduction of the proposed privacy handling mechanism is discussed. In Section 7.4 the relevance and practical use of the system are discussed related to three different scenarios.

Chapter 8 presents the conclusion of this master thesis and future work.

2. Background

2.1 Pervasive computing environments

"Environments saturated with computing and communication capability, yet gracefully integrated with human users" [33]

In the literature you often find the terms ubiquitous and pervasive computing used interchangeably. Mark Weiser [39] first introduced the concept of ubiquitous computing in his research (1988-1994). His vision was to make computers available wherever you went, but at the same time make them invisible to the user. The term pervasive computing is used about the same type of environments. The Pervasive Computing conference in 2001 [30] defined pervasive computing as follows:

Pervasive Computing is a term for the strongly emerging trend toward:

- *Numerous, casually accessible, often invisible computing devices,*
- *frequently mobile or imbedded in the environment and*
- *connected to an increasingly ubiquitous network structure*

Pervasive computing is essentially about interacting with a smart environment where the technology is hidden from the user. This will radically change today's situation with mainly "desktop computing environments" [3]. In a pervasive environment you can perform your computation tasks on the run instead of sitting in your office with your personal computer all day long. Highly dynamic environments and less need for user attention are two of the main objects. This places new demands on applications which introduce the need for context aware computing [18].

2.2 Context

In the literature we find several approaches on how to define context and context awareness. According to Dey [13] context is "all about the whole situation relevant to an application and its set of users". Dey 's [13] definition of context is:

"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application including the user and applications themselves."

Context aware computing was first defined by Schilit and Theimer [34] as "software that adapts according to its location of use, the collection of nearby people and objects, as well as changes to those objects over time". Dey [13] takes this further and states as follows:

"A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task."

2.2.1 Classification

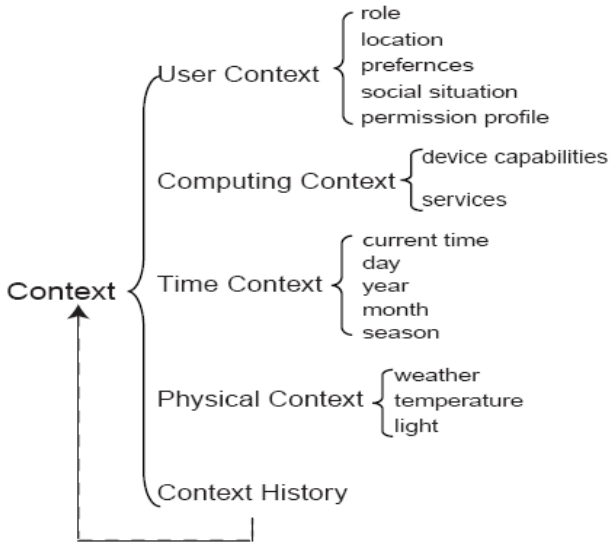


Figure 2-1: Classification of context types [28]

Mostéfaoui et al. [28] present the classification of context which is presented in Figure 2-1. Context history is recorded context over time. Context history is useful in order to discover relevant services and present these to the user. Another term used to describe user context is presence. Presence information refers to the state of the user such as availability, reach ability and other information set by the user [29]. A user's presence information can be derived from available context information such as location or the use of a service (e.g. talks on the phone).

2.2.2 Characteristics

In their work to develop a uniform context model, Henricksen et al. [18] characterize context information as follows:

- **Temporal characteristics:**
    - *Static information:* Information in a pervasive context system that is invariant, such as a person's date of birth.
    - *Dynamic information:* Information that changes over time, such as a person's location or activity.

  Static information is obtained only once or a few times and might be collected directly from the user. Dynamic information on the other hand is frequently changing and should be obtained indirectly through sensors or by other means.
- **Context information is imperfect:** Context information might be incorrect, inconsistent or incomplete due to several reasons. The dynamic nature of pervasive computing environments causes a rapid is collected by sensors or produced by derivation algorithms and users might be incomplete or faulty.

- **Context information has many alternative representations:** Context information is often derived from sensors. The representation of sensor output compared to what is a useful level of input information for an application might differ significantly. This gap requires various kinds of processing of context information.
- **Context information is highly interrelated:** Relationships between people, devices and communication channels can be evident like ownership, or less evident like from where information is derived. A relationship where the characteristics of the derived information (e.g. persistence and quality) are intimately linked to the properties of the information it is derived from, is called a dependency.

2.3 Architecture – Collection of context data

Implementations of context aware systems can be realised in several different ways. Which approach to use depends on different system requirements and conditions, such as location of sensors (local or remote), the amount of possible users and available resources. Another important aspect to consider is how context information is acquired. The choice of method gives a predefinition of the architectural style of the system. These three approaches to context-data acquisitions methods are presented by Chen [7]:

- **Direct sensor access:** Devices with sensors locally built in often use this approach. There are no layers to gain and process data. The client software gathers information directly from the built in sensor.
- **Middleware infrastructure:** This approach introduces a layered architecture to context-aware systems with the intention to hide low-level sensing details.
- **Context server:** With this distributed approach multiple client access to remote services is permitted. The gathering of sensor data is moved to a context server. This relieves clients of resource intensive operations and gives the possibility to reuse sensors.

To improve extensibility and reusability of systems it is necessary to separate detecting context and using context. Ailisto et al. [2] suggest a five-layered conceptual architecture as depicted in Figure 2-2. The architecture includes layers for detecting (sensor and raw data retrieval) and using context (storage management and application) [4].

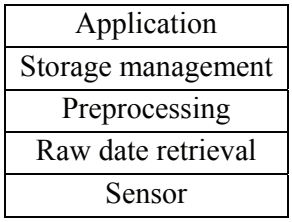


Figure 2-2: Layered conceptual framework for context-aware systems [2]

The different layers of Figure 2-2:

- **Sensor**: This layer consists of a collection of different sensors. By sensors it is not only meant sensing hardware, but also other sources that provide context information such as users and devices/device managers. Sensors can be classified into three groups:
    - *Physical sensors* are hardware sensors such as RFID, GPS and GSM, a microphone or a camera.
    - *Virtual sensors* are typically software applications and services such as electronic calendars or e-mail.
    - *Logical sensors* combine information from physical and virtual sensors with additional information from for instance a database to derive new information.
- **Retrieval of raw context data**: This second layer makes use of appropriate drivers for physical sensors and APIs for virtual and logical sensors. This is where the query functionality is implemented.
- **Preprocessing**: This layer is responsible for reasoning and interpreting contextual information.
- **Storage and management**: On this layer the gathered data is organized and offered to the client via a public interface. Client access can be offered in two ways:
    - *Synchronous:* The client is polling the server for changes via remote method calls. The client sends a message requesting information and waits until it receives an answer from the server.
    - *Asynchronous:* The client subscribes to specific events and gets a notification when the event occurs.
- **Application:** This layer implements the actual reaction on different events and context instances.

2.4 Context Aware Applications

In order to give the reader a better understanding of practical use of context information and context aware systems an example of context aware applications will be presented. Dey identifies these three categories of features that a context-aware application can support [13]:

- *Presentation* of information and services to a user
- *Automatic execution* of a service for a user
- *Tagging* of context to information to support later retrieval

An example of a context aware application is a service that detects when a user is moving from a low capability zone to a zone with higher capabilities, (i.e. the access to services/devices improves related to the user's needs). Such a service is described in Egil Østhus' master thesis [42]. The scenario depicted in his report describes a service which makes it possible to discover when the available services are better suited for the user's needs at the moment, than the one he is currently using. Such a system exploits the knowledge of the user's context to derive that another available service will be better suited for the given task. This is an example of how context information can be used by an application to improve service.

3. Privacy

"Privacy is the right to be left alone"

-Judge Brandeis [38]-

In this chapter the concept of privacy will be discussed. The definitions, legislative principles and user concerns related to new technological features introduced by pervasive computing environments will be looked into.

3.1 Definitions

Privacy is an abstract notion. People often think about privacy as a right, but it might be more correct to describe privacy as a value or an interest. It is hard to argue for one absolute standard for what privacy protection should include, as it might vary in different contexts such as geography (e.g. at home or at work), expectations and manners [24]. In the literature you find several different definitions on privacy. The most prominent are probably Judge Brandeis' definition which is quoted above and Alan Westin's definition.

"Privacy is the claim of individuals, groups or institutions to determine for themselves when, how and to what extent that information about them is communicated to others."

-Alan Westin [40]-

These two definitions focus on two different aspects of privacy. While the first concerns the individual's right to control a personal space, the latter is concerned about the individual's right to control the flow of personal information [17]. In line with this thinking Roger Clarke [20] divides the definition of privacy into two parts, giving one part the name information privacy.

***"Privacy** is the interest that individuals have in sustaining a 'personal space', free from interference by other people and organisations."*

***"Information Privacy** is the interest an individual has in controlling, or at least significantly influencing, the handling of data about themselves."*

The term privacy is sometimes misused to refer to the security of data against various risks, like data being accessed or modified by unauthorized persons or the security of data during transmission. Privacy and security are two different terms and should be kept separately. In the context of data privacy, privacy refers to a value, while the term security, in the context of data security, refers to a methodology or a technology. Security can both serve and hinder privacy. An example is when a company decides to filter all outgoing e-mails to hinder sensitive information to leak out. This is a security mechanism at the same time as it threatens privacy because it implies reading all employees' e-mails [9] [31].

3.2 Privacy History and Legislation

The development in information technology that might interfere with privacy has often been the motive power behind development of legislative protection of privacy. Privacy has been of interest since the beginning of the 1800th century. Then it mostly concerned the written press and photography. The modern privacy debate, concerning information privacy, started with the introduction of automatic data processing in the 1960s. This discovery resulted in the making of personal records and the possibility to store and process information of all individuals [5] [23].

The US Privacy Act of 1974 was one of the most influential early pieces of privacy legislation. The principles of the act are based on the notion of "fair information practice" which in turn is based on the work of Columbia University political economist Alan Westin [23]. The principles are as follows:

- ***Openness and transparency:*** *There should be no secret record keeping. This includes both the publication of the existence of such collections, as well as their contents.*
- ***Individual participation:*** *The subject of a record should be able to see and correct the record.*
- ***Collection limitation:*** *Data collection should be proportional and not excessive compared to the purpose of the collection.*
- ***Data quality:*** *Data should be relevant to the purposes for which they are collected and should be kept up to date.*
- ***Use limitation:*** *Data should only be used for their specific purpose by authorized personnel.*
- ***Reasonable security:*** *Adequate security safeguards should be put in place, according to the sensitivity of the data collected.*
- ***Accountability:*** *Record keepers must be accountable for compliance with the other principles.*

The Norwegian legislation is based on two fundamental principles; the notion of personal information and personal records. Personal information is any information that can be connected to an individual. A personal record is a collection of personal information organized such that it is easy to retrieve information about one particular person [5]. The "Personal Data Act" of 14 April 2000 No.31 relating to the processing of personal data states [11]:

Purpose of the Act

The purpose of this Act is to protect natural persons from violation of their right to privacy through the processing of personal data. The Act shall help to ensure that personal data are processed in accordance with fundamental respect for the right to privacy, including the need to protect personal integrity and private life and ensure that personal data are of adequate quality.

Definitions

For the purposes of this Act, the following definitions shall apply:

1) personal data: any information and assessments that may be linked to a natural person,
2) processing of personal data: any use of personal data, such as collection, recording, alignment, storage and disclosure or a combination of such uses,

This Act shall apply to

a) processing of personal data wholly or partly by automatic means, and
b) other processing of personal data which form part of or are intended to form part of a personal data filing system.

3.3 The Importance of Privacy

In everyday life people give out personal information about themselves all the time. Each time you use a credit card, visit the doctor's office or other institutions or simply turn on you mobile phone you give out pieces of information about yourself. All this information is stored in registers and might even be further distributed.

In his book "Privacy in the Danger Zone" [5] Jon Bing gives an illustration which shows what might happen if all registered information about a person was processed in order to find out as much as possible about this person's doings. The experiment was carried out in the 1970s by a manager at Honeywell-Bull and was meant as a horror vision. Today retrieving and comparing information from these various registers had not been a problem at all without privacy legislation.

DATE	HAPPENING	COMMENT
260884	*Mr. Jensen buys a fur coat in size 8 at Fur shop, Oslo*	*Mrs. Jensen is a size 10*
191079	*Jensen is number 6. in a bullfight in Las Palmas*	*The National Insurance Administration's archives shows that Jensen has been reported sick the 15.-22. October due to a back injury*

Figure 3-1: The 1970s' "Horror vision" [5]

The example given above is meant to be humoristic, but if we think about which authorities that would benefit from making comparisons like this it might not seem that amusing. In the same book Bing raises the questions; what if the banks were allowed to analyse a potential customers consumptions patterns and the insurance companies were allowed to check the health registers? This might give a situation where decisions concerning a person's rights were solely based in information contained in registers. The question is; does this information give a correct picture of the whole truth? The example also illustrates how seemingly insignificant pieces of information can divulge a lot of information when put together.

3.3.1 User perspectives

Given the illustration above it seems quite evident that some kind of privacy protection is essential. However, as mentioned in Section 3.1 privacy is not a technology, but a value or an interest. To maintain the individual's privacy often results in a trade-off between the interest of the individual and the interests of the majority. One example is the use of surveillance cameras to reduce crime. A survey made on Norwegians' attitude towards and knowledge of privacy shows an increasing acceptance among people when it comes to use of surveillance equipment such as cameras [32]. They also believe that personal information they have to give out is treated in a satisfactory way. The general thinking seems to be that only those who have something to hide need privacy protection. People also tend to think that nobody is really interested in what they are doing anyway.

"We now stand before a theoretical possibility to exterminate crime in the society, in the sense that all people are under constant surveillance. But this is a society I am sure we will have great hesitations about living in, because that would probably not be a good society."

-Lars Sponheim, politician [12] –

The problem is where to draw the line for when the interest of a better or safer society should weigh out the interest of privacy for the individual. As the director of the Norwegian Datatilsynet discuss in the editorial of this year's privacy report [12], the privacy question is often a question about the majority having the opportunity to dictate the few. Jon Bing talks about "the balance of power" [5]. He argues that access to personal information about another person gives the possibility to exert power over this person. Such relations of power might typically be between the state and citizens, between employer and employee or the individual and the neighbourhood. When decisions are made it might be essential for the individual to have control over what information about themselves that are taken into account.

Sun CEO Scott McNealy takes this to the other extreme when he says: "You already have zero-privacy anyway, get over it" [23]. Some argue that too much privacy legislation does more harm than good. If everyone can find out everything about everyone much of the information will lose its interest. Langheinrich [23] summarizes the opinion of several critics of to much privacy legislation into these four points of privacy measures which actually are necessary.

- ***Feasibility****: what can technology achieve or better prevent? Laws and legislation require enforceability. Privacy violations have to be traceable in order to make the violators accountable.*
- ***Convenience****: the advantages of free flow of information outweigh the personal risks in most cases. Only highly sensitive information, like sexual orientation, religion, etc might be worth protecting. Semi-public information like shopping habits, preferences, contact information, even health information, might better be publicly known so that I can enjoy the best service and protection possible.*
- ***Communitarian****: personal privacy needs to be curbed for the greater good of society. Democratic societies may choose to appoint trusted entities to oversee certain private matters in order to improve life for the majority.*

- ***Egalitarian****: if everybody has access to the same information, it ceases to be a weapon in the hands of a few well-informed. Only when the watchers are being watched, all information they hold about me is equally worth the information I hold about them. Eventually, new forms of social interaction will evolve that are built upon these symmetrical information assets.*

3.4 Features with pervasive environments that challenges privacy

Pervasive computing environments have certain properties that make them different from other computing environments with respect to privacy [23]. These properties can be summarized as follows:

- **Ubiquity/pervasiveness:** The computing environment surrounds us. Consequently design decisions made for such system will affect large parts of our daily lives.
- **Invisibility:** The computers are invisible to the users. This will make it hard to know when we actually are interacting with a computing or communication device.
- **Sensing:** Increasing processing power and shrinking of computing technology make it possible to make sensors that accurately perceive certain aspects of the environment. Temperature, light and noise have been captured for some time, but next generation sensors will be able to make high quality video and audio records with cameras or microphones smaller than buttons.
- **Memory amplification:** Development of memory prosthesis and amplifiers make it possible to transfer what-ever sensors capture to devices with limitless storage capacity.

3.4.1 Crossing new borders

MIT professor emeritus Gary T. Marx argues that: “central to our acceptance or sense of outrage with respect to surveillance, regardless of how it is done, are the implications for crossing personal borders”. He further argues that violations of personal borders involve one or more of the following four conditions [27].

- **Natural borders**: Physical limitations of observations, such as walls and doors.
- **Social borders**: Expectations of confidentiality due to a person’s role. This can be expectations tied to professions such as doctors or lawyers. It can also be the expectance that a friend or family member will not betray your trust.
- **Spatial or temporal borders**: People expect that parts of their lives which are separate from each other, either in time or social space, remain separate. You do not expect a wild youth to interfere with your life at 50 or you may not expect to meet your friends and colleagues in the same bar.
- **Borders due to ephemeral or transitory effects**: You do not expect an unreflecting utterance or action to be remembered forever or pop up later.

Pervasive computing systems will give far greater possibilities for crossing these borders. Most obvious is maybe the two latter ones. With various types of recording equipment much more of what we say or do will be recorded and potentially stored forever. The same equipment is also becoming smaller and smaller. This increases the risk of crossing physicals borders in ways that until now have not been possible, at least not for normal persons, without access to expensive listening devices. Social borders might suffer due to the same reasons. Recording equipment can accidentally be left without the owner's presence and conversations which were not intended for the owner's ears might be recorded. In this way a person might indirectly listen in on personal conversations when they later pick up the recordable device.

3.4.2 Context aware systems

When considering the effects of a pervasive environment discussed above the most obvious solution might be to put strict restrictions on collection of personal information. However, offering services in a pervasive environment often require collection and storage of context data. This introduces problems like when and where it is allowed to collect data, for how long can the data be stored, who has access to the collected information and for what purposes can the data be processed and used?

In a pervasive environment there is no single, clearly assigned user device per user, and the mobility is high. A user is moving around changing devices and switching between service providers. Co-operation between service providers will also occur. A user in such a system has to rely on many providers to be able to be provided with a magnitude of devices, networks and services. In order to exploit the full potential of context aware systems, context information has to be distributed to other entities than the context owner. Situations where dissemination of context information occurs:

- When context information is necessary to provide a service.
- When other users request or subscribe to information about a user.

To avoid a situation where someone is constantly watching you, the user has to be able to regulate the degree of access granted to other users of the system. Different users' privacy requirements are heterogeneous of nature due to difference in information sensitivity and user preferences. In addition user preferences can be context-dependent, i.e. sometimes you want people to know where you are and sometimes not.

Summarizing the points above gives this list of main issues that have to be considered in a context-aware system which gives an answer to the question raised in the problem statement in Section 1.3; which privacy issues should be handled in a pervasive computing environment?

- Permission to collect personal data
- Permission to store personal data
- Processing of personal data – for which purposes is this allowed
- Access to personal data / distribution of personal data

3.5 Privacy Strategies

There are three general approaches for handling privacy issues [17]. These are law, self-regulation and technical solutions. The principal of legislation was discussed in Section 3.2. Traditionally European countries have relied more on this approach to handling privacy than many non-European countries. The USA for instance has traditionally taken the more liberal approach of self-regulation. It is often argued that since users are concerned about privacy the industry will come up with solutions that are satisfactory. This has not proved to be very successful and a shift in strategy towards a legislative approach has been suggested [17].

In the rest of this report the focus will be on the technological approach. One of the fundamental principals in today's democratic society is to give people the possibility to respect other people's safety, property or privacy [26]. This is supported by corresponding norms, legal deterrence and law enforcement to create a reasonable expectation that people follow those rules. The idea is to let people follow the rules and then punish the wrongdoers. Applying this principal in order to secure information privacy implies offering technical solutions that takes privacy concerns into consideration. The presumption for this to work is a possibility to capture wrongdoers and the existence of a way to punish them.

4. Privacy system technology

In this chapter the privacy challenges introduced by pervasive computing environments will be further looked into. The privacy issues most important to handle will be identified and the technological solutions to solve these potential problems, will be discussed.

4.1 Design Principles

Marc Langheinrich [23] has described a set of design principles for privacy enforcement in a pervasive environment. He points out that by following these principals he is not trying to achieve total security or total privacy, but to prevent unwanted accidents such as "data spills of highly personal information that people have never asked for on their doorstep". The goal is to allow people who want to respect other people's privacy to be able to "behave in such a way, so that they will eventually be able to build a long lasting relationship based on mutual trust and respect". This complies with the democracy principle mentioned in Section 3.5. Another goal is to achieve a balance between convenience and control for the user when interacting with entities in a pervasive environment. The principles are as follows:

Notice: One fundamental principal in any data collection system is the Principal of Openness, or also called Notice. In most legal systems today you can not collect any data information without the subject's knowledge. In a pervasive environment it is often difficult for the data subject to realize that data collection is actually taking place. It will therefore be necessary with a mechanism to declare collection practices (i.e. privacy policies) in addition to efficient ways to communicate these to the user (i.e. policy announcement).

Choice and consent: It is not enough simply to announce and declare data collection. The user must be able to choose not to accept. This means that the data collector must receive explicit consent from the data subject. The most common way of giving explicit consent is still a written contract. However, in the world of electronics this is a bit more complicated. To use a digital signature is possible, but as it is not only a question of authenticity but also explicitness. The use of a digital signature to sign a statement does not necessarily give explicitness. It might very well be the user's personal software that signs the request without the user's actual knowledge. An alternative is to use a querying mechanism that asks the user to press a button if consent is given. In a pervasive environment, however, this might be physically impossible and unusable. The present device might not have a tactile interface and the amount of querying that would have to be answered might be too high.

Another problem is the requirement of choice. The user must have the possibility to decline an offered service without the service being shut down for everyone. In order to give users a true choice a selection mechanism has to be provided which gives the users other choices than "take it or leave it". An example is a tracking service in a building. Declining this service should not result in the use having to stay outside the building.

Anonymity and pseudonymity: An alternative to personal data collection is to operate with pseudonyms or anonymity. Anonymity can be defined as "the state of being not identifiable within a set of subjects." The larger the set of subjects, the stronger is the anonymity. Pseudonymity is an alternative to anonymity where a subject can be associated with the same ID several times, but the real identity is not revealed. This gives the possibility to personalize a service by using the same ID several times, but at same time having the possibility to get out

of it when desired by changing the ID. The use of anonymity or pseudonyms poses no threat on privacy since collection and use of such data can not be traced back to one individual. The problems caused by the requirement of explicit choice can thus be avoided. However, there might be situations where you do not want to stay anonymous. . The problem that arises as a consequence of this approach is that you do not always want to operate anonymously. There has to be a reasonable balance between total anonymity and total openness.

Proximity and locality: The feasibility in implementing an efficient and reliable system with the desired privacy aspects might prove difficult in a pervasive computing environment. One possibility to preserve some desirable state of privacy protection while facing this technological reality, are the principles of proximity and locality. By proximity it is meant that a data collection device only collects data while its owner is present. By locality it is meant that the collected information is tied to the places at which it is collected. If a system supports mechanisms to encode and use locality information for collected data, access restriction can be enforced based on the location of the person wanting to use the data.

Adequate security: The idea is; once security is solved, i.e. once authenticity and trusted communication is achieved, privacy will be a by-product that follows inevitably from a secure environment. However, ubiquitous devices will introduce a whole new set of constraints, mainly in the area of power consumption and communication protocols. The size of the devices will introduce constraints on the available power and the mobility of the devices limits the time to go through with an orderly security protocol. One way to solve complexity of security mechanisms is only to employ robust security in situation where highly sensitive data is being transferred.

Access and recourse: For a system to be trustworthy a set of regulations to separate acceptable from unacceptable behavior is required. Mechanisms to detect violations and enforce penalties according to the rules are also necessary. This is more in the realm of legal practice, but technology can help implementing specific legal requirements such as use limitation, access or repudiation.

In the fair information practice the principles of Collection and Use Limitation is set forth. These principles can further simplify access requirements by requiring data collectors to:

- Only collect data for a well-defined purpose (no “in-advance” storage)
- Only collect data relevant for the purpose. Personal data should only be disclosed if it is really necessary for the functionality of the system. The number of attributes disclosed should be limited to those absolutely necessary.
- Only keep data as long as it is necessary for the purpose

Combined with anonymization and pseudonymization these principals save time and effort in the process of properly collect, protect and manage larger amounts of sensitive personal information. A system should also provide a way for users to access their personal information in a simple way through standardized interfaces and the users should be informed about the usage of their data once it is stored.

4.2 Work done in the area of privacy protection

Since there are no standard for context-aware systems there are obviously no developed standard for enforcing privacy in context-aware systems either. In this section some of the available solutions and approaches that have been suggested will be looked into.

4.2.1 Privacy Enhancing Technology (PET)

The use of technical solutions to solve potential privacy problems is often labelled Privacy Enhancing Technologies (PET) [17]. PET is divided into four categories:

- Policy tools, helping users understand privacy policies
- Anonymisers, allowing users to be anonymous
- Encryption tools, encrypting content
- Filters, e.g. cookie managers

Originally PETS were conceived as ways to design or redesign the infrastructure to preserve privacy and limit surveillance and data collection.

4.2.2 Privacy policies and policy announcements

As discussed previously a user should be noticed when ever information about them is collected. To achieve this some kind of announcements system is necessary. An analogy mentioned in [23] is radio traffic announcements. A constant radio broadcast would however rapidly drain battery of small mobile devices, while using RFID tags to passively announce such information in rooms or buildings would be perfectly acceptable. RFID tags would not use any battery at all and to solve the problem of storage size the actually information can be outsourced to a web site while the tag only announces the URI.

Another solution to the announcement problem can be the use of policies. This is a mechanism that enables the user to gain more control over the use of their personal information. Policy is a technique in which a specification of high-level rules control and adjust the low-level system behaviour. The use of policies is common in computing systems that feature security and privacy protection. In a policy-based system declarative policy languages are used to define the policies. Languages that can be used are meta-languages, such as XML or RDF, or semantic web languages, such as OWL. The advantage of using these languages is that they are distinct from the actual system programming languages. This makes the defined policies easier to read and edit for humans. In addition, by separating the logic from the control of the system implementation, make systems using policies for privacy protection more flexible and adaptable than other non-policy-based systems[6].

The Platform for Privacy Preferences Project (P3P) [41]: The Platform for Privacy Preferences (P3P) developed by the World Wide Web Consortium (W3C) is an initiative to solve the problem of announcement to secure privacy on the Internet. Web site operators can use the P3P language to describe the sites privacy practice to its visitors. At its most basic level this is a set of multiple choice questions which covering the different aspects of the web

site's privacy policies. This is represented in a machine readable format. The users configure their browsers or other software tools to check if the Web site's policy is in accordance with their own preferences. Part of the idea is that once Web sites and Internet users can better communicate about privacy preferences, consumers will be able to make better judgments about which Web sites respect their privacy concerns.

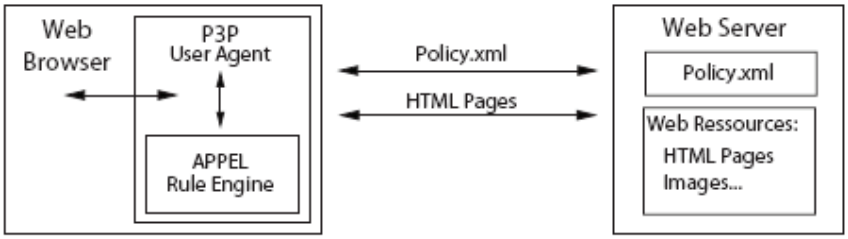


Figure 4-1: P3P example [24]

4.2.3 Anonymisers

The anonymizing tools available for the web today such as www.anonymizer.com, give the Internet user the possibility to hide their IP-address when they visit a web site. The technology behind such tools is already well established for use on the Internet, but the feasibility of such methods in a pervasive environment might be limited. Communication in pervasive environments is much more dynamic. The long chains of communication that are used to hide the identity of a user might not last long enough because devices constantly enter and leave the same scene. In direct communication the real identity is disclosed because the MAC–address, the fixed hardware address, is used in the wireless protocols.

The Daidalos project [10] addresses the problem of revealing personal information that can be linked to your real identity when you use a service. Their approach is to partition sensitive data into smaller sets that separately do not reveal that much information. In this way a user can act under different pseudonyms which are linked to only a limited set of attributes. Only one trusted party has the possibility to link the pseudonyms to a real identity. A similar approach is used in GSM to give users a virtual id for location information. When a user is outside his or her home location domain it is not possible to track the user's location based on information obtained by the visiting domain.

4.2.4 Transparency and Trust tools

Transparency and trust tools are meant to increase consumer trust in transactions or data exchange by providing additional background information about the transfer, its conditions, and the parties involved.

TRUSTe: An example of a trust tool is TRUSTe [37]. TRUSTe certify and monitor web site privacy and e-mail policies, monitor practices and resolve consumer privacy problems. Companies that adhere to TRUSTe's strict privacy principles can display the TRUSTe® Web Privacy Seal on their web site. The principles include:

- Creating a **privacy policy** to be reviewed by TRUSTe
- Posting **notice and disclosure** of collection and use practices of personally identifiable information
- Giving users **choice and consent** over how their information is used and shared

4.2.5 Data tagging – proximity, locality

An example where access to data elements are restricted by locality is described by Jiang and Landay in their article Modelling Privacy Control in Context Aware Systems [22]. They base their theoretical model for privacy control on an abstraction of information spaces and unified privacy tagging. An information space is a way to organize information, resources, and services around important privacy relevant contextual factors in context aware systems. An information space can typically be all information and resources within a restricted area (i.e. an office) or information concerning a group of people. Each object in an information space is associated with a privacy tag. A document in an office for instance has a tag that says which information space it belongs to, what kind of operations that are permitted to be performed on it and a privacy property that defines where it can be used.

4.3 Examples of privacy systems

In this section some examples of more extensive solutions on how to handle privacy in pervasive computing environments will be described.

4.3.1 Identity Management

To improve a person's privacy in a pervasive computing environment Jendricke et al. [21] propose a situation-based control over published data and offered services. With what they call Identity Management a user's personal device will present different subsets of a user's identity depending on the perceived context. This system addresses the privacy issue of restricting access to personal data. The need of such a system is based on the situation that arises in a pervasive computing environment where all devices are "smart" and communicate with each other all the time. To prevent the situation where the user's personal identity and whereabouts is constantly revealed it is necessary with some kind of control mechanism that checks if the receiving part is actually entitled to receive this information.

In practice the identity manager allows the user to determine the personal data that should be offered in a situation. The situation is given by the user's context. The set of data that is offered represents the user in the given situation and is the user's (partial) identity. The situation is specified by different URLs. The situation recognized by comparing the URL to pre-defined settings or rules. What identity that should be used in a given situation is derived from the user's previous action such as earlier choices of a particular identity. If the situation is new or the different situations can not be distinguished from each other the system might not be able to derive which situation to use. Some rules to determine what to do in these situations must therefore exist. One of the following approaches has to be chosen:

- **Completely user-controlled determination of the situation:** The situation is changed manually by the user.
- **Semi-automated determination of the situation:** The user is presented a limited set of different situations to choose from. The situations which are presented are based on the conceived context.
- **Fully automatic determination of the situation:** Based on the context of the user the identity manager determines the appropriate situation.

P3P can be used as an add-on for identity management. Each identity might contain its own privacy preferences.

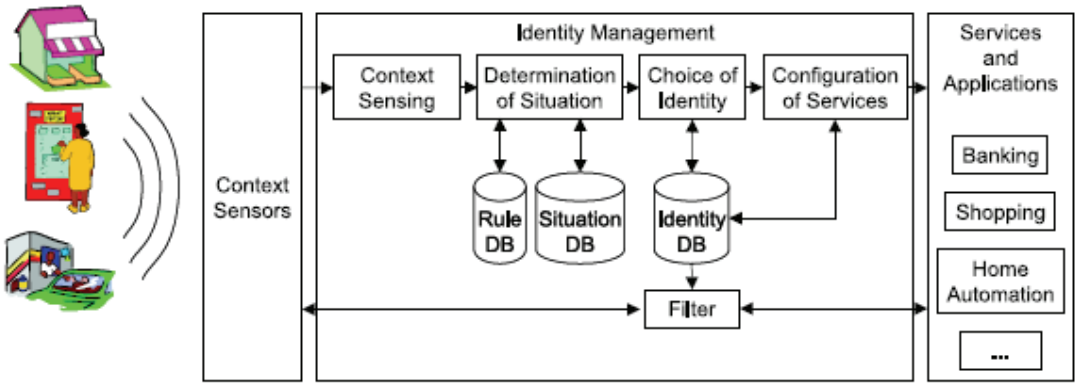


Figure 4-2: Identity management [21]

With identity management Jendricke et al. introduce the terms “situation” and “(partial) identity”. The problems they want to address with the Identity Management system is enclosure of personal details when using an IT-system and who gets to know personal data stored on a mobile device. They expect more acceptance of pervasive computing by introducing this system. The privacy of the user is respected and it relieves the burden of the user by giving the user a useable tool for protecting his or her privacy.

In their article Jendricke et al. claim to fulfil the principles of Langheinrich which is outlined previously in this chapter in the following way:

- **Notice**: The user device shall demand the attention of the user to choose an identity unless the situation is familiar and a previously chosen identity can be assigned.
- **Choice and consent**: The main task of the identity manager is to give the user control over which personal data he or she gives away. The system authenticates itself when the user explicitly allows it to do so or when a pre-configured situation arises. In this way the user has full control over the data that is revealed to potential communication partners.
- **Anonymity and Pseudonymity**: By default the identity manager reveals no identity. Pervasive environments are highly mobile which introduces the need for a new anonymity mechanism. When authentication is needed the user can configure his or

her device to reveal authentication data only in well-defined contexts and to well-defined communication partners.

- **Proximity and locality**: This principle should be applied when neither of the points above can be fulfilled and cryptographic algorithms can not be used.
- **Security** – The identity manager supports cryptography when the pervasive device is capable of running cryptographic algorithms and key exchange protocols are available. The manager automatically uses an encrypted connection if there is one.
- **Access and recourse**: The principle of access can not be assured by this system because the mechanisms to handle this principle go beyond configuration of the user's hardware or software.
-

4.3.2 Policy tagging of data elements

In [26] Langheinrich suggests a privacy awareness system (PawS) for ubiquitous computing environments. The system allows data collectors to both announce and implement usage policies. The system also provides technical means to keep track of their personal information, i.e. where it is used, stored and eventually removed from the system.

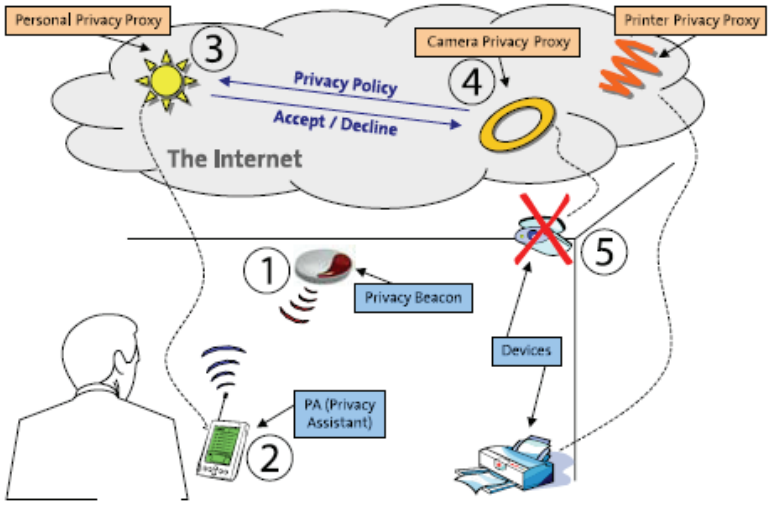


Figure 4-3: Policy tagging [26]

Langheinrich addresses the fact that though anonymization technologies and encryption schemes are sufficiently good to secure secret communication and masked identity, there might be situations where we want to reveal our real identity and whereabouts. The PawS system aims to strike a reasonable balance between total anonymity and total openness. The idea is to give people the possibility to respect other people's privacy as discussed in Section 3.5.

When designing the general architecture of this system Langheinrich followed his own design principles. However, he states that the principles of anonymity, pseudonymity and security are useful tools as supportive parts of the infrastructure, but should not be taken as isolated solutions to handle privacy. He then gets these four core concepts for the system:

- **Machine-readable privacy policies to provide choice and consent**: Using the P3P framework data collectors can write XML-documents to describe for example who is collecting information, what data is collected, for what purpose it is collected, and for whom it is being collected. The users describe their privacy preferences in a similar way and automating processing judges the acceptability of the process.
- **Policy announcement mechanisms to give notice**: P3P is a web technology which uses HTTP-headers and URI-locations on each web server to help users locate policies. In a pervasive environment an alternative mechanism is needed. There are two different ways of collecting data, the user can actively seek the service or the services can work continuously in the background (e.g. audio or video tracking). In the first case the P3P policy is embedded into the service discovery protocol, while in the second case a privacy beacon must be used that constantly announces privacy policies of implicitly running data collections.
- **Privacy proxies to support access**: Privacy proxies handle privacy relevant interactions between data subjects and data collectors. Each ubiquitous computing environment has a service proxy and each user has a personal privacy proxy. The user's personal proxy handles all interaction between service proxies in order to exchange user data or query their "usage log".
- **Privacy aware databases for recourse**: When data has been collected from a user it is stored in a back-end database. In order to prevent accidental use of the data it is stored together with the individual privacy policy it was collected under. In this way the database can make sure that the data is being used as it was intended to and provide the user with a "usage log" of their personal data.

4.3.3 CoBrA – Context Broker Architecture

Harry Chen and Tim Finin [6] have developed a framework for an agent based pervasive computing environment. This is a context-aware computing infrastructure called Context Broker Architecture (CoBrA). Their framework model is based on the presence of an intelligent context broker which handles context information input from devices and agents in the environment, and from other sources such as profile information on the web. The broker processes and reasons over this information to maintain a coherent model of the environment.

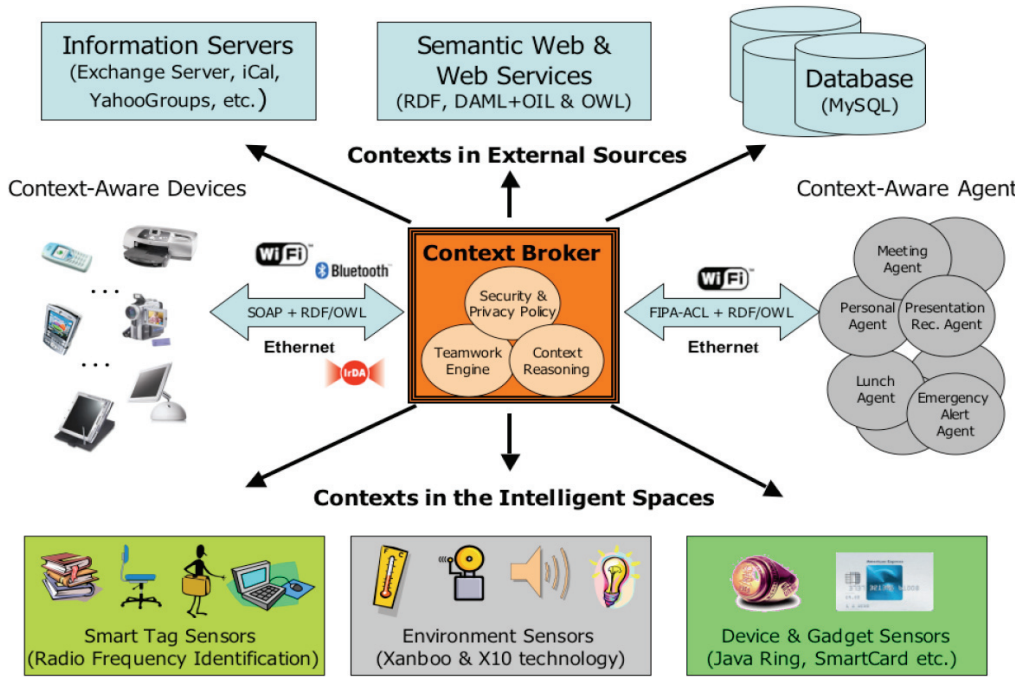


Figure 4-4: Model of the intelligent context broker [6]

In the CoBrA architecture the users are able to define a privacy policy to control the sharing of their context information. These user-defined policies are enforced by the broker agent. No agent is allowed to share user information without permission from the broker.

CoBrA has adopted the SOUPA[1] (Standard Ontology for Ubiquitous and Pervasive Applications) policy ontology to define policies to protect user privacy. An ontology defines an area of knowledge and gives a shared understanding of the specification of entities and their relationship with each other. SOUPA is a set of ontologies used to support knowledge representation and knowledge sharing in pervasive computing systems. The following design principle "*policies are rules that regulate the permission for computing entities to perform actions"* is the foundation base of the SOUPA policy ontology. The policies are defined by the human users to permit or forbid computing entities to perform different types of actions.

[1] More information about the SOUPA ontologies can be read at: http://pervasive.semanticweb.org/soupa-2004-06.html

An action can be to invoke computing procedures to access user information or to access services in the computing environment.

The privacy approach in this system is to use a policy predefined by the user to handle access control. The principle of notice is supported as the user defines a policy where only entities that are given access rights are allowed to receive information. In this way no other entities can collect data without the user's knowledge. Due to this solution the principle of choice and consent are supported as well. The policy gives the possibility to share some data and hide the rest. In this way anonymity and pseudonymity can be achieved.

4.3.4 E-wallet

Fabien L. Gandon and Norman M. Sadeh [16] introduce a Semantic Web architecture to reconcile privacy and context awareness. Their architecture aims at supporting automated discovery and access of personal resources in support of a variety of context-aware applications. This is done through the use of Semantic Web services that represent each source of contextual information (e.g. a calendar, location tracking functionality, collections of relevant user preferences, organizational databases). Each user has a directory, a semantic e-Wallet, which contains his or her contextual resources. This e-Wallet also enforces the user's privacy preferences.

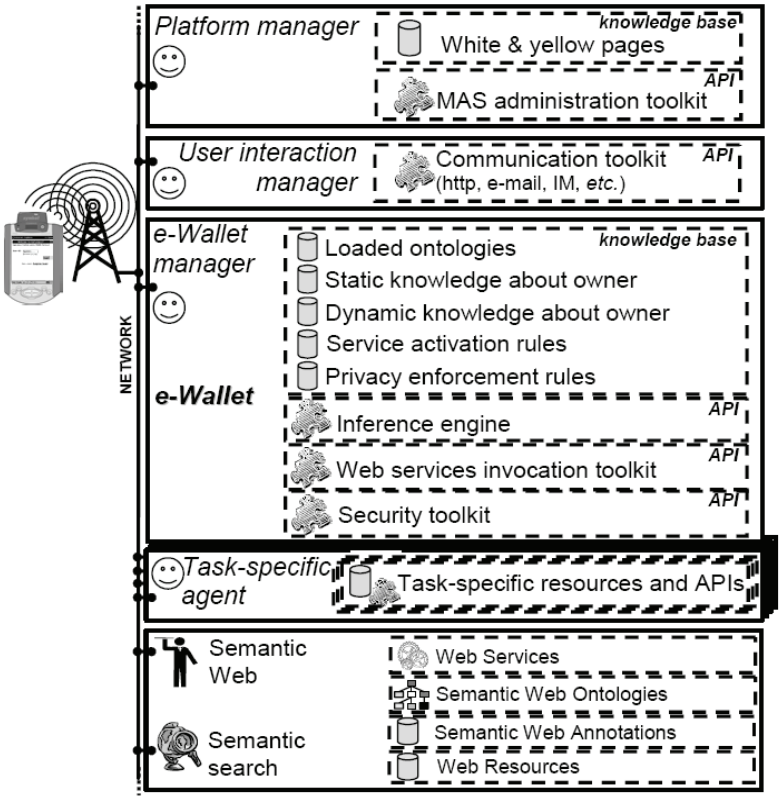


Figure 4-5: An overview of the Semantic Web environment [16]
(The smiley faces represent agents)

In the e-Wallet architecture privacy is enforced by giving the user control over access to personal information based on different conditions. This is done through privacy preferences which encapsulate knowledge about what information the user is willing to share with others in different situations. The preferences are represented in OWL. They are divided into two categories:

- **Access control rules** which express who has the right to see what information under what conditions. Related to the scenario in Section 1.2, Tom can say that his location information shall only be visible to his colleagues between 8am and 5pm.

- **Obfuscation rules** which are about giving out personal information with different levels of accuracy or inaccuracy. There are two types of obfuscation rules:
    - *Obfuscation by abstraction* is about abstracting away certain details about the user's current context (e.g. Tom gives out the information that he is travelling by train, but not the departure and destination of the journey).
    - *Obfuscation by falsification* is about situations where the user does not want to appear as though he is withholding information and is providing false information instead (e.g. Tom does not provide his business partners with the information that he is having a meeting with another associate, but he does not want to appear off-line either. His context information for this group is therefore at "at work, but busy in staff meeting").

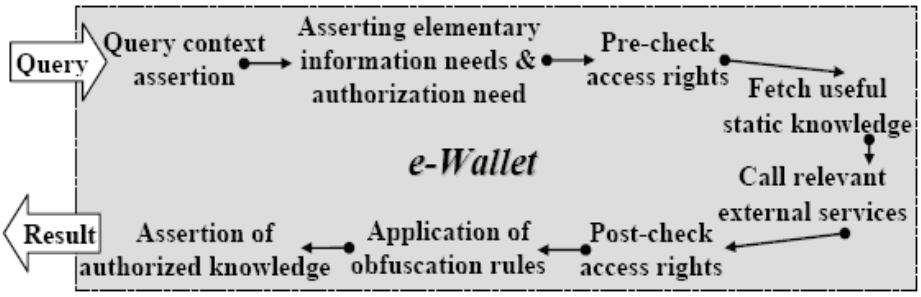


Figure 4-6: The main steps involved in processing a query submitted to an e-Wallet [16]

The main steps as illustrated in Figure 4-6:

1. Query context assertion – parts of the requesting user's context information such as name and role, is loaded in to the e-Wallet's inference engine. This is for later use when the query is being processed.
2. Assertion of the information contained in the query, that is what the user is asking for, and determination of authorizations needs of the user.
3. Pre-checking whether the requesting user is entitled to receive the information s/he is demanding. In the "Tom" scenario this could be to check if one of Tom's colleagues is allowed to make queries about his location.
4. In this step the e-Wallet is controlling the user's context data that would respond to the query. If the query could not be responded with only resources from the local knowledge base stored in the e-Wallet, it has to continue to the next step.
5. The e-Wallet is invoking external resources such as Web services to fetch the information which was not stored locally.

6. The query is now checked once more to make sure that Tom's current contextual situation is matching with the access rights given to the querying user. Tom's colleague is entitled to ask about Tom's location as long as Tom is still at work.
7. The next step is to check for obfuscation rules. Tom might only want to reveal which building he is in and not the specific room.
8. The answer is generated.

This system uses a similar privacy approach as the CoBrA system, where a predefined policy is used to handle access control. Thus the principle of notice and the principle of choice and consent are supported in this system as well, as the user also here defines a policy where only entities that are already given access rights are allowed to receive information. No other entities can collect data without the user's knowledge and approval.

4.4 Motivation and theoretical foundation

The systems described throughout this chapter show different solutions for privacy handling in context aware systems. In this section each of the approaches will be evaluated in order to give a foundation on how to handle privacy in the context management system which will be presented in Section 5.1.

4.4.1 Evaluation of existing systems

Identity Management [21]:
In the Identity management system the user's personal device presents different subsets of the user's identity, depending on the user's context, to a communication partner. In this way the user is given control over published data and offered services based on his or her situation. The privacy issue in focus is restriction of access to personal data. The need for such a service is based on the development of a pervasive computing environment where "smart" devices communicate all the time. This system is suggested as a mean to avoid a situation where the user constantly reveals his or her identity and whereabouts.

The user decides which data that should be revealed in a specific situation. A situation is based on choices made earlier. If the system recognizes the context to be the same as when a particular identity set was used earlier the same identity set will be chosen again. If the system is unable to recognize the situation the user either has to choose the identity set manually or the identity set used in a similar situation will be chosen.

The user can differentiate what personal information that is shared, but the focus is not really on context information as opposed to profile information such as name, address etc. The communication in focus is between a user and institutions such as banks, stores etc. and not if your family or friends should be able to get all information about you at all times.

PawS [26]: In this system Langheinrich chooses a slightly different approach. Instead of denying access to context data the PawS system aims to assure that the collected data is treated in a satisfactory way. The motive is to allow people to respect other people's privacy if they want to. If some systems provide this possibility and others do not, users might prefer to use those who do.

The system provides means to keep track of data after it has left the owner. In order to restrict access to a user's personal data, the user has a personal privacy proxy that handles interaction with other entities which involves revealing personal data. Through the use of P3P policies a user can describe his or her privacy preferences. This policy will be processed prior to a communication process in order to judge the acceptability of the data exchange.

Similar to the Identity management system, the PawS system focus on interaction between a user and an institution. The new elements to this system are the means to keep track of data after it has left the "owner" and the use of policies to describe privacy preferences.

CoBrA [6][7]: This system provides the user with means to define a detailed privacy preference policy and provides reasoning mechanism which enables the system to decide what actions to take in different situations. The system controls if a context consumer is entitled to receive the requested context information by processing the policy.

The CoBrA system has more focus on sharing context information in addition to personal data (e.g. name, address, phone number etc.) than the two preceding examples. This system also focuses on the interaction between the personal devices of two human users, in addition to situations where a user's personal device interact with other system entities such as a server or agent.

E-Wallet [16]: Similar to the CoBrA system the E-Wallet system uses predefined policies to control access to personal data. In addition to determine what information that can be shared based on user identities, this system also considers the situation of the user, similar to the Identity management system. This system also includes a way to abstract away details of the shared information.

4.4.2 Differentiable and customizable privacy handling

One of the tasks in this assignment is to suggest a way to handle privacy in a context management system. This system will be described in Section 5.1. To find a suitable solution privacy challenges and principles have been studied and the four previously described systems have been evaluated. The result of this evaluation is the identification of which elements that should be a part of the system and which requirements that should be fulfilled.

Elements that should be included:

1. Restriction of access to personal data for specific users/entities
2. Situation based access control to context data
3. Limitation of user input requirements
4. The possibility to customize the privacy policy to the individual user's needs
5. The possibility to differentiate the abstraction level of the information which other entities of the system receives

The two first points concern restriction of access to personal data. This might be the most obvious privacy mechanism to include as context data reveals a lot of information about one specific user. This supports Westin's definition on privacy: "the claim of individuals, groups or institutions to determine for themselves when, how and to what extent that information about them is communicated to others" which is the most relevant definition of privacy when we talk about privacy in computing environments. Limitation of access to personal data also supports the design principles of Langheinrich; notice, choice and consent. The principles do

not say directly that access to personal information should be controlled, but by controlling who has the right to collect your personal data you know who is able to collect the data (notice) and you have given the permission (choice and consent). All the four systems described above support this in one way or another. The three different approaches drawn from these examples are; determining access based on the data subjects situation (Identity management), determining access based on the access policy implemented by the data collector (PawS) and determining access based on the identity of the data collector (CoBrA). The E-Wallet system combines situation based and identity based control of access to personal data.

The three next points are not elementary to implement in order to secure privacy in a computing system, but the fulfillment of these requirements will enhance the user experience of the system. It might even be a premise for the user's willingness to use the system. As Langheinrich discusses in his design principles [23]; if notice is achieved trough alerting the user each time data collection occurs, and choice and consent are achieved through manual input from the user, the user experience of the service might be reduced. Under certain conditions the system might be inadequate to use. The nature of pervasive computing environments will make manual solutions like this unsuitable due to the amount of requests for data exchange. The third point on the list, limitations on user input requirements, is therefore not important in order to secure the user's privacy in the system, but to achieve user acceptance of the system.

In order to achieve a minimum of user input the user has to be able to customize a privacy profile after his or her needs. The users' privacy preferences are heterogeneous and vary from situation to situation. A user friendly system should therefore enable the user to predefine a personal profile which can be altered later, but which does not need input as long as the user's privacy preferences do not change to any great extent.

In the CoBrA system the user defines a personal privacy policy. The policy is reasoned over by the system through the use of a reasoning algorithm in order to decide which actions to take (choice and consent). The PawS system use P3P profiles and processing of these in order to check if the user agrees with the privacy policy of the data collection system. The idea in the Identity Management system is that the system will learn what actions to take based on previously made choices. The E-wallet system also uses a predefined user profile which is processed in order to enforce the user's privacy preferences. As seen all the systems use some sort of predefined profiles which are processed in different ways to determine which actions the system should take.

To be able to customize the service to each user's needs, access control should not be based on only granting or denial of access. The user should be able to differentiate to what degree information is revealed, i.e. it should be possible to abstract away some details and reveal the rest. The E-wallet system addresses this issue with the use of obfuscation rules (see Section 4.3.4).

4.4.3 Definitions

The systems which have been evaluated so far have been developed by different research groups and have different approaches to privacy depending on the computing system in mind. The terminology has varied from system to system. Throughout the evaluation the

terminology used by the authors of the articles has been used. It is now time to define the terminology that will be used in the rest of the work on this assignment.

Privacy Policy: Different terms have been used to describe a user's privacy preferences, i.e. how his or her privacy should be maintained. In the rest of this report this will be referred to as the user's privacy policy.

Access/access rights: When the discussion concerns if a system entity (often a user) is allowed to receive the information she/he requests it will be referred to this entity's access rights. Access refers to access to context information.

Context owner: The term context owner will be used to refer to the user which the context information concerns. The context owner is the end user/entity that has the legal rights to the context information and which the information describes.

Context provider: The context provider can be the same entity as the context owner, but not necessarily. The context provider can be an actor that handles context information on behalf of the context owner. It can also be an actor which provides context information independent of a person (e.g. an actor that provides information about the physical environment in a room; temperature, air pressure etc.).

Context consumer: The context consumer is the actor who wants to receive context information about another user/entity. The context consumer can be a user of the system or 3part service provider.
As both the context owner and context consumer can be users of the system these terms will be used to describe respectively the user which the context concerns and the user which receives context information. Using these terms will help to avoid confusion of the two different roles a user might take. These terms are also more descriptive since neither a context provider nor a context consumer has to be an actual person. The term user is usually associated with a person.

Request / Subscription: A request is the message sent from a context consumer to the context management system to get some context information about another entity/user (i.e. context owner). A subscription is when the context consumer wants to receive the same type of context information on a regular basis (e.g. each time a change in this type of context data occurs)

4.4.4 The Privacy Policy Enforcer (PPE)

The design and implementation in this assignment will demonstrate how privacy can be enforced in a context management system. At the same time it will illustrate a possible solution to some of the privacy issues pointed out in this report. The focus is in particular on Westin's definition of privacy to be: "the claim of individuals, groups or institutions to determine for themselves when, how and to what extent that information about them is communicated to others".

Through the use of the Privacy Policy Enforcer (PPE) the user will be given the possibility to predefine a set of rules. These rules are consulted by the context system when a query for context information arrives in order to restrict access to personal data and context information. The motivation for this particular solution is the new situation introduced by

pervasive computing environments as it is for the other system as well. The main goal will be to offer a solution that gives the user the possibility to give various degrees of access to context information to the same entities/users depending on the perceived/current context for the user.

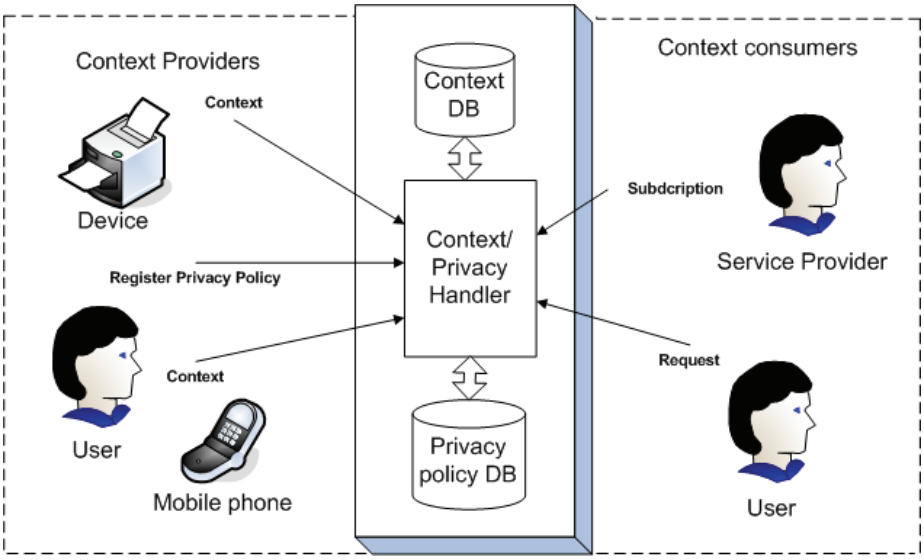


Figure 4-7: The Privacy Policy Enforcer

The mechanism for restricting access to context information will be based on user identity and current context (i.e. the user's situation when personal information is revealed). The choice to restrict access based on these parameters as opposed to the context consumer's privacy policy is that it is not always you want an entity to receive your personal information. Although you know for sure that it will not be further distributed. Evaluating the recipient's privacy policy can be an add-on to a system that restricts access based on identity and current context.

Current context is included as one of the control parameters in order to make the system more flexible. If only identity is used the user has to change the access control parameters manually when his or her preferences changes. With a context based access control parameter the system can better adapt the privacy policy to the user's current situation. A user may want to share information with another user only in some situations.

One last reason for choosing this approach is the type of communication that is envisioned. The solution will mainly focus on enforcement of privacy in communication between two users of a context management system (i.e. communication of context information to a 3rd part via a management system, see Figure 4-8).

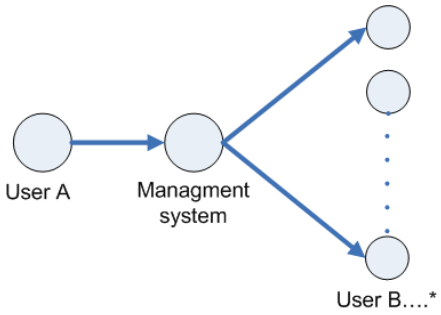


Figure 4-8: Communication of context information to a 3rd part via a management system

The Privacy Policy Enforcer does not necessarily have to be an alternative to the evaluated systems. The PPE can be a supplement to maintain control over who has the possibility to get access to a user's personal information.

5. Enabling technologies

This chapter will give an overview of the technologies which are important for the infrastructure, which is necessary to realize a context management system. The context system itself and the components relevant to privacy handling will be described.

5.1 Akogrimo

Akogrimo (Access to Knowledge through the Grid in a mobile World) [35] is a project funded by the EC under the FP6-IST program. The project goal, on the technology level, is an integration of mobile communication into the Open Grid Services Architecture (OGSA) [1].

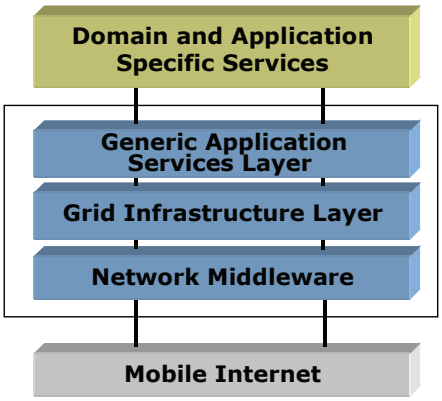


Figure 5-1: Overview of the Akogrimo layers [35]

In this chapter the parts of the Akogrimo architecture that are relevant for context handling and constitutes the infrastructure for the privacy handling which will be outlined in Chapter 6, will be presented. The Network Middleware layer of Figure 5-1 presented above offers A4C (Authentication, Authorization, Accounting, Auditing and Charging) services, service discovery, presence and context management. The focus in the rest of this chapter will be on presence and context management.

5.1.1 The context management architecture

The Akogrimo context management architecture operates with a management system with one central instance, the Context Manager, which handles context data. The system gathers raw context data from numerous sources such as sensors, users or devices/device managers. This data is processed and refined by the Context Manager and further distributed to context consumers which may add extension modules to handle domain specific context inference. To receive context data the context consumers can either execute queries towards the context manager or subscribe for notifications of changes in certain context data. The queries can be made using semantic languages such as RDF or OWL.

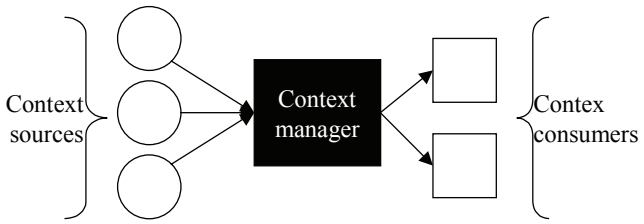


Figure 5-2: Generic flow of context data [35].

The purpose of introducing a system with a centralized Context Manager is to improve scalability and factor out functionality that otherwise had to be duplicated in each service.

Issues addressed by the Context Manager:

- Filtering out relevant data and forwarding this to interested parties.
- Converting heterogeneous context data to a uniform format.
- Dealing with incomplete and inconsistent context data.
- Inferring higher-level context data from basic data (e.g. mapping of user location to location of building).

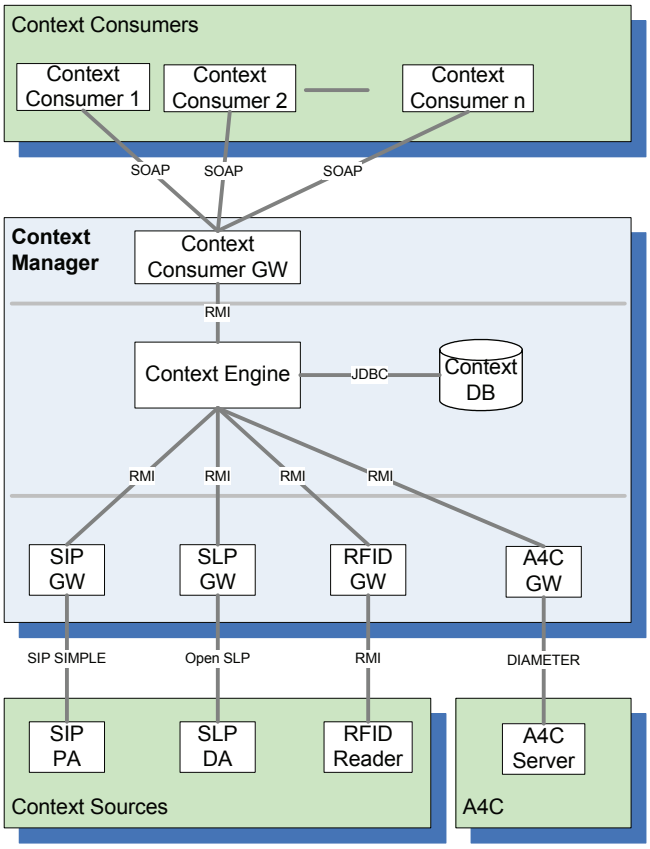


Figure 5-3: Context Manager architecture [29]

5.1.2 Context model

Figure 5-4 shows how context information is organized. There is however not made a specific suggestion on how to handle the user's privacy preferences. "Preferences" which is an attribute of the User entity includes all types of preferences (e.g. preferred devices/services and subscriptions).

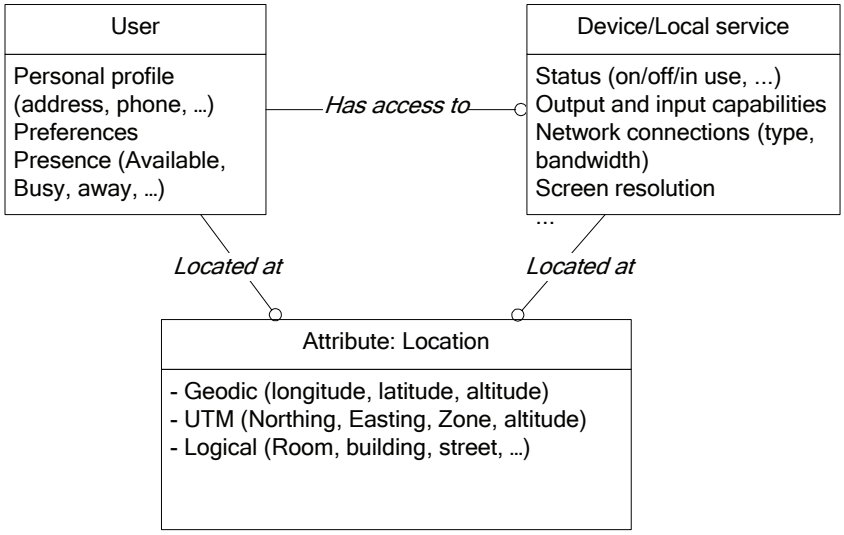


Figure 5-4: User-oriented context [29]

Location: Location systems deliver geographic coordinates of system entities (people (users), things and devices) [29]. The geographic coordinates are used to infer where these entities are located, e.g. which room, street, town, at home etc. A multitude of technologies exist which can be used to makes service that locate and track individuals, such as GPS, WLAN, GSM, RFID and active badges. Universal Transverse Mercator (UTM) is a system which delivers geographic coordinates.

Presence: Presence information refers to information about the state of the users such as availability, reach ability, and other information set by the user (e.g. mood, interest, etc.) [29]. Presence information is the base of instant messaging (IM). This type of information may change frequently. Either it is changed manually by the user or as a result of other available context information such as location and use of a service (e.g. the user talks on the phone).

Devices and local services: Devices and local services in immediate proximity of the user are often an integral part of the application provided to the user [29].

5.1.3 Collecting context data

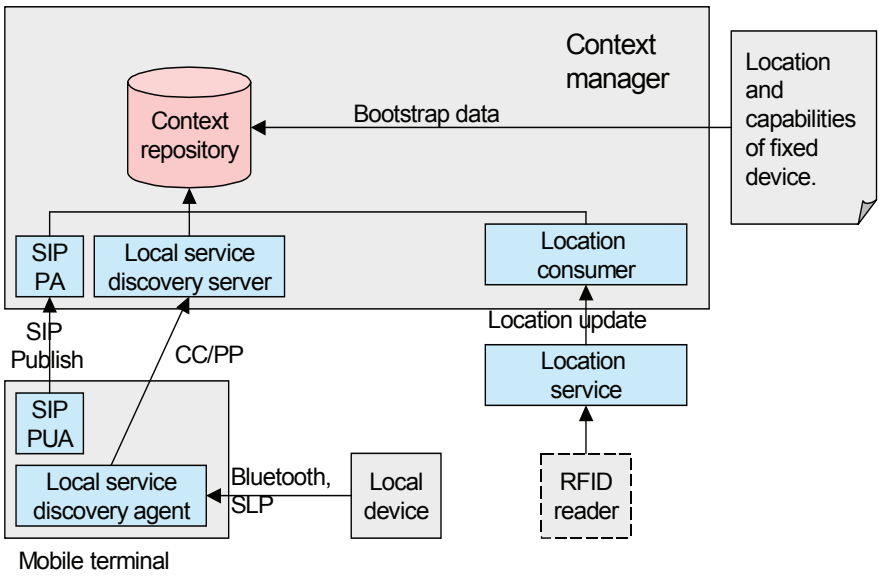


Figure 5-5: Collecting context data [35]

Figure 5-5 shows how context data is collected. The user device, i.e. here the Mobile terminal, contains a SIP Presence User Agent (PUA). The PUA sends SIP Presence data (PUBLISH requests) to the context manager. The Context manager acts as a SIP Presence Agent (PA) and the Mobile terminal also contains a local service discovery agent, which by the use of suitable protocols such as Bluetooth or SLP, discovers devices in the vicinity. When the discovery is performed, the result is reported to the local service discovery server which is part of the context manager. The Mobile terminal also has to report the capabilities of the terminal itself in this process. The local service discovery server will be implemented as a Web service accepting CC/PP documents.

5.1.4 Using and inferring context data

Figure 5-6 shows how context consumers can subscribe to or make a query for context information. The query is done using a Web service which accepts queries in a semantic language. If the consumer subscribes to a given type of context data a notification will be sent each time a change in the relevant data occurs. Web service based notifications will be used for the publish/subscribe mechanism while a semantic language will be used to express the queries and results.

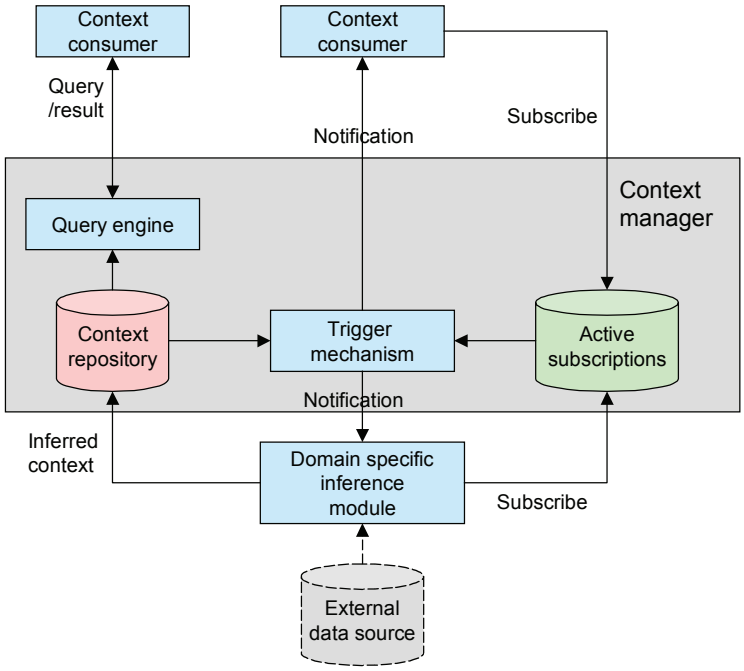


Figure 5-6: Using and inferring context data [35]

6. Design and Implementation

The scenario outlined in Section 1.2 describes a service where a user is able to customize the system's access policy to his needs. The system enables the user to express and enforce his privacy preferences depending on the situation he is in (i.e. his current context). In this way the user is able to benefit from the advantages a context sensitive system can offer and at the same time keep his privacy.

One of the tasks in this project is to demonstrate how such a service can be implemented. The implementation aims to solve some of the previously discussed privacy issues as discussed in Section 4.4. In Section 6.1 the method that was followed during the work on the system design and implementation is described. Section 6.2 presents the overall description of the system and the system requirements. The details of the design are described in Section 6.3 and 6.4, and the implementation is presented in Section 6.5. The test parameters and the test results are presented in Section 6.6.

6.1 Method

This project has to some extent followed the method called Rational Unified Process (RUP). As RUP is intended to be used on larger projects than this, it had to be taken into consideration to which degree all phases and activities were relevant. In addition, the limited time influenced which phases that was possible to complete.

6.1.1 Rational Unified Process (RUP)

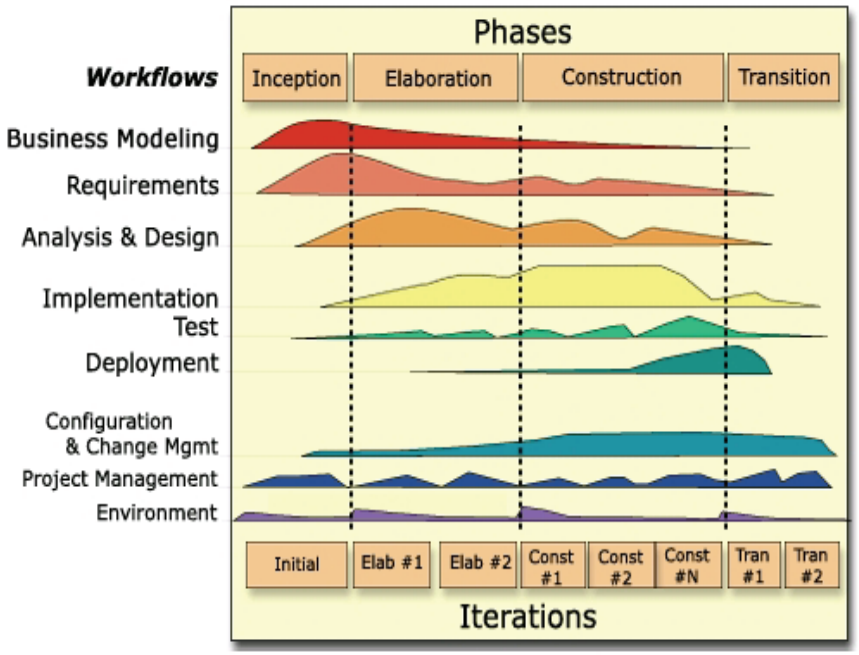


Figure 6-1: The RUP workflow [36]

Figure 6-1 shows the RUP workflow. The horizontal axis represents the time and shows the dynamic aspects of the process model. The vertical axis represents the different aspects of the process in form of different activities [42].

6.1.1.1 Supporting Workflows

The Inception Phase: The goal of this phase is to agree on the objectives of the project. This include establishing the project's scope and boundary conditions, determining the critical use of the system, describe a candidate architecture, estimate cost and schedule and the potential of the project. The activities in this phase include establishing a business case and development of a candidate architecture.

Comment: The first iteration of this phase was carried out through the work with the project assignment [15]. The result of this work has been further evaluated and adjusted.

Elaboration Phase: The goal of this phase is to further analyze the business domain and create the architecture.

Comment: This phase was carried out as described. The design was further analyzed and the architecture was created. The activities of this phase are documented in Section 6.4 of this chapter.

Construction Phase: In this phase the remaining features of the system is developed and tested.

Comment: This phase was carried out as described. The activities in this phase is documented in Sections 6.5-6.6

Transition Phase: During this phase the system is released to the users.

Comment: This phase was not started and will be left to future work.

6.1.1.2 Engineering workflows

The core engineering workflows of the RUP model, as illustrated in Figure 6-1, are the upper six.

Business modelling: The purpose of business modelling is to be able to identify one or more business cases where the project can play a part. As this is a student assignment this was not looked into during this project.

Requirements: The goal of this activity is for the developers and the customers to agree on the system's functionality, what the system shall do and how the software works.

Comment: The requirements were identified through creation of a scenario (Section 1.2) and a study of existing systems and technologies (The result is presented in Section 4.4). This resulted in a description of a concept. The requirements were verified through Use Case modelling.

Analysis and design: The intent of the analysis and design activity is to create a blueprint of the software which describes in detail how the software works.

Comment: The work carried out in this phase is documented in Section 6.4. The design includes an overview of the context model and a detailed overview of the privacy model. The overview aims to give a better understanding of each part of the system and the functionality provided by each element. An information model is included in order to get an overview of the information contained in the system and the organization of this information. Message sequence charts and process graphs illustrate the information exchange between the different entities in the system.

Implementation: This activity is when the actual software is written.

Comment: This phase was carried out as described. The Privacy Policy Enforcer was implemented in order to demonstrate the idea for access control to context information which was presented in 4.4. The documentation of this phase is presented in Section 6.5.

Test: The goal of this activity is to verify that the implemented software behaves according to the design and the requirements.

Comment: The activity was carried out through creation of different test sets and execution of the tests. The test parameters and the test results are presented in Section 6.6. A more detailed description is included in Appendix B.

6.2 Overall Description and Requirements

The Privacy Policy Enforcer (PPE) was presented in Section 4.4. As described, the user will have the possibility to predefine a set of rules. The context management system will be able to consult these rules when a query for context information arrives. The system will consist of two parts; a user interface where the user can fill in values to define his or her privacy preferences and a control mechanism which checks if context consumers are allowed to receive the information they either request or subscribe to. The user will be able to change his or her privacy preferences whenever it is needed.

Through the user interface the context owner will be able to give different types of access to different context consumers. This will be achieved by organizing the context consumers in groups. Each group will be granted certain access rights to context information. The context owner will also have the possibility to define different situations based on his or her current context and the point in time. By associating a group of context consumers, a situation and a type of context information with a value the system will be able to determine which context consumer entities that should receive what information under what circumstances.

As stated in Section 1.3, this design will focus on privacy concerns related to context dissemination. The PPE system will be designed and implemented in order to demonstrate how privacy can be enforced from the moment context has been collected by the context management system and until it is disseminated to context consumers. The object is to prevent a third party from getting access to personal information without permission from the context owner.

The technology which is needed to collect and communicate context information to the context management system is assumed to be implemented as described in Chapter 5. The management of context information is handled by the Context Manager as described in Section 5.1. The handling of context information to control privacy is the task of this assignment. The processing of context information for other purposes is already dealt with by the Akogrimo Context Manager.

Table 6-1: The system requirements

Requirement	Description	Priority
S1	The user shall be able to grant/deny other specific users/system entities access to his or her context information.	High
S2	The user shall be able to define access to his or her context based on the context consumer's id, his or her current context and the type of context (see Figure 2-1) that will be shared.	High
S3	The system should require a minimum of user input from the context owner	
S4	Current context information that should be taken into account when deciding access rights are: • presence (occupation) • localisation • time of day • available services	High
S5	The user should be able to specify the accuracy of the context information to which the context consumer is granted access. (building vs. room etc)	High

6.3 Use Cases

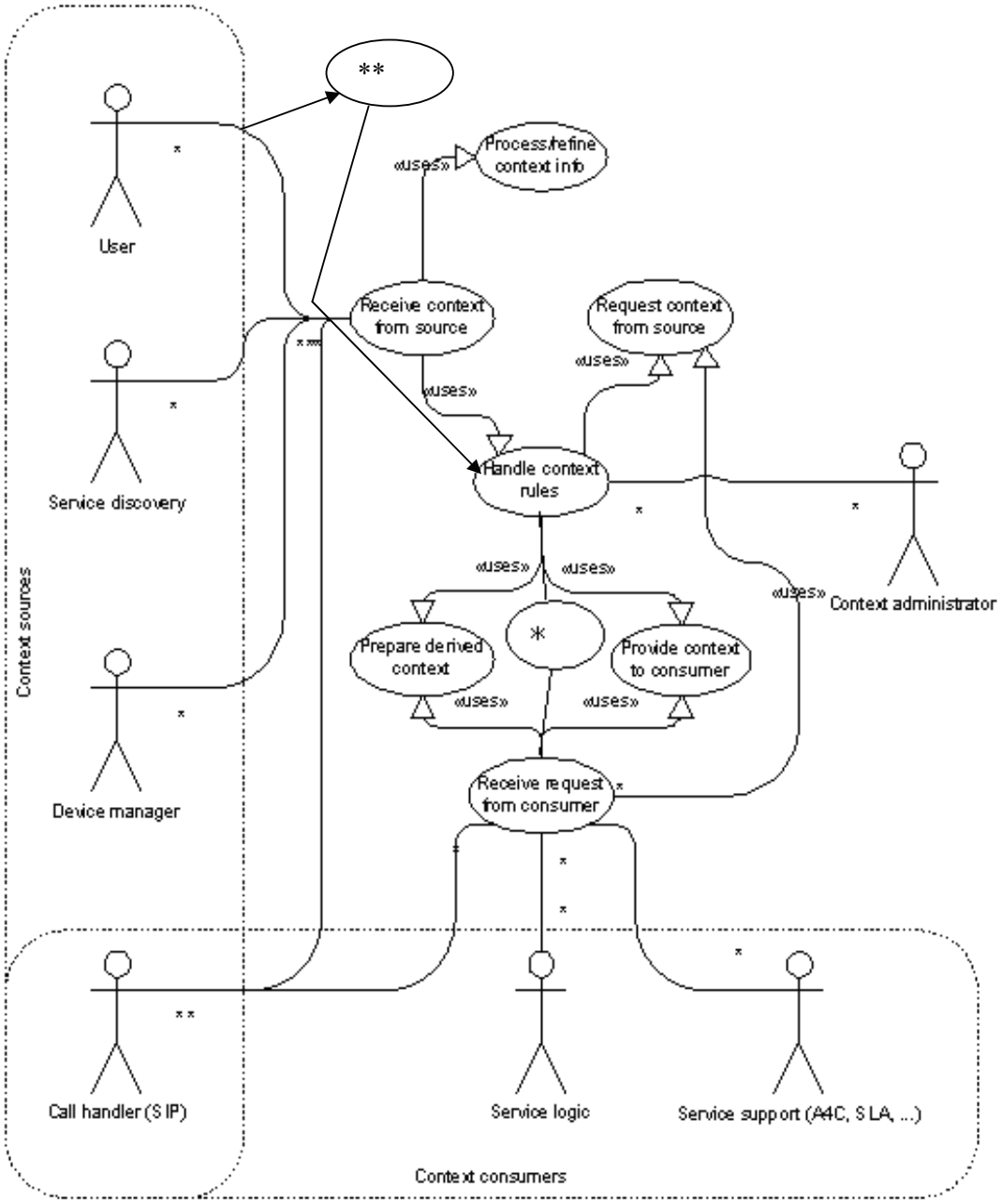


Figure 6-2: Actors and Use Cases relevant for context [35]
(* Handle privacy Use Case, ** Handle User input Use Case)

The Use Case in Figure 6-2 gives an overview of the relations between Use Cases and actors in the Akogrimo design [35], which are relevant for context handling. In order to enforce privacy the "Handle privacy"-Use Case (*) and the "Create Privacy Policy"-Use Case (**) have been added to the original Akogrimo figure. In order to identify the high level user-functional requirements of privacy handling in the system these Use Cases will be further outlined in the rest of this chapter.

The actors in Figure 6-2 which are involved in the Use Cases relevant to privacy handling are the user and the context consumers (Call handler (SIP), Service logic, Service Support). The "Receive Request from consumer"-Use Case is used by consumers to specifically request context through a well-defined API. A normal flow of tasks will be as follows [35]:

1. Receive external request
2. The Use Case verifies A4C and the semantics of the request
3. Context is retrieved
   - from persistent data, or
   - requested from sources if necessary (by using the "Request context"-Use Case)
4. Engage the "Handle privacy"-Use Case to verify that the consumer is allowed to receive context (*)
5. Engage the "Prepare derived context"-Use Case if necessary
6. Resolve conflicts concerning inconsistent context data.
7. Engage the "Provide context"-Use Case to reply to the request (reply with an error message in case of errors)

(*) This point is added to the original flow depicted in the Akogrimo report.

6.3.1 Create Privacy Policy

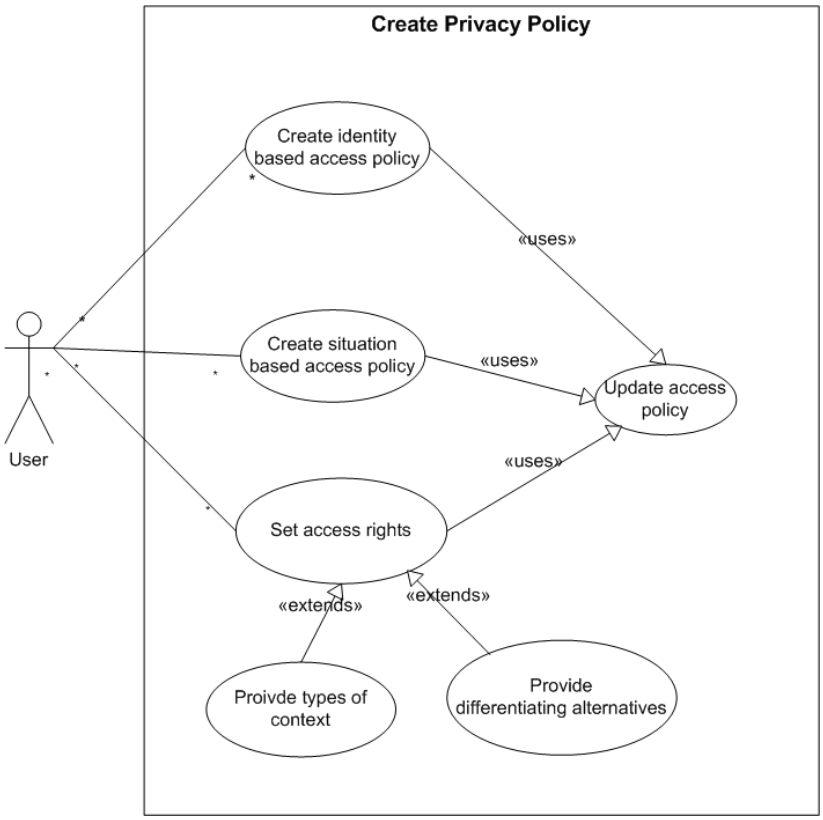


Figure 6-3: Use Case: Create Privacy Policy

Figure 6-3 shows the high-level functionality of the processes when the user creates the Privacy Policy. The user interacts with three different Use Cases through a user interface which in turn uses the “Update access policy”-Use Case. The “Provide types of context”-Use Case and the “Provide differentiating alternatives”-Use Case extend the “Set access rights”-Use Case. The first of these Use Cases is used when the user chooses which type of context she/he wants to share with other users. The latter Use Case is used if the user wants to hide some details of the type of context she/he wants to share. A normal flow in the “Create Privacy Policy”-Use Case will be:

1.
    a. Create identity based access policy
    b. Update access policy

2.
    a. Create situation based access policy
    b. Update access policy

3.
    a. Set access rights
        i. Choose type of context
        ii. Choose differentiating level
    b. Update access policy

When this flow is completed once, the order of number 1, 2 and 3 does not matter. All steps do not have to be included.

6.3.2 Handle request from consumer

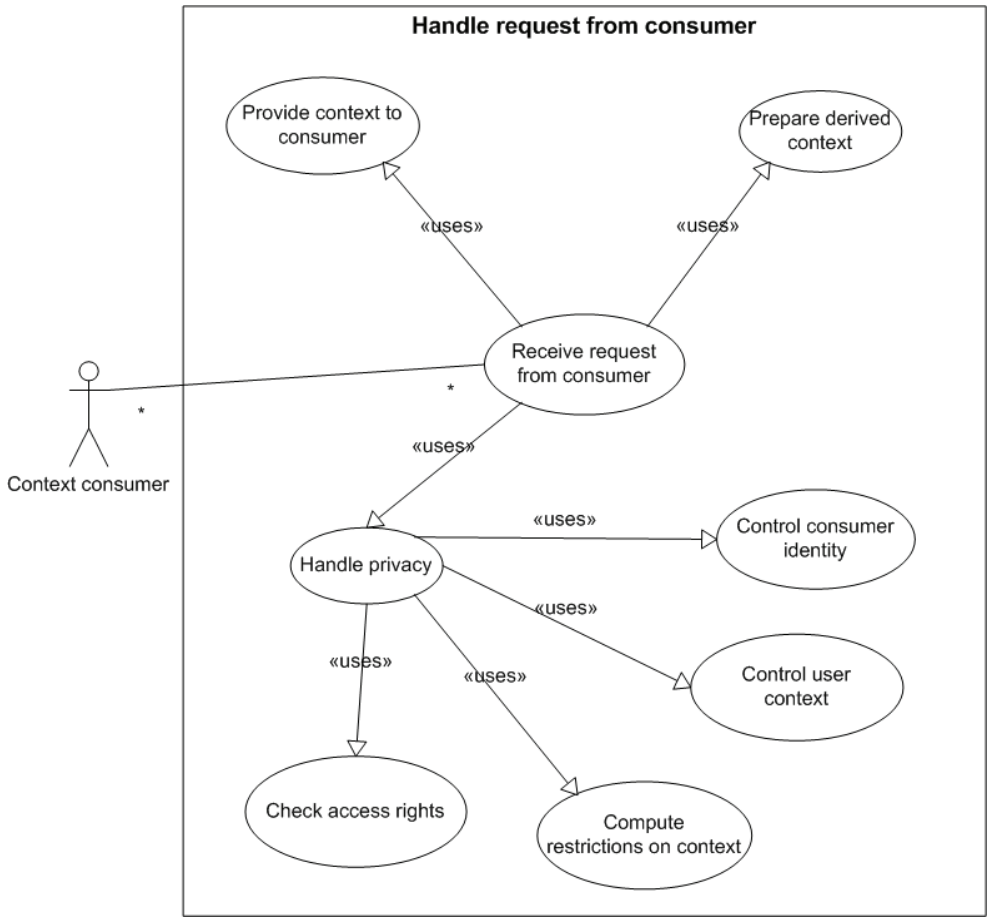


Figure 6-4: Use Case: Handle request from consumer

Figure 6-4 shows an extract of the Use Case in Figure 6-2, including only the Use Cases used by the "Receive request from consumer"-Use Case. The purpose of this figure is to give a more detailed picture of the functionality of the "Handle privacy"-Use Case. The "Handle privacy"-Use Case engages the Use Cases which control if the consumer is entitled to receive the requested context information. A normal flow of "Handle privacy"-Use Case will be:

1. Receive request from "Receive request from consumer"-Use Case
2. Engage the "Control consumer identity"-Use Case
3. Engage the "Control user context"-Use Case to check if the context owner is in a situation where she/he wants to share the requested context
4. Engage the "Check access rights"-Use Case
5. Engage the "Compute restrictions on context"-Use Case
6. Return the access rights and restrictions

6.4 Design

In this section the design of the system will be provided. The work documented here is referred to as the "analysis and design phase" of the RUP-method. The design includes an overview of the context model and a more detailed overview of the privacy model. A short overview of which features that are relevant to include in the system, but which will not be included in the implementation will also be presented. A description of the information contained in the system represented by an Information model, Message Sequence Charts and Process graphs are also presented.

6.4.1 The Context Model

The context system which handles all processing of context which is not directly related to privacy handling is the Akogrimo context manager system as described in Section 5.1. The Context Manager is the central unit of this system. The module which handles privacy enforcement will be a part of this unit. Figure 6-5 illustrates the different elements in the system architecture and the flow of information between the three units; context providers, the Context Manager and context consumers. The context providers provide context information to the Context Manager. The user is both a context provider and context owner. In addition to providing context information to the Context Manager the user also provides the privacy preferences which are handled by the PPE. The context consumers send queries or subscribe to context information.

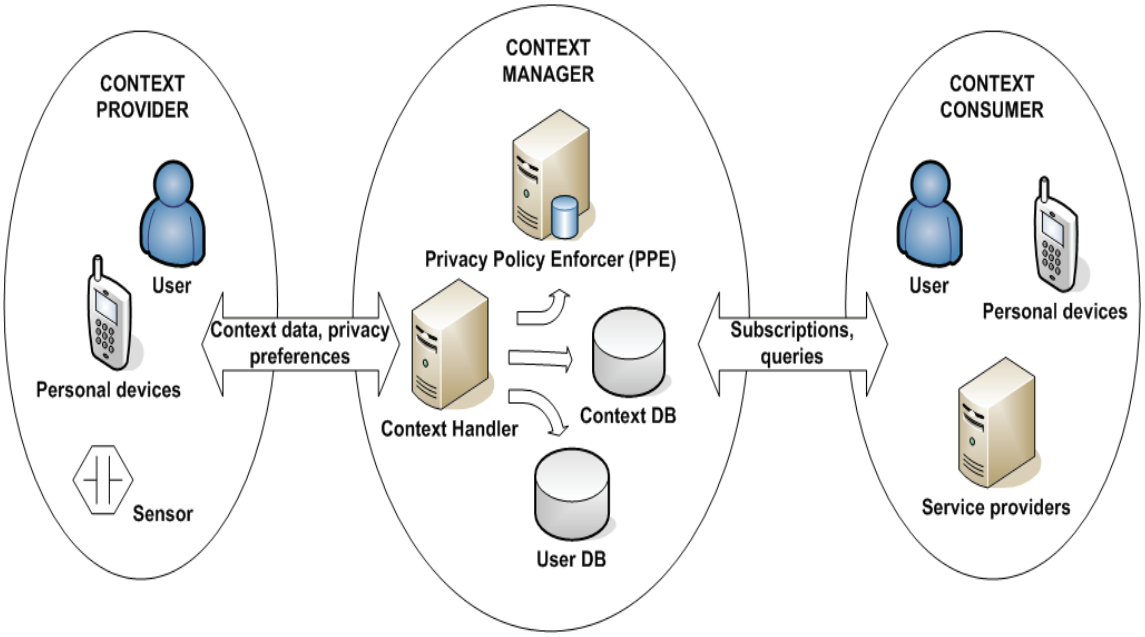


Figure 6-5: Overview of the system architecture

6.4.2 The Privacy Model

Privacy handling is divided into two parts; the initiating part and the control part. In the initiating part the context provider defines the Privacy policy. In the control part it is verified if a context consumer is allowed to receive the requested context information. This check is carried out each time a context consumer initiates a request for context information. The control is also performed to check if a context consumer's subscription to context information is valid when a change in context occurs.

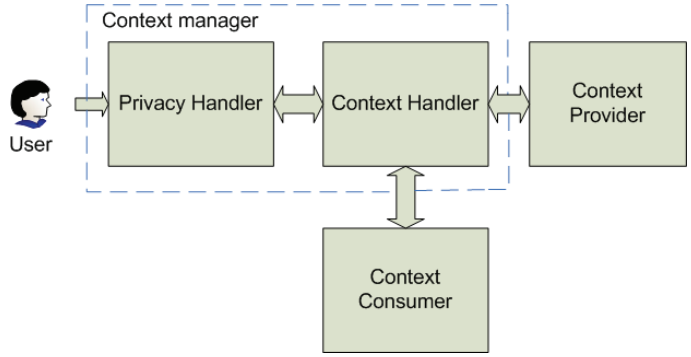


Figure 6-6: The Communication flow in Privacy handling

Figure 6-6 illustrates the communication flow between the different units of the context management system in order to handle privacy. In Figure 6-5 the user is a part of context providers, while in Figure 6-6 the user is showed as separate elements. This is done to show that the user gives input directly to the Privacy Handler. This happens when the user creates or up dates the privacy policy.

6.4.2.1 Part one – creating the Privacy Policy

The context owner defines a privacy policy by giving values to the parameters of three objects; Access groups, Current context and Shared context. Table 6-2 illustrates the different objects and their parameters.

Table 6-2: The objects and parameters of the privacy policy

Object	*Parameters*	
Access groups	Name of group	
	List of context consumers	Name of context consumer
		Context consumer id
Shared context type	Context type	
	Detail level	
Current context	Name of situation	
	List of context types	Values
	Time interval	Start time/ stop time

6.4.2.2 Access Groups

The context owner creates groups of context consumers by giving the group a name and adding the context consumers to the group by listing their name and identity (user@domain). To allow a context consumer to receive specific information the consumer's name and identity has to be a part of the group which is associated with this context information. The context consumers listed in a group do not have to be actual persons. The identity can belong to other system entities such as 3rd part service providers (e.g. id=service@serviceprovider) or web sites. The identity can also belong to a role (e.g. id=administrator@....). This will not always be the same person, but this person's actions will be limited by the role.

Example: Tom, a user of the system, creates a group called "Family". In this group he wants to add three persons, his two children and his wife. He does this by adding their full name and their user-identity in the system. He can then give specific access rights to those four.

6.4.2.3 Shared Context Type

The context provider defines what kind of information she/he wants to share with context consumers. This is done by giving value to the Shared context type and the Detail level parameters. The Shared context type parameter can be set to Location, Presence, SLP service or Contact information. Detail level can be set to level 1, 2 or 3. For each Shared context type the Detail level gives the level of abstraction presented to the context consumer.

Example: Tom wants to share his location information with his friends, but not the exact position. The parameters of the Shared context object could then be as follows:

- Shared context type= Location
- Detail level = 2.

Tom will now share his location context with a predefined radius of 100m (see Table 6-3: Definitions of Detail level). As a result his friends will only be able to see the area (including e.g. one or more buildings) he is in and not the exact location (e.g. room or precise coordinates).

Table 6-3: Definitions of Detail level

Shared context type	*Detail level*	
Location	Level 1	Radius = 0m
	Level 2	Radius = 100m
	Level 3	Radius=1000m
Presence	Level 1	Set 1 of values of presence status
	Level 2	Set 2
	Level 3	Set 3
SLP service	Level 1	Set 1 of different services
	Level 2	Set 2
	Level 3	Set 3
Contact information	Level 1	Share: all
	Level 2	Share: job related information (phone number etc.)
	Level 3	Share: phone number

In the implementation of the PPE system the values of the Detail level will be fixed values.

It is possible to give the user the possibility to define the values of the different detail levels. More context types and detail levels can also be added. This will however be left to the next iteration of the system development.

6.4.2.4 Current Context

The user can define different situations she/he might be in during the day. This is based on different context types and the value of the type when the user is in the given situation.

Example: Tom wants to define a situation for when he is at work. He calls the situation "Work" and chooses Location to be one of the context types the situation should be based on. When he chooses the context type Location he has to set the values, Utm-northing and Utm-easting[2], to be the values they will have when he is at work. He can also set the time for when he is normally at work. He can then give certain access rights based on his own situation. If he wants to share certain information only when he is at work, the system will check if his location has the given values, and if the time is matching with the time interval. If the result is correct the context consumer can get the requested information.

Table 6-4: Definition of Current context values of the situation "Work"

Context Type	*Values*	
Location (see description)	Utm-northing[2]	235739579
	Utm-easting[2]	174297920
Presence	Status	"Online"
Service Information	Type of service	"WLAN"

Location: When the context owner defines the Current context situation he provides the exact position coordinates (Utm-northing and Utm-easting). However, when a context consumer requests context information it is not likely that the context owner has this exact position. To check if a Current context situation applies when the context information is requested the system will therefore check if the user is inside an area with the given position as its centre. This value can either be set by the system administrator or by the context owner. In the implementation this will be a fixed value of 100 meters.

6.4.2.5 Set Access Rights

When the context owner wants to give certain persons access to parts of his or her personal information under certain conditions, an association is made between one instance of each of the objects; Access group, Current context and Shared context type. This association will be set to the value 1 to tell the system that the access is granted. To deny the access the value is changed back to zero, which is the initial value.

Example: Tom wants his family to be able to se his location, with detail level 2, when he is at work. He therefore makes an association between the three objects which has been described in the previous examples.

[2] See Section 5.1.2 for computation of Location values.

- ("Family", "Shared Context", "Work")

This will give the situation marked with (*) in Table 6-5..
The access rights (1/0) can be thought of as the elements in a three dimensional matrix.

Table 6-5: Example of information contained in a privacy policy

Shared Context / **Current Context**	**Type**: Location **Detail level**: 2	**Type**: Presence **Detail level**: 2
Name: Home **Type**: Location **Value**: Position **Time interval**:17:00:00-07:00:00	Groups: **Name**: Family **List Of users (name/id):** Alice/alice@domain Bob/bob@... Sam/sam@...	Groups: **Name**: Family **List Of users (name/id):** Alice/alice@... Bob/bob@... Sam/sam@...
Name: Work **Type**: Location **Value**: Position **Time interval**: 08:00:00-16:00:00	(*)Groups: 1.**Name**: Family **List Of users (name/id):** Alice/alice@... Bob/bob@... Sam/sam@...	Groups: 1.**Name**: Family **List Of users (name/id):** Alice/alice@... Bob/bob@... Sam/sam@... 2. **Name**: Colleagues **List Of users (name/id):** Per/per@... Lisa/lisa@...

6.4.2.6 Part two – The control mechanism

When the context owner has registered all the privacy preference parameters and the privacy policy is created, the system will consult this policy prior to communication of context information to other system entities. If the access right is set to 1, access to the requested context information is granted, otherwise it is denied. The access will be denied if the context consumer is not part of a group or if the consumer is part of a group that is not allowed access to this particular information. Access will also be denied if the current context of the context owner is not in accordance with the situation when the requested information is shared with this group.

If a request arrives from a context consumer that does not belong to any of the groups, the consumer is added to a list of subscribers. This list will be checked when a context change occurs or when the user changes the privacy policy. The subscription will be removed from the list if it remains invalid for a certain period of time.

When access to context information is granted the system has to check for restrictions, i.e. the abstraction level of the information which is given by the Detail level. The Detail level only gives how strict the restriction is and not what it actually means. This is computed by the system depending on what type of context the restriction concerns. If the context type is Location the restrictions will typically give a larger area description (i.e. building instead of room). If the context type is contact information fewer details will be given. Figure 6-7 illustrates the different processes in the progress of privacy handling and the communication flow between the different entities.

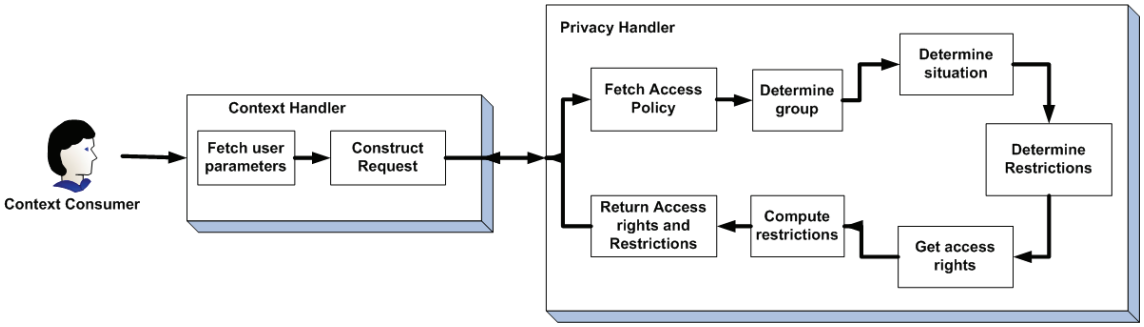


Figure 6-7: Process in the progress of privacy handling

6.4.3 Information model

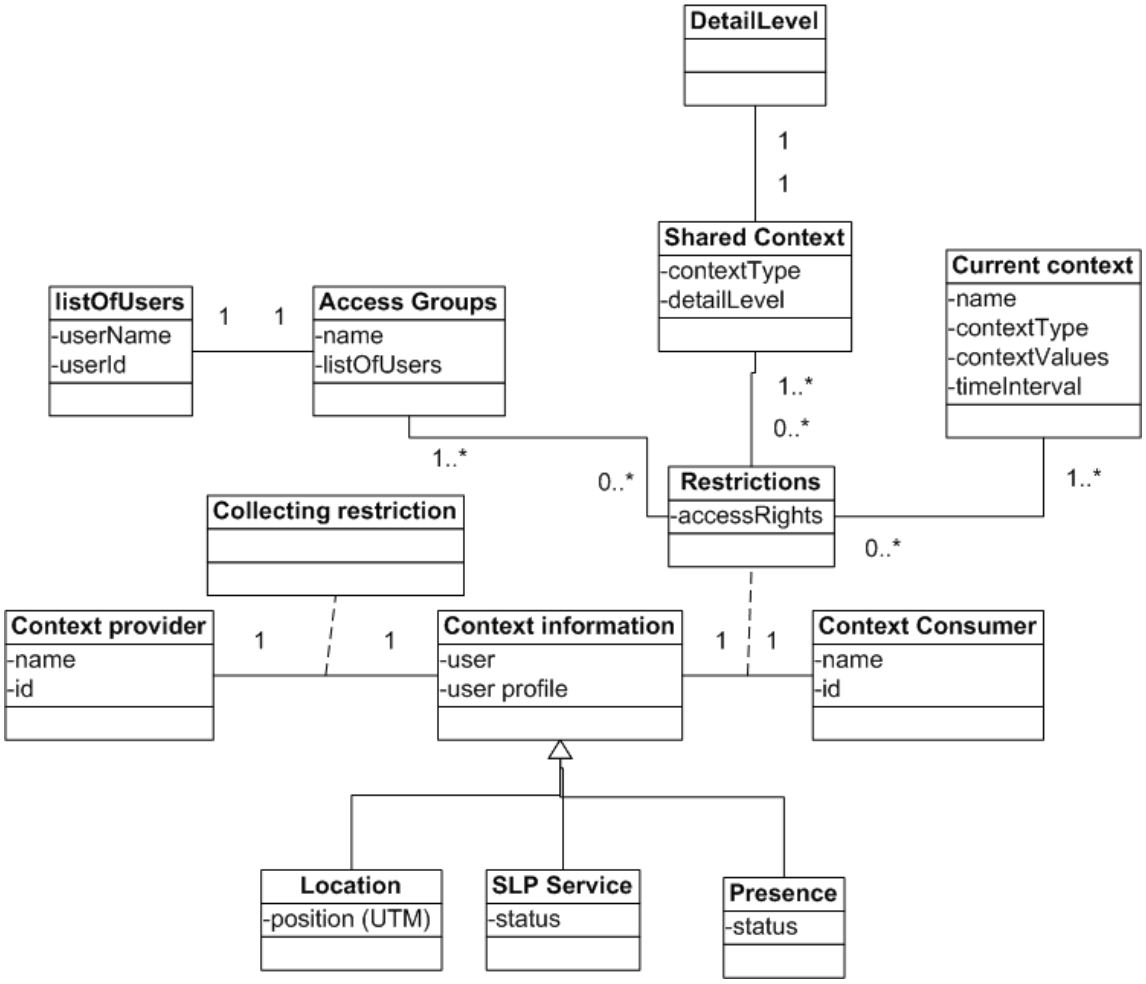


Figure 6-8: Information model

Figure 6-8 shows the information model which represents the context information in the system and the restrictions which are put on dissemination of context information to context consumers. The model shows the entities which are part of the restriction on the association between context information and context consumer. The information contained in these entities is predefined by the user through a user interface. The context associated with one particular user is dynamic and changes according to the user's situation.

Each context provider is associated with a set of context information. Context information consists of Location, Service and Presence information. This association is constrained by a Collecting restriction. This restriction is not further outlined in this assignment, but is included in the figure as it should be a part of a complete system. Each context consumer will be associated with the context information a context owner when the consumer requests to see part of this information. This association is constrained by a set of restrictions. The restrictions consist of Access groups, Shared context and Current context.

6.4.4 Message Sequence Charts (MSC)

This section shows the interaction between the different components of the system that takes part in the privacy handling.

6.4.4.1 Creating the Privacy Policy

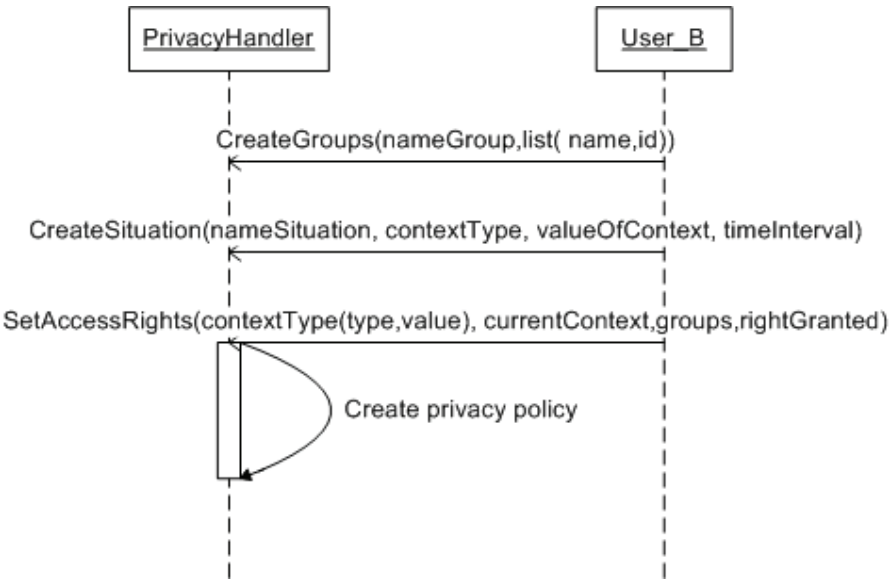


Figure 6-9: MSC – Creating the Privacy Policy

The MSC in Figure 6-9 shows how one user, user B, defines an access policy by sending the messages, CreateGroups, CreateSituation and SetAccessRights. The message CreateGroups contains the parameters nameGroup and list. NameGroup is the name the user chooses to give the group, and list is a list of all the context consumers the user has added to this group. The CreateSituation message contains the parameters nameSituation, contextType, valueOfContext and timeInterval. The SetAccessRights message is sent when the user is making an association between an Access group, a Shared context type and a Current context situation to tell if access is granted for this instance. The privacy policy is computed based on these input parameters from the user.

6.4.4.2 Request message from context consumer

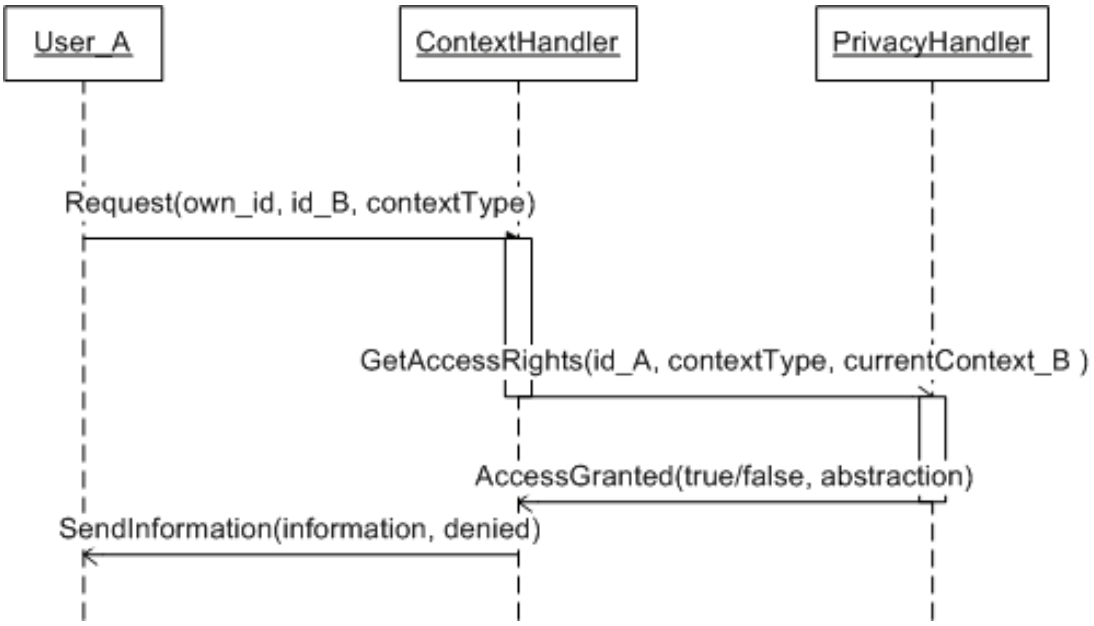


Figure 6-10: MSC – Request

The MSC in Figure 6-10 shows the interaction between the context consumer, User A, and the context and privacy handlers. User A requests some information about User B. The Context handler receives the request and processes the information in order to get the parameters necessary to compute a GetAccessRights message. This basically means to retrieve the current context information about User B. This information is sent to the Privacy handler which compares the parameters in the GetAccessRights message with the parameters of the privacy policy. If the result of this processing is true, access is granted. If the result is false it is denied. The level of abstraction tied to the context information is computed and sent as a part of the AccessGranted message.

6.4.4.3 Handling of subscriptions

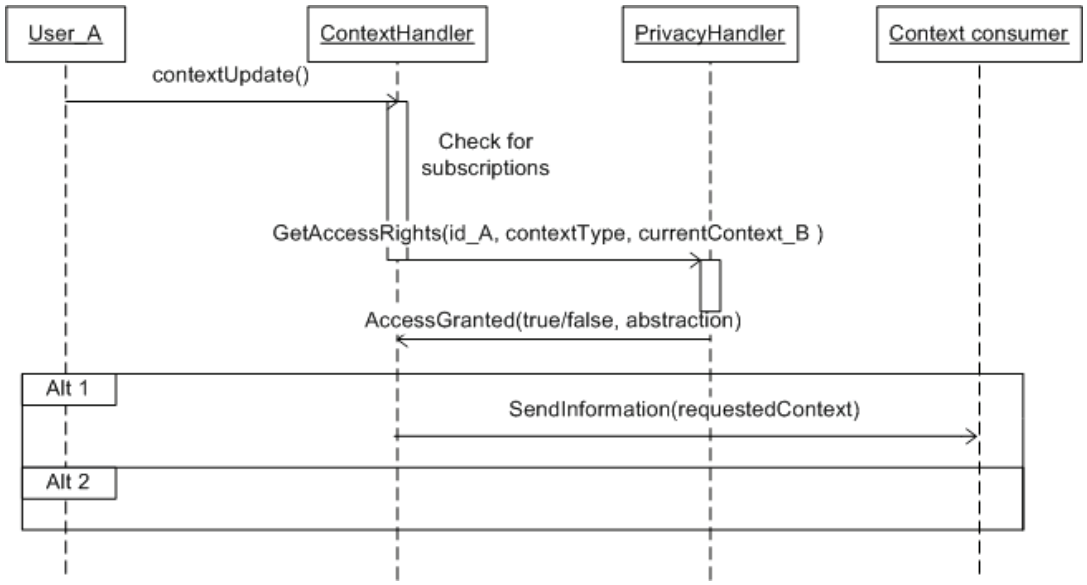


Figure 6-11: MSC – Subscription

The MSC in Figure 6-11 shows the interaction when a subscription message arrives. The exchange of messages is the same as when a request arrives, except for the response. If access is granted the information is sent, but if it is not granted, the context consumer (User A) will not receive any notice at all. The context consumer shall not be made aware that a change in context has occurred as this will indirectly reveal information about the context owner.

6.4.5 System behaviour

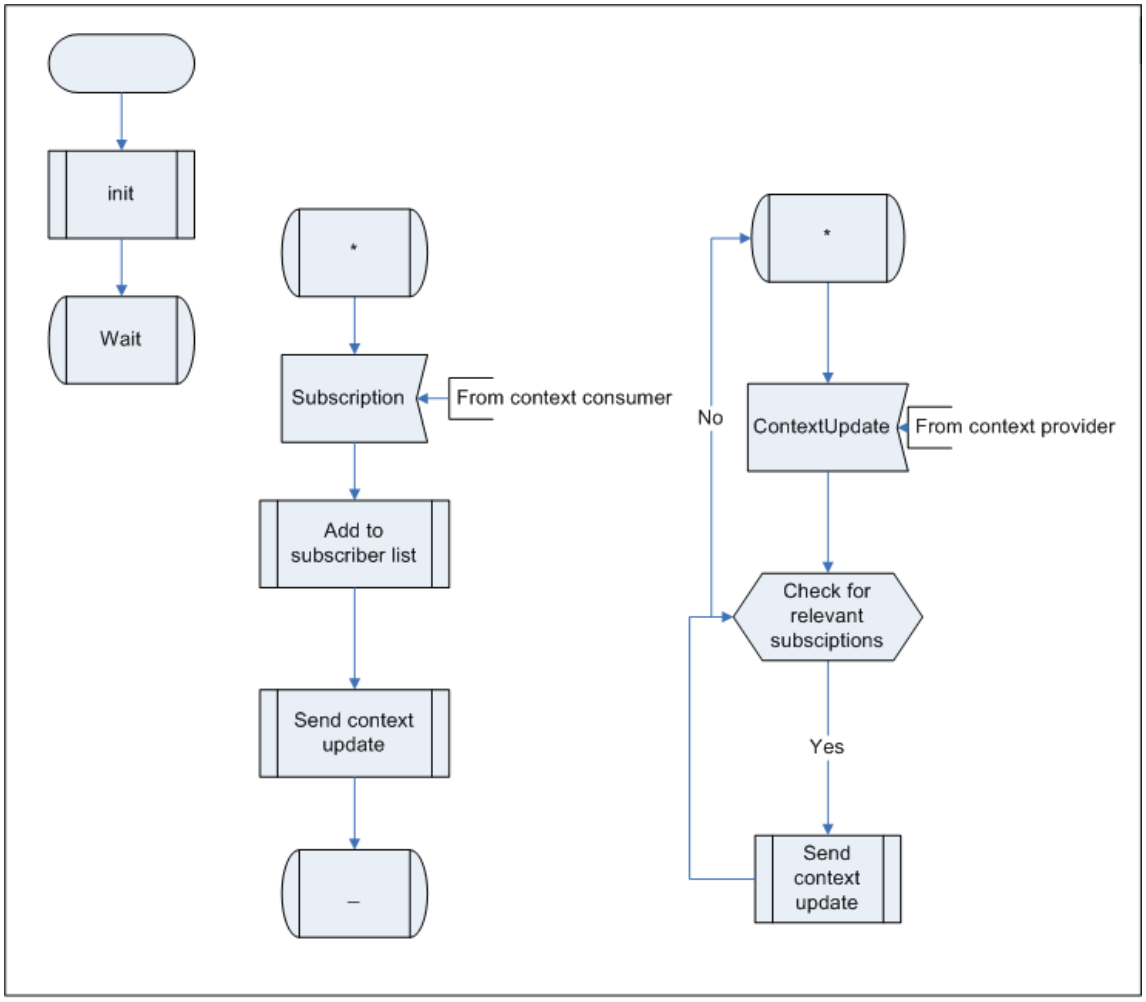


Figure 6-12: Processes in the context handler

The system behaviour is described with SDL process diagrams. Figure 6-12 illustrates the processes in the Context handler. The process is waiting for a subscription or contextUpdate signal. The Subscription signal is sent from a context consumer when the consumer initiates a request. The consumer is added to a subscriber list if it is not already part of the list. The "Send context update"-procedure will then be invoked. When context information tied to an entity changes an update is sent to the Context handler. This triggers a check for relevant subscriptions. If the result is yes the "Send context update"-procedure is invoked and the process loops back to check if there are more relevant subscriptions. This continues until all subscriptions have been handled and the process returns to the state wait.

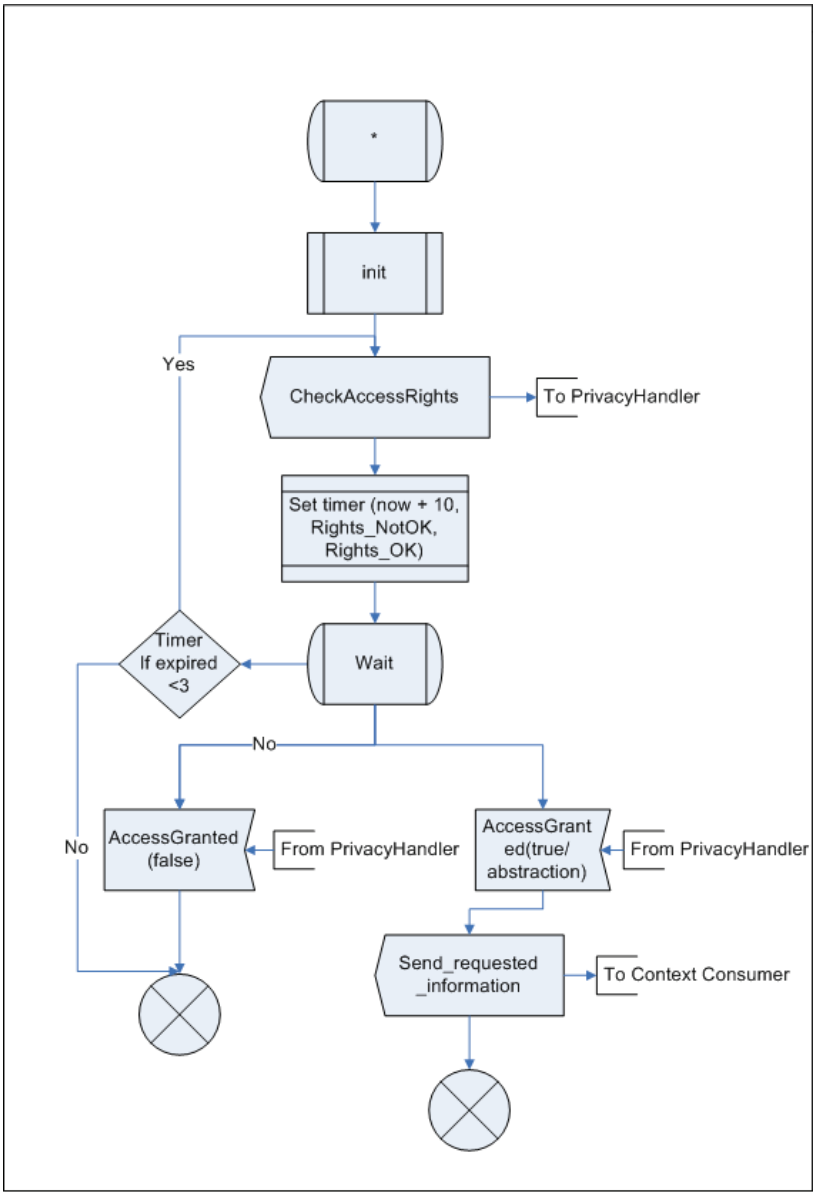


Figure 6-13: Procedure – send context update

Figure 6-13 illustrates the "Send context update"-procedure. The procedure is initiated as described previously. A "CheckAccessRights"-signal is sent to the privacy handler in order to control if the subscription is valid. A timer is set and if the response is not arrived before 10 time units have passed the signal is sent once more. This is repeated three times. If the response is still not arrived it is assumed that the rights where not ok and the procedure ends. This is the same result as if the "Rihgts_NOK"-signal arrives from the Privacy handler. If the "Rights_OK"-signal arrives the requested information is sent to the Context consumer.

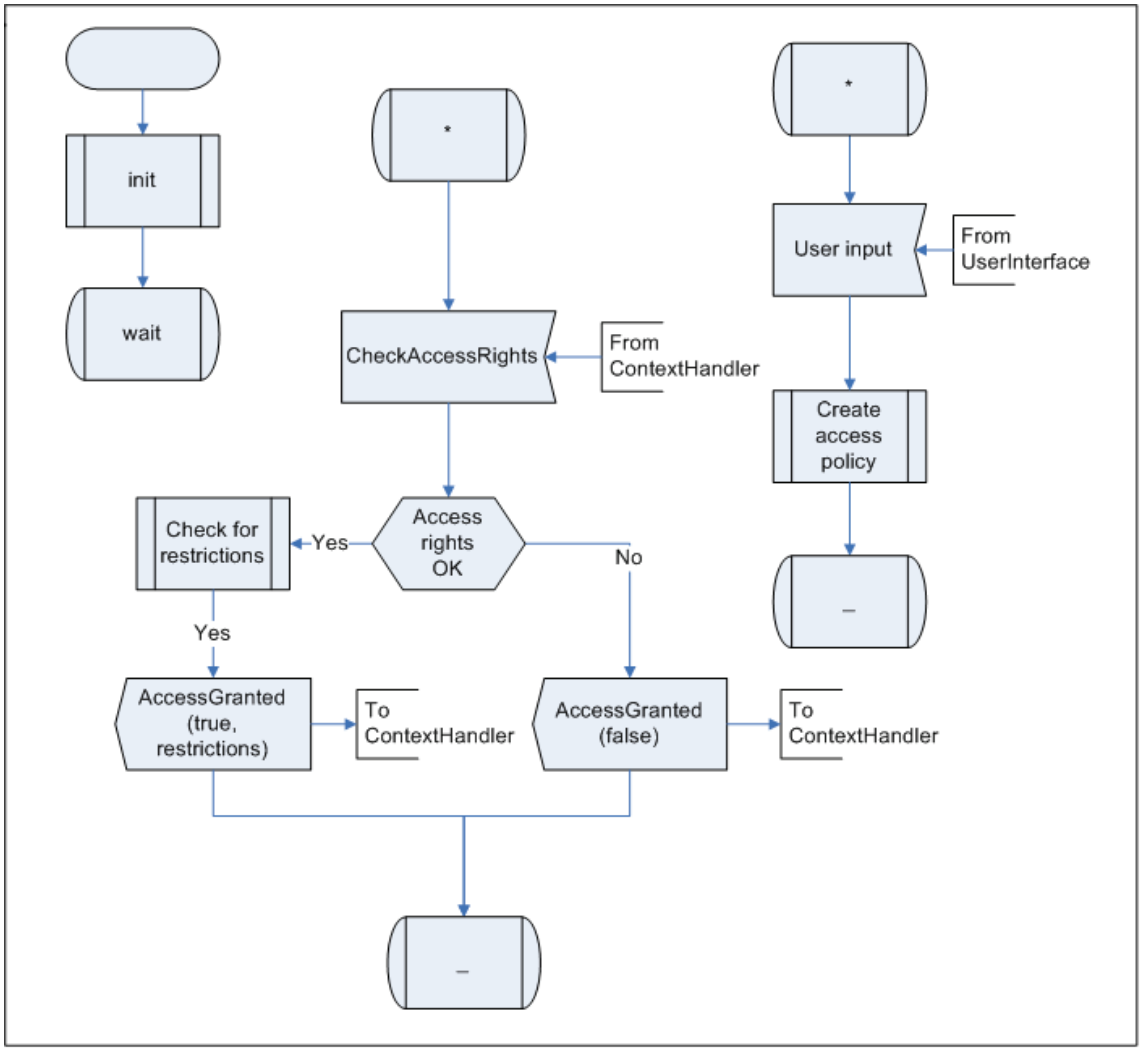


Figure 6-14: Processes in privacy handler

Figure 6-14 illustrates the processes related to privacy handling. Upon initiation the process rests in a "wait"-state until either a "UserInput"-signal or a "CheckAccessRights"-signal arrives. The user input initiates the "Create access policy"- procedure. The "CheckAccessRights"-signal is a result of the "Send context update"-procedure illustrated in Figure 6-13. The access rights are controlled and if the result is ok a check for restrictions are made. The "AccessGranted"-signal is returned to the Context handler containing two parameters; one which tells that access is granted and one which gives the restriction. If the result of the access control is negative a parameter with the value false is returned to the Context handler

6.4.6 Further design which will not be implemented

Due to limited time some features that should be a part of the complete system will not be implemented during this iteration of the development process. The implementation will aim to demonstrate the core features of the design while features that would give added value in user experience will be left for future work. This section presents the most important features that should be considered.

6.4.6.1 Handling of consumer identities

A request from a context consumer will be rejected if the consumer identity is not part of any of the Access groups. This could however be a new contact which the context owner wants to share information with, but does not yet know the identity of. In this situation the Context owner should get some kind of notification and a query if this identity should be added to any of the groups. Alternatively, in addition to the groups, there should be a list of contacts which the Context owner can later choose to add to a group. A new consumer identity would be added to this list. The process graph in Figure 6-12 shows that every request is treated as a Subscription. If the identity is unknown the context consumer will be added to a list of subscribers which will be activated if the context owner later registers this identity in any of the groups which makes the subscription valid

6.4.6.2 User Interface improvements

In the design outlined until now the possibility for differentiating context types is predefined by the system administrator. This could also be input parameters from the user. However, different users have different preferences. Some want high flexibility and personalization possibilities, although this often includes a higher administration burden on the user, while others want it as simple as possible with a minimum of user input.

A solution to meet demands in both directions is to make different versions of the system based on these two approaches:

1. Configured by system administrator: The definition of the differentiating of context types are handled by the system. The user only chooses levels (this is the solution that will be implemented in the demonstrator).
2. The user decides what the differentiating of context should imply (e.g. the user sets the radius in differentiating of location information instead of the predefined values described in Table 6-3).

6.4.6.3 Differentiating contact information

Differentiating of contact information can be used by the user to show different identity sets (e.g. as it is described in the Identity management system). Some of these sets could represent different pseudonyms. This is a possible solution on how to handle anonymity and pseudonymity.

6.5 Implementation

In the first part of this chapter the complete system has been described. In this chapter the implementation details will be outlined. The implementation of the demonstrator illustrates how the processes in the context handler can be implemented. The demonstrator includes an implementation of the send context update procedure, illustrated in Figure 6-13, and the processes in the context handler, illustrated in Figure 6-14.

6.5.1 Overview

The implementation consists of several java packages. Figure 6-15 shows the packages which are included in the privacy handling procedure. The contextmanger.entities package is part of the Akogrimo architecture [35].

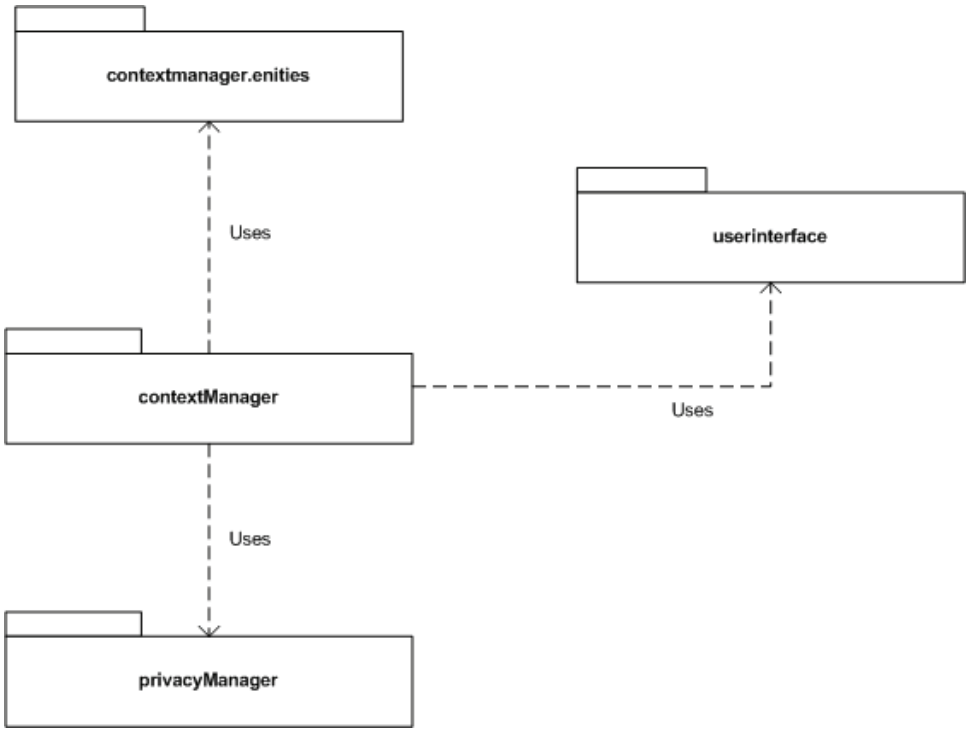


Figure 6-15: Package overview

6.5.2 contextmanger.entities

The package contextmanager.enitites is part of the Akogrimo framework. The classes which are used from this package are User, Context, RfidContext, SipPresence, SlpService and Position. These classes represent the user instances and the context instances. In addition Position is included to in order to compute the user's location.

6.5.3 The contextManager package

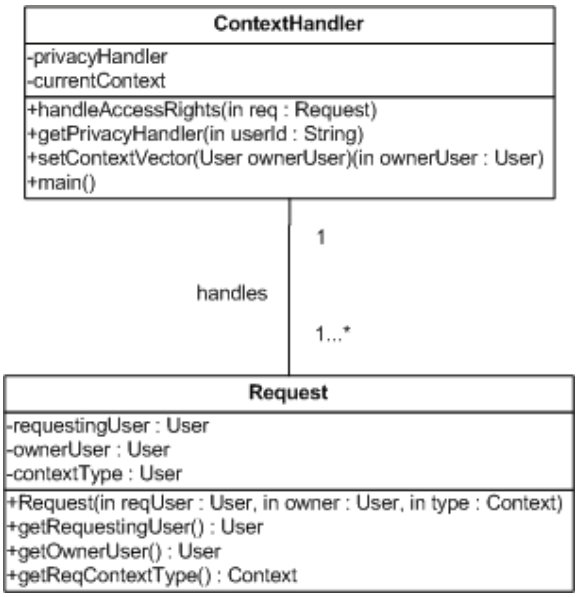


Figure 6-16: The contextManager package

The *contextManager package* implements the classes and methods which handle request/subscription messages. The package is included in the demonstrator in order to test the program. The *ContextHandler* class represents the part of the *Context Manager* which handles privacy enforcement when a request arrives or a subscription is relevant (i.e. a context change occurs and a consumer has a subscription to which the change has relevance). The *Request* class represents a request or a subscription. When a request/subscription is handled a object of the *Request* class is instantiated and the *handleAccessRights()* method in the *ContextHandler* is called. The *contextManager package* is a part of the *Context Handler* in Figure 6-6 and Figure 6-7.

- **Request**

An object from this class is instantiated when a request or subscription is handled.

  - *getRequestingUser()*
  - *getOwnerUser()*
  - *getReqContextType()*

- **ContextHandler**
  - *handleAccessRights(Request r)*

    i. The method retrieves the identity of the context owner, the identity of the context consumer and the requested context type from the Request object.

    ii. An object from the *PrivacyHandler* (see Figure 6-17) is instantiated with the context owner id as parameter.

iii. The *setContextVector(contextOwner)* method is called with the context owner id as parameter. This method returns a vector of the context owner's current context values.

iv. The *checkAccessRights()* method is called with the context consumer identity, the requested context type and the vector with the values of the current context of the context owner as parameters

v. The *getContextRights()* method is called which returns the response of the method call above. The response is either (true, eventual restrictions) or (false).

- *setContextVector(User ownerUser)*

  This method retrieves the context owner's current context parameter values and put them in a vector

6.5.4 The privacyManager package

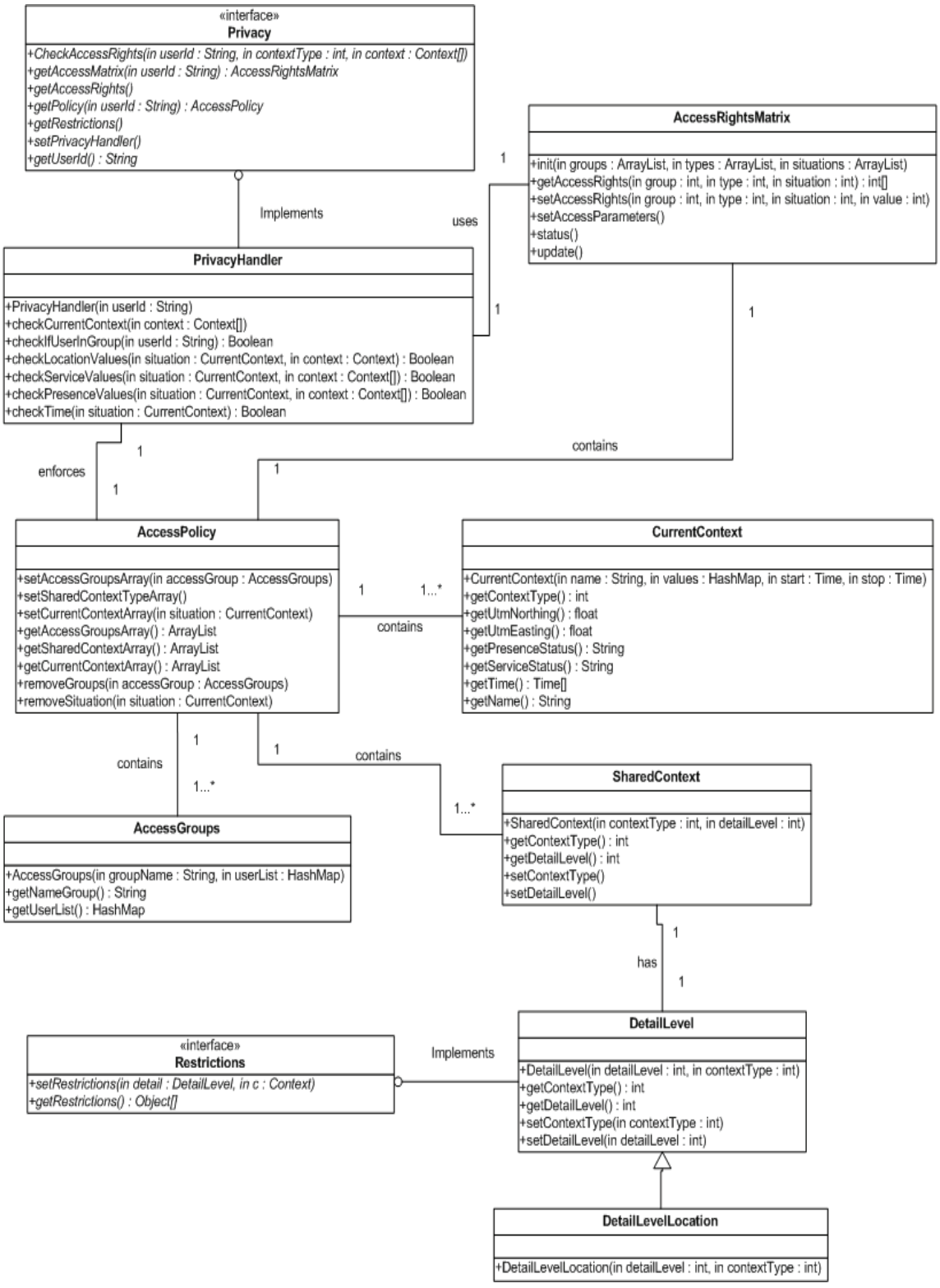


Figure 6-17: The privacyManager package

- **PrivacyHandler**

This class handles the methods that control if the parameters of the Request match the parameters of the access policy.

- checkAccessRights(String, int, Context[])

This method is called by the ContextHandler upon receiving a Request message (or Subscribe message). The parameters are a String which gives the context consumer identity, an Integer which gives the type of the requested context information and an array of the current context parameter value of the context owner. These parameters are further used in the following methods to verify if they match any of the user defined parameters:

- checkCurrentContext(Context[])

This method retrieves each element from the array of Context (these elements gives the current situation of the user) and runs a check if there are any user defined Situations that matches the current situation of the user. For each of the element the contextType decides which values that shall be controlled. The following methods are run accordingly.

- checkLocationValues(CurrentContext, Context)
- checkPresenceValues(CurrentContext, Context)
- checkServiceValues(CurrentContext, Context)

For each Situation the time of day is controlled against the time interval set by the user.

- checkTime(CurrentContext)

- checkIfUserInGroup(String)

This method takes the context consumer identity as a parameter and checks if the context consumer is part of any of the user defined groups.

- getRestricitions()

This method gets the restriction associated with the context type that will be shared. The method returns a value that will be returned by the *getAccessRights()* method if the result of the *checkAccessRights()* method is true.

- getAccessPolicy()
- This method retrieves the access policy for one user from a database (in the program the method reads an *AccessPolicy* object from file)
- getAccessRights()
- This method returns a boolean true if access is granted and the associated restrictions (if there are any restrictions). If access is denied the method returns a boolean false.

- **AccessPolicy**

- setAccessGroupsArray(AccessGroups)

When an AccessGroup is created this method adds the group to an ArrayList of all the groups.

- setShareContextTypeArray()

This method associates each type of context with the different *detailLevels* and instantiates an object of the *SharedContextType* class for each element. The *SharedContextType* instances are added to an ArrayList.

- setCurrentContextArray(CurrentContext)

When a situation is created the *CurrentContext* object is instantiated and added to an ArrayList of all these object instances.

- createAccessMatrix()

This method calls the *init()* method in *AccessRightsMatrix* class. The ArrayLists of *AccessGroups*, *CurrentContext* and *SharedContextType* are parameters of this method.

- **AccessRightsMatrix**

  - Init(ArrayList,ArrayList,ArrayList)

This method creates a three dimensional matrix where all the elements, representing the access rights, are set to zero.

  - setAccessRights(int,int,int,int)

This method sets one element in the matrix to the value of the last integer. The position in the matrix is given by the first three integers. These integers represents the position of AccessGroups, ContextType and Situation in the in the respective ArrayLists.

  - getAccessRights(int, int, int)

This method returns the access rights.

- **DetailLevel**

This class gives the methods to get and set the detail level of the context type.

The following classes are used to compute the abstraction for the different context types:

- DetailLevelLocation – only the detail level of location is implemented
- DetailLevelPresence
- DetailLevelService
- DetailLevelContactInformation

6.5.5 The userinterface package

This package handles the Gui interface towards the context owner. This is where the user fills in the parameters of the Privacy Policy. The interface is only included to give an idea of how the system would work and is not meant to be a suggestion on how to present the service to potential users. The user interface is presented in Appendix A-2.

6.6 Testing

To verify that the implemented demonstrator performs according to the requirements three different test sets were defined and executed in order to se if the output was as expected. In addition several tests on the functionality of the user interface were included.

6.6.1 Test environment

The testing was executed locally on one computer. The software used to both write and run the implementation was Eclipse.

6.6.2 Test results

A more detailed description of the tests and the test result can be found in Appendix B.

Table 6-6: Test description

Test	Description
T1	User input – editing access groups: Add Group
T2	User input – editing access groups: Remove Group
T3	User input – editing access groups: Add user to group
T4	User input – editing access groups: Remove user from group
T5	User input – editing situations: Create Situation
T6	User input – editing situations: Set time interval
T7	User input – editing situations: Set context values
T8	User input – editing situations: Delete Situation
T9	User input – set access rights
T10	User input – remove access rights
T11	Get status of given access rights
T12	Update access policy
T13	Test set 1: One context owner and one context consumer
T14	Test set 2: One context owner and several context consumers

Table 6-7: Test summary
P-passed, F-failed, NT-not tested, NA- not applicable

Requirement	Result	Test	Comment
S1, S2	P	T1	
S1, S2	P	T2	You have to exit the window before the group is removed from the list in the Gui
S1, S2	NA	T3	Not implemented
S1, S2	NA	T4	Not implemented
S2	P	T5	
S4	P	T6	
S4	P	T7	
S1, S2	NA	T8	Not implemented
S1, S2	P	T9	
S1, S2	P	T10	
S1,S2	P	T11	Not presented in Gui interface
S1,S2	F	T12	If new groups or situations are added the access rights which are set previously disappear. If only new access rights are added the old ones are intact.
S1, S2,S5	P	T13	
S1, S2,S5	P	T14	

6.6.3 Comment on test results

The focus in the implementation has been on the functionality of the privacy handling and not the user interface as this was considered to be out of scope of this assignment. Some of the tests that failed were due to lack of support in the user interface. The implemented functionality of the Privacy Handler was as expected.

7. Discussion

In this chapter the demonstrated Privacy Policy Enforcer (PPE) will be evaluated. To what degree the system meets the system requirements and the actual achievements in privacy enforcement will be discussed. In what way the system fulfils the design principles, which were pointed out in Section 4.4 will be looked into. The added value for the user and employment of the system will be discussed by looking at three scenarios of practical use.

7.1 Achievements

In this section the functionality the PPE system offers to the user will be discussed. It will be looked into how the presented solution fulfils the system requirements. The system requirements were defined in Section 6.2.

PPE: The proposed PPE system presents a solution on how to handle privacy enforcement for a user of context aware services. PPE offers three mechanisms to the user:

- Access control based on the identity of the context consumer
- Access control based on correspondence of the current context of the context owner and the situation where this context information should be shared
- Differentiating of the precision level of disseminated context information

By implementing PPE the user has the possibility to control the sharing of different types of context information connected to his or her person. The policy is based on both the identity of the context consumer and the current context of the context owner. Due to this the access rights of a context consumer will vary dynamically with the current context of the context owner. The policy also offers the possibility to differentiate a type of context information. This mechanism enables the user to choose between different levels of exposure of one type of context information. As a result the user does not have to deny or grant all access to a context type, but can instead reveal some information and hide the rest.

The System Requirements: The goal was to find a solution that restricts access to a user's context information. The solution should give the user the possibility to predefine a policy which the system should consult to enforce privacy. The access control should enable the system to give access to context information based on the user's current context and the context consumer's identity in addition to differentiate one type of context information into different detail levels. The context types that should be considered were location, presence, time and service information. To what degree the implemented demonstrator of the PPE system comply with the system requirements are summarized inTable 7-1.

Table 7-1: Fulfillment of the system requirements
C-complied, NC-not complied, PC-partially complied

Requirement	Description	Result	Comment
S1	The user shall be able to grant/deny other specific users/system entities access to his or her context information.	C	Implemented as specified
S2	The user shall be able to define access to his or her context based on the context consumer's id, his or her current context and the type of context (see Figure 2-1) that will be shared.	C	Implemented as specified
S3	The system should require a minimum of user input from the context owner.	PC	It is hard to argue for an absolute standard for what is a minimum of user input. The requirement is complied in the sense that the user does not have to provide input frequently.
S4	Context information that should be taken into account when deciding access rights are: • presence (occupation) • localisation • time of day • available services	C	Implemented as specified
S5	The user should be able to specify the accuracy of the context information to which the context consumer is granted access. (building vs. room etc)	PC	The differentiating mechanism is only implemented for location information

As seen in Table 7-1 the differentiation of context information is only realized for location information. The relevance of differentiating information of other context types, such as presence and service information was found to be small. Service information might be relevant to differentiate in some situations (e.g. only share some services available to the user and not all). The implemented solution does not present a way of organizing the differentiation of presence and service information, but the design opens for later inclusion of this functionality.

7.2 Coverage of the privacy design principles

In Section 4.4 the privacy mechanisms that should be included in the PPE system were identified. It was discussed which design principles presented by Langheinrich [23] that would be covered as a result of implementing these mechanisms. These principles were notice, choice and consent. In this section it will be discussed how the PPE design and implemented solution comply with these design principles.

7.2.1 Notice

The principle of notice states that the context owner should be made aware of possible collection of context information. If a user has implemented PPE, only consumer identities which are pre-approved can collect information. The context owner will therefore always be aware of which context consumers that have the possibility to collect information. Based on this solution the principle of notice is considered to be covered.

7.2.2 Choice and consent

The principle of choice and consent states that the user should have to give his or her consent in order to enable a consumer to collect context information. The possibility to decline the request should also be present.

Due to the same mechanism which secures notice, choice and consent will also be covered. The context owner chooses to give his or her consent to data collection by adding an identity to an Access group. To give the user an actual choice to turn down the request is partially covered. The user can choose not to share information with an entity, but if the information is required to for instance entering a building, the only choice might be to stay outside. An alternative could be to use the differentiating mechanism to present a less detailed set of context information.

To what degree the PPE system presents a satisfying solution on how to give consent depends on the character of the communication path. If the context consumer is a user wondering about a colleague's whereabouts, giving consent in advance might be a satisfactory solution. However, if the request is initiated in order to offer a service in a pervasive computing environment, the context owner does not know the identity of the consumer in advance. It is therefore not possible to give consent before a request is received. A mechanism which allows the user to choose to accept such a request is discussed in Section 6.4.6. This is not implemented as a part of the final solution. The unknown identity will here be added to a list of subscribers which the context owner can later add to a group.

7.3 User perspectives

In this section the added value PPE brings to the user of a context aware service will be discussed. Does the solution give sufficient flexibility without placing too high demands on user input? Is the system too complex for an unskilled user? How often does the user have to change the privacy setting parameters?

One of the goals with the PPE system was to present a flexible privacy enforcement system to the user without adding a large amount of extra demands for administration needs. The

system should not be so complex that the user does not bother to use it or does not keep the privacy policy up to date.

The user's privacy preferences vary and the perception of what is too much demand on user input differs. A possible solution to this problem was presented in Section 6.4.6. A provider of a context management system can present different variants of configuration possibilities of PPE. The variants differ in the degree of predefined privacy settings set by the service providers. A version that requires a minimum of user input would give the user a less flexible and personalized service, but the user would get away with less administrative work. The implemented demonstrator represents a version where differentiating values are set by the provider of the system (i.e. the abstraction resulting from choosing the different details levels are predefined). A version in the other end of the scale would give the user the possibility to set the values of the detail levels manually. The user could define that location with detail level two would reveal the town she/he is in. In a predefined version detail level two might give the building.

The demand on the frequency of user input to the system depends on how detailed the user wants to control the access to context information. User input in this context is the messages sent from the user to the PPE system in order to create or update the privacy policy. The sequence of message exchange is illustrated in the MSC in Figure 6-9. Independent of the version of PPE, the user can choose to create several groups and lots of different situations or the user can create fewer groups and situations. The most natural way of configuring the system would probably be for the user to create a set of groups and a set of situations which will be more or less static and then set or remove access rights more frequently, and possibly add or remove context consumers from the groups and change the values of the situation parameters.

The system offers dynamic change of access rights as they will change with the context owner's current context. This flexibility, when it comes to the user's needs for privacy in different situations, would probably be satisfactory as long as the user does not change habits drastically. The element that represents the greatest need of altering the privacy settings is new consumer identities, especially if consumer identities represents all system entities and not only human users. In a pervasive environment unknown identities would probably pop up quite often and a mechanism to handle those should be implemented.

7.4 Employment of the PPE system

The solution presented with the PPE on how to control access to context information could be relevant to use in several situations. In Section 1.2 a scenarios were presented which described how a person in a job situation profited from the use of a context-aware service. The scenario also illustrated the importance of a privacy enforcement mechanism to fully exploit the potential of the context-aware service. In this section another scenario will be presented in order to demonstrate how the privacy problems experienced can be solved with the PPE. The goal is to find out to what degree employment of PPE will contribute to better utilization of the context system and give a better user experience.

7.4.1 Student scenario

Lisa is a medical student who spends her school days both at the university either in lectures or the study rooms, and at the hospital. Both the students and the employees at the university and at the hospital use a context aware application which enables them to access information about their colleagues' and friends' whereabouts and occupations. All the students have their regular days when they work at the hospital. With this system it is always easy to find out who is available and where they are. The system can also be used outside the hospital and campus area. The users of the system can choose to log out when they leave the hospital or the campus area for the day, but it is still possible to benefit from the services offered by the system. Lisa finds the system quite convenient to use both during the school day and in her spare time. However, she normally logs off when the school day is over because she does not like the idea that all her friends and the employees at the hospital can see where she is all the time.

7.4.2 Using the PPE system

By using the PPE system to control the access to her context information Lisa can easily avoid the problem mentioned above. With the PPE Lisa organizes all her contacts in four groups; Friends, Family, Employees at hospital and Employees at university. She also defines three situations based on her presence status, location and the time of the day. When she is in the hospital or the university from 9am-5pm and her presence status is "Available" all the groups can access her context information. If her presence status is "Occupied" only the employees at the hospital and the university can access information about her. When her location is outside the campus area and the time outside the time interval, 9am-5pm, only her friends and family can access her context information.

As we see by using the PPE system Lisa avoids the problem of being "watched" all the time without having to log on and off the system. In this way she can exploit the advantages the system is offering both during school hours and in her spare time. She also avoids the trouble of constantly having to change the settings of the system. This is handled automatically once she has created the groups and situations. Of course there might be changes to her privacy preferences, but this does not happen several times a day. Consequently by using this system Lisa will have her privacy when she does not want to be disturbed, and she will be visible to her friends when she is ready for a break. For Lisa the two big advantages of using this system are; one; she does not have to configure the system each time her situation changes, two; she can trust that she knows exactly who can access her context information at all times.

7.4.3 Discussion – Added value compared to solutions used today

As we see from these three scenarios the PPE system enables the users to better exploit the potential of the context-aware service.

The ability to assign different access rights to different users makes it possible to benefit from a context-aware service, similar to the described services, in situations where you do not want everyone to be able to see you.

A service used by lots of people today is MSN Messenger. This is a service which Lisa probably would use in her situation to easily get in touch with her friends at the university. The system allows you to "block" people you do not want to contact you, either permanently or periodically. However, it is not possible to block a whole group of people at the same time, and if you block a person the person remains blocked until s/he is unblocked manually by the

user. The information revealed in such system is minimal as other user's only see that you are logged on to the Internet somewhere in the world, but if the system where to include more information a more flexible solution would be useful.

The Privacy Policy solves the two drawbacks of this system. Access can be denied or granted to groups of users and the access rights changes dynamically with the user's situation. The Privacy Policy system also considers other types of information than presence status.

8. Conclusion

The goal of a pervasive environment is for the computing system to adapt dynamically to the user's needs. To exploit the full potential of pervasive computing environments, information about all system entities has to be collected and further disseminated. This information is often highly sensitive information and as discussed throughout this report, this result in privacy issues which have to be dealt with in order for such a system to be used.

The first part of this master thesis consisted of a literature study to identify which privacy issues that should be handled in a pervasive environment. The privacy issue identified as the most crucial was to handle the control of access to context information. The second part of this thesis investigated how access to a user's context information can be controlled. Different technologies and architectures have been studied in order to create a solution on how to enforce privacy in a flexible and differentiable manner.

The solution that is proposed is the Privacy Policy Enforcer (PPE). The scenarios which are outlined in Section 1.2 and further discussed in Section 7.4, show that by applying the PPE system in a context management system, the potential of a context aware service can be better exploited. Through the use of the PPE system a more dynamic control of access to context information is achieved. As shown in the scenarios, without any privacy enforcement mechanism the alternative would be to turn the system off to avoid sharing of context information with certain people.

In addition to controlling the most basic demand for access control (i.e. if a certain entity in the system has access or not) one of the objects of this master thesis was to find a solution that was both flexible and opened for differentiating possibilities (i.e. abstract certain details away from the context information which is presented to consumer entities). The PPE system controls access to context information based on the context owner's situation in addition to the context consumer's identity. This enables the system to change access rights to certain entities when the context owner's situation changes, a solution which gives the system flexibility. The user can specify by detail level if any details of the different context type parameters should be hidden from certain context consumers. In this way the system can differentiate the context information that is revealed to different context consumers.

8.1 Future work

The implemented version of the PPE system does not include a mechanism to handle new consumer identities. A solution which handles this should be a part of a complete system. Future work would include considering a way to make the system capable of determining which new identities that should be allowed access or possibly which consumer request that should result in a notification of the user.

Another issue to consider is how the system could be used to set a privacy policy on behalf of a context owner. Future work would include considering which actors that should handle this and how privacy can be secured in the relation ship between this actor and the context owner.

In this thesis only dissemination of context from the context manager to the context

consumers were considered. Privacy issues concerning the gathering of context information should be considered in a complete context management system. Future work would include considering the different mechanisms for collecting context information in order to find out how these could get consent from the user to collect information and a way to disable the collecting mechanism for certain users if the collection is refused.

References

[1] Akogrimo, Project web site: http://www.akogrimo.org/ (accessed June 2006)

[2] Ailisto, H., Alahuhta, P., Haataja, V., Kyllönen, V., Lindholm, M., *Structuring Context Aware Applications: Five-Layer Model and Example Case,* VTT Electonics (2002) http://www.comp.lancs.ac.uk/computing/users/dixa/conf/ubicomp2002-models/pdf/Ailisto-Ubicomp%20Workshop8.pdf

[3] Anagnostou, M. E., Juhola, A., Sykas, E. D., *Context Aware Services as a step to Pervasive Computing*, http://context.upc.es/Papers/ContextLobster.pdf

[4] Baldauf, M., Dustdar, S., Rosenberg, F., *A Survey on Context-Aware Systems.* Distributed Systems Group Information Systems Institute Vienna University of Technology (2004)

[5] Bing, J., *Personvern i faresonen*, Cappelens Forlag, Oslo (1991)

[6] Chen, H., Finin, T., *An Ontology for Context Aware Pervasive Computing Environments*, Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review (2003)

[7] Chen, H., *An Intelligent Broker Architecture for Pervasive Context-Aware Systems*, PhdThesis, University of Maryland, Baltimore County (2004)

[8] Cheng, H.C., Zhang, D., Tan, J. G., *Protection of Privacy in Pervasive Computing Environments*, International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II, pp. 242-247 (2005)

[9] Cobb, M. *Privacy vs. security.* Advisor: http://www.advisor.com/Articles.nsf/aid/COBBM73 (accessed fall 2005)

[10] Daidalos Pervasive Systems Privacy and Security Framework and Mechanisms, Deliverable D421, Daidalos Consortium 2004 (2004)

[11] Datatilsynet: http://www.datatilsynet.no/templates/Temaforside____191.aspx, (accessed May 2006)

[12] Datatilsynet, Personvernsrapporten 2006, http://www.datatilsynet.no/templates/Page____1428.aspx (accessed June 2006)

[13] Dey, A. K., *Understanding and using context*, Future Computing Environments Group, College of Computing & GVU Center, Georgia Institute of Technology, pp 92-99, Personal and Ubiquitous Computing, Volume 5 , Issue 1, pp4 - 7 (2001)

[14] Dourish, H., Grinter,R. E., Delgado de la Flor, J., Joseph, M., *Security in the wild: User Strategies for Managing Security as an Everyday, Practical Problem,* Personal and Ubiquitous Computing, Volume 8, Issue 6, pp 39-401 (2004)

[15] Egeland, S. B., *Infrastructure to communicate context information*, Project 5th year, Department of Telematics, the Norwegian University of Science and Technology (2005)

[16] Gandon, F. L., Sadeh, N. M., *Semantic Web Technologies to Reconcile Privacy and Context Awareness*, School of Computer Science – Carnegie Mellon University, Journal of Web Semantics, Volume 1, Issue 3 (2004)

[17] Hauknes, C., *User-centered Privacy Aspects in Connection with Location Based Services*, Complex:Oslo (2003)

[18] Henrickesen, K., Indulska, J., Rakotonirainy, A. *Modeling Context Information in Pervasive Computing Systems.* Proceeding of Pervasive 2002 (Zürich, Switzerland): pp167-180 (2002)

[19] Henricksen, K., Wishart, R., McFadden, T., Indulska, J., (2005).*Extending Context Models for Privacy in Pervasive Computing Environments*. Proceedings of the 3rd Int'l Conf. on Pervasive Computing and Communications Workshop (2005)

[20] Introduction to Dataveillance and Information Privacy, and Definitions of Terms:http://www.anu.edu.au/people/Roger.Clarke/DV/Intro.html (accessed May 2006)
[21] Jendricke, U., Kreutzer, M., Zugenmaier, A., *Pervasive Privacy with Identity Management*,
[22] Jiang, X. and Landay, J. *Modeling Privacy Control in Context-aware Systems Using Decentralized Information Spaces*, IEEE Pervasive Computing 1(3) (2002)
[23] Langheinrich, M. , *Privacy by design – Principles of Privacy-Aware Ubiquitous Systems*, Ubicomp Proceedings (2001)
[24] Langheinrich, M., *Privacy Invasions in Ubiquitous Computing*, Ubicomp Privacy Workshop (2002)
[25] Langheinrich, M., *Personal Privacy in Ubiquitous Computing – Tools and System Support*, Submission for a Doctor degree in Science ,Swiss Federal Institute of technology Zurich (2005)
[26] Langheinrich, M., *A Privacy Awareness System for Ubiquitous Computing Environments*, 4th International Conference on Ubiquitous Computing (Ubicomp 2002), pp. 237-245 (2002)
[27] Marx, G.T., *Murky Conceptual Waters: the Public and the Private*, Ethics and Information Technology, Vol. 3, no. 3, pp. 157-169, (2001)
[28] Mostéfaoui, S. K., Mostéfaoui, G. K., *Towards a Contextualisation of Service Discovery and Composition for Pervasive Environments*, Proceedings of the Workshop on Web-services and Agent-based Engineering (WSABE) (2003)
[29] Osland et al., *Enabling context-aware applications* (2006)
[30] Pervasive computing 2001, National Institute of Standards and Technology: http://www.nist.gov/pc2001/ (accessed May 2006)
[31] Privacy/Data Protection Project, *Privacy and confidentiality*: http://privacy.med.miami.edu/glossary/xd_privacy_basicdef.htm (accessed May 2006)
[32] Ravlund, I., *Setter vår lit til storebror… og alle småbrødre med*, TØI rapport 789/2005, TØI, Oslo (2005)
[33] Satyanarayanan, M., *Pervasive computing*, IEEE Personal Communications, Vol. 8, No. 4, pp.10-17 (2001)
[34] Schilit, B.,Theimer, M., *Disseminating Active Map Information to Mobile Hosts*, IEEE Network, 8(5) pp22-32 (1994)
[35] Telenor, Akogrimo_D.4.2.1, *Overall Network Middleware Requirements Report Version 1.0.* (2005)
[36] Tometa Software, http://www.tometasoftware.com/rational_unified_process.asp (accessed May 2006)
[37] TRUSTe: http://www.truste.org/
[38] Warren, S., Brandeis, L., *The right to privacy*, Harvard Law Review, 4:193 – 220 (1890)
[39] Weiser,M. Ubiquitous Computing: http://www.ubiq.com/hypertext/weiser/UbiHome.html (accessed May 2006)
[40] Westin, A., *Privacy and Freedom*, Atheneum, New York (1967)
[41] W3C - http://www.w3.org/TR/P3P11/#Introduction (accessed June 2006)
[42] Østhus, E., *SIP session management in pervasive environments*, Master Thesis, Norwegian University of Technology and Science (2005)

Appendix A: Delivery details

A-1 Setup

1. Run Eclipse and choose the catalogue PPE as workspace

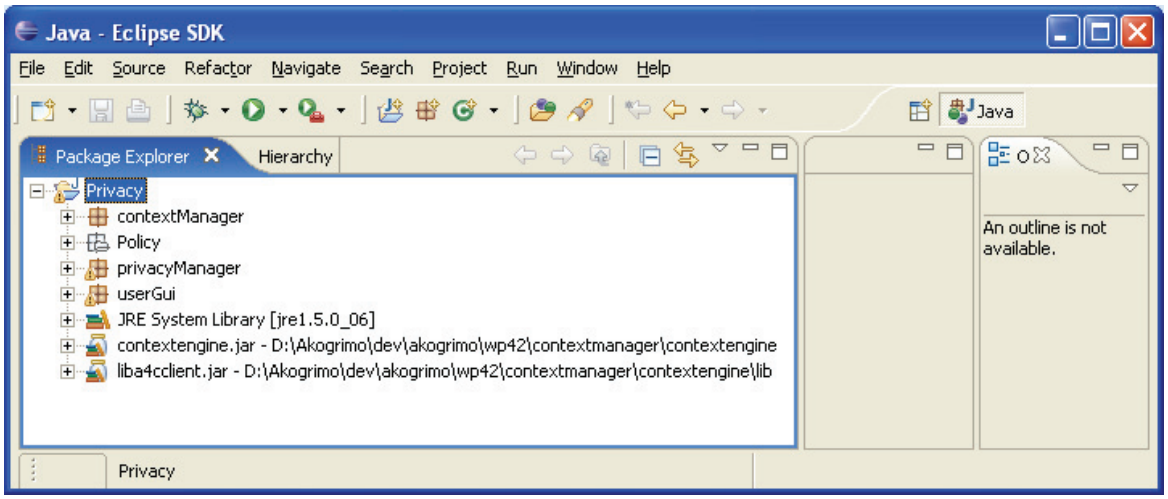


Figure A-0-1: Screen shot of Eclipse Workspace

If the window looks like this the program is ready for execution. If the package explorer field is empty the Privacy project has to be imported into the workspace. Chose: File->import->Existing Projects into Workspace.

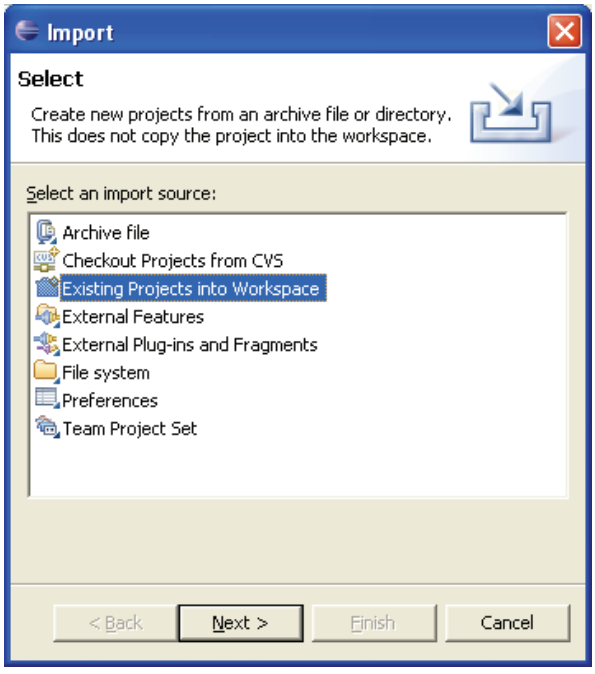


Figure A-0-2: Screen shot: Import of existing project into Eclipse Workspace

A-2 Execution

The program is delivered with a ready privacy policy for the user Tom, tom@akogrimo.org. The policy is saved as three objects in the Policy catalogue in the workspace catalogue (PPE/Privacy/Policy).

Table A-1: The enclosed Privacy Policy

Shared Context / Situation	Type: Location Detail level: 1	Type: Location Detail level: 2	Type: Service Detail level:	Type: Presence Detail level: 1
Home	Family	Friends	Family	Colleagues
Work	Family Colleagues		Colleagues	Friends

Table A-1 shows the access rights which are set in the enclosed privacy policy. The parameter values are described in the test details in Appendix B.

A-2-1 Execution in order to test the existing privacy policy:

I. Run the ContextHandler

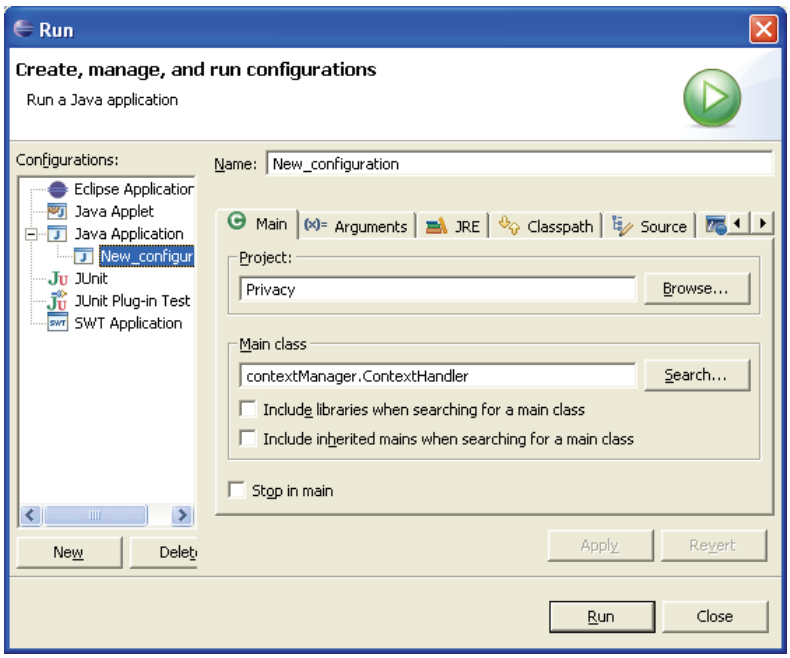


Figure A-3: Run the program in Eclipse

As the program is not connected to a context manager system yet the following code was written to test how the requests were handled by the privacy manager. The code creates one context owner and two context consumers. The context consumers each send three requests.

```
public ContextHandler(){

            //context owner
            User tom= new User("tom@akogrimo.org", "Tom" );

            //context consumers
            User alice= new User("alice@akogrimo.org", "Alice" );
            User bob= new User("bob@online.no", "Bob");

            //requested context
            Context reqContext= new Context(Context.RFID_LOCATION,"");
            Context reqContextTwo= new Context(Context.SIP_PRESENCE,"");
            Context reqContextThree= new Context(Context.SLP_SERVICE,"");

            //the context owner's current context - a vector
            setContextVector(tom);

            //oppretter request for alice
            Subscription_request req= new
            Subscription_request(alice,tom,reqContext);
            handleAccessRights(req);
            Subscription_request reqTwo= new
            Subscription_request(alice,tom,reqContextTwo);
            handleAccessRights(reqTwo);
            Subscription_request reqThree= new
            Subscription_request(alice,tom,reqContextThree);
            handleAccessRights(reqThree);

            //create req for Per
            Subscription_request reqPer= new
            Subscription_request(bob,tom,reqContext);
            handleAccessRights(reqBob);
            Subscription_request reqTwoBob= new
            Subscription_request(bob,tom,reqContextTwo);
            handleAccessRights(reqTwoBob);
            Subscription_request reqThreeBob= new
            Subscription_request(bob,tom,reqContextThree);
            handleAccessRights(reqThreeBob);
```

Expected output:

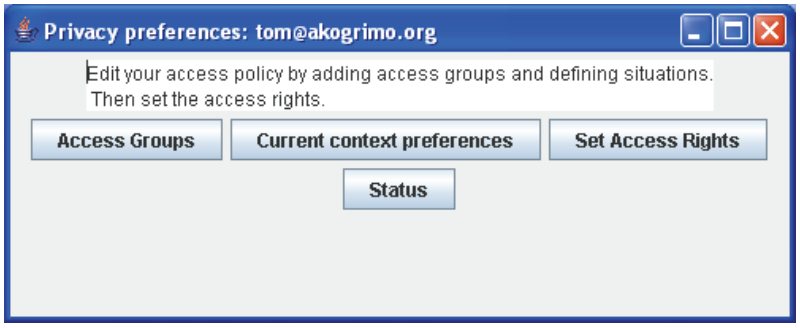
```
Checks access for: alice@akogrimo.org
Access granted from contexthandler: true
Checks access for: alice@akogrimo.org
Access granted from contexthandler: false
Checks access for: alice@akogrimo.org
Access granted from contexthandler: false
Checks access for: per@domain.no
Access granted from contexthandler: true
Checks access for: per@domain.no
Access granted from contexthandler: false
Checks access for: per@domain.no
Access granted from contexthandler: true
```

A-2-2 Execution in order to create a new privacy policy and test these:

II. Remove the files in the catalogue //PPE/Privacy/Policy

III. Run the UserInput in order to create the privacy policy
The objects will be written to files which will be placed in the catalogue //Privacy/Policy
The following steps are included in creating the Privacy Policy:

1.

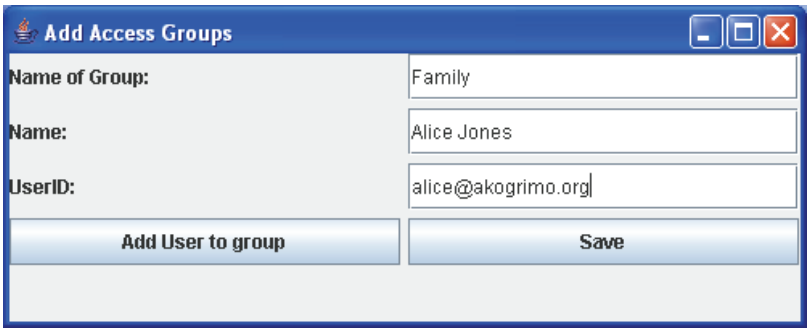


When UserInput is executed this window pops up. Press a button to choose which p parameters to fill inn.

2.

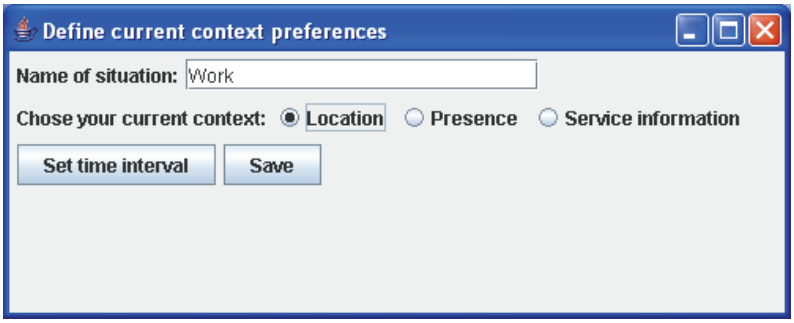
Press add group.

3.



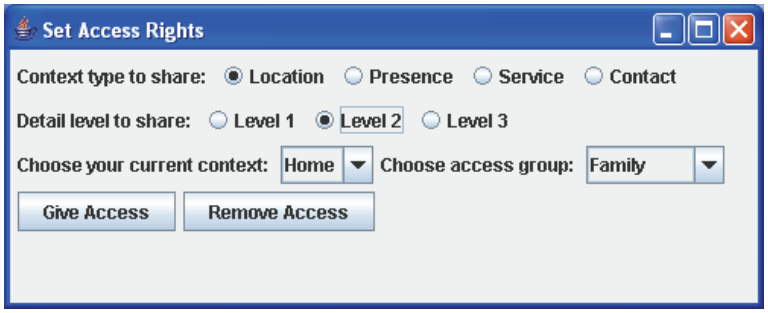
Then you get this window. Fill inn name of group and add users.

4.



When you press the “Current context preferences”-button in the first window you get the following window. Fill in name and check the context information you want to define this situation. When you check each of them a new window will pop up where you can fill in values.

5.



When you press the “Set access rights”-button in the first window you get the following window. Here you can add or remove rights based on the other parameters you have filled in.

I. Run the context handler

To request has to be altered in the constructor of the ContextHandler if other requests shall be tested.

Appendix B: Test details

B-1 Input

Input from context owner:
User name: Tom
User ID: tom@akogrimo.org

Privacy Policy:

- **Access groups:**

1. Family: Alice – alice@akogrimo.org, Bob – bob@online.no, Sam- sam@online.no
2. Friends: Peter –p@dom.no
3. Colleagues: Per – p@dom.no, Lisa – l@dom.no

- **Current context:**

8.1.1.1 ***Home*** *– Time interval: 5pm – 11pm*

Context Type	Values	
Location	Northing	10 000
	Easting	20 000
Presence	Status	"Available"
Service Information	Type of service	"WLAN"

8.1.1.2 ***Work*** *– Time interval: 8am – 4pm*

Context Type	Values	
Location	Northing	300 000
	Easting	400 000
Presence	Status	"Occupied"
Service Information	Type of service	"WLAN"

- **Shared context:**

Context type	Detail level	
Location3	Level 1	Radius = 0m
	Level 2	Radius = 300m
	Level 3	Radius= 1000m
Presence	Level 1	Not defined
	Level 2	……
	Level 3	……
SLP service	Level 1	……
	Level 2	……
	Level 3	……

[3] See Section 5.1.4 for computation of Location values.

- **Access Rights:**

Shared Context / Situation	Type: Location Detail level: 1	Type: Location Detail level: 2	Type: Service Detail level: 1	Type: Presence Detail level: 1
Home	Family	Friends	Family	Colleagues
Work	Family Colleagues		Colleagues	Friends

Input from context consumer 1:
Situation: one provider – one consumer

Request(alice@akogrimo.org, tom@akogrimo.org, location)
Request(alice@akogrimo.org, tom@akogrimo.org, presence)
Request(alice@akogrimo.org, tom@akogrimo.org, service)

Input from context consumer 2:
Situation: one provider – several consumers

Request(alice@akogrimo.org, tom@akogrimo.org, location)
Request(alice@akogrimo.org, tom@akogrimo.org, presence)
Request(alice@akogrimo.org, tom@akogrimo.org, service)
Request(peter@akogrimo.org, tom@akogrimo.org, location)
Request(peter@akogrimo.org, tom@akogrimo.org, presence)
Request(peter@akogrimo.org, tom@akogrimo.org, service)
Request(per@akogrimo.org, tom@akogrimo.org, location)
Request(per@akogrimo.org, tom@akogrimo.org, presence)
Request(per@akogrimo.org, tom@akogrimo.org, service)

B-2 Tests

T1, T2, T3, T4 – Editing Access Groups

User input:

a. Add group
The test was successfully completed
b. Remove group
The test was successfully completed
c. Add/remove user to/from existing group
This is not implemented

T5, T6, T7, T8 – Editing Situations

a. Create situations:
   i. Set time
   ii. Set values

This was successfully completed

b. Delete situations

This is not implemented

T9, T10 – Set access rights

a. Set rights

Result:

```
Context Type: 2 Detail: 1 Situation: Home Group: Family Rights: 1
Context Type: 2 Detail: 1 Situation: Work Group: Family Rights: 1
Context Type: 3 Detail: 1 Situation: Home Group: Family Rights: 1
Context Type: 1 Detail: 1 Situation: Work Group: Friends Rights: 1
Context Type: 2 Detail: 2 Situation: Home Group: Friends Rights: 1
Context Type: 1 Detail: 1 Situation: Home Group: Colleagues Rights: 1
Context Type: 2 Detail: 1 Situation: Work Group: Colleagues Rights: 1
Context Type: 3 Detail: 1 Situation: Work Group: Colleagues Rights: 1
```

b. Remove rights

Result:

```
Context Type: 2 Detail: 1 Situation: Home Group: Family Rights: 1
Context Type: 2 Detail: 1 Situation: Work Group: Family Rights: 1
Context Type: 3 Detail: 1 Situation: Home Group: Family Rights: 1
Context Type: 1 Detail: 1 Situation: Work Group: Friends Rights: 1
Context Type: 2 Detail: 2 Situation: Home Group: Friends Rights: 1
Context Type: 1 Detail: 1 Situation: Home Group: Colleagues Rights: 1
Context Type: 2 Detail: 1 Situation: Work Group: Colleagues Rights: 1
```

c. Add rights

Result:

```
Context Type: 2 Detail: 1 Situation: Home Group: Family Rights: 1
Context Type: 2 Detail: 1 Situation: Work Group: Family Rights: 1
Context Type: 3 Detail: 1 Situation: Home Group: Family Rights: 1
Context Type: 1 Detail: 1 Situation: Work Group: Friends Rights: 1
Context Type: 2 Detail: 2 Situation: Home Group: Friends Rights: 1
Context Type: 1 Detail: 1 Situation: Home Group: Colleagues Rights: 1
Context Type: 2 Detail: 1 Situation: Work Group: Colleagues Rights: 1
Context Type: 3 Detail: 1 Situation: Work Group: Colleagues Rights: 1
```

T11 – Get status

Get a table of current given rights

```
Context Type: 2 Detail: 1 Situation: Home Group: Family Rights: 1
Context Type: 2 Detail: 1 Situation: Work Group: Family Rights: 1
Context Type: 3 Detail: 1 Situation: Home Group: Family Rights: 1
Context Type: 1 Detail: 1 Situation: Work Group: Friends Rights: 1
Context Type: 2 Detail: 2 Situation: Home Group: Friends Rights: 1
Context Type: 1 Detail: 1 Situation: Home Group: Colleagues Rights: 1
Context Type: 2 Detail: 1 Situation: Work Group: Colleagues Rights: 1
Context Type: 3 Detail: 1 Situation: Work Group: Colleagues Rights: 1
```

This was not implemented in the Gui interface. The status method which prints all the access rights works.

T12 – Update access policy

It should be possible to update the access policy:

d. Are the existing groups and situations still part of the list?
The existing groups and situations are still part of the list when new objects are added.
e. Is the access right correct?
When groups or situations are added the access rights which are set previously are removed. To only update the access rights was successfully tested.

Comment: This test partially failed due to error in the update method in the matrix class.

T13 – Input from one context owner and one context consumers

Test set 1: *Input from context owner 1 + Input from context consumer 1*

Comment: In this test scenario one user is creating a privacy policy with the input parameters described in and one user is sending three requests.

```
Checks access rights with parameters: alice@akogrimo.org, 2, 21
Restrictions:
      Max east, north: 400000.0, 300000.0.
      Min east, north: 400000.0, 300000.0
Access granted from context handler: true

Checks access rights with parameters: alice@akogrimo.org, 1, 21
Access granted from context handler: false

Checks access rights with parameters: alice@akogrimo.org, 3, 21
Access granted from context handler: false
```

1. Output when context information is requested:
2.

If true:

i. Are the restrictions correct?
The location restriction is correct. This is the only restriction which is implemented.

ii. Is the user in the situation at the moment?
The access granted complies with the situation of the user. The test was performed at 14.00 when the context owner was in the situation “work”.

3. Does the answer match the assigned rights?

The answer did match the assigned rights.

T14 – Input from one context owner and several context consumers

Test set 2: *Input from context owner 1&2 + Input from context consumer 2*

```
Checks access rights with parameters: alice@akogrimo.org, 2, 21
Access granted from context handler: true
Restrictions:
      Max east, north: 400000.0, 300000.0
      Min east, north: 400000.0, 300000.0

Checks access rights with parameters: alice@akogrimo.org, 1, 21
Access granted from context handler: false

Checks access rights with parameters: alice@akogrimo.org, 3, 21
Access granted from context handler: false

Checks access rights with parameters: peter@domain.no, 2, 21
Access granted from context handler: false

Checks access rights with parameters: peter@domain.no ,1 , 21
Access granted from context handler: false

Checks access rights with parameters: peter@domain.no ,3 , 21
Access granted from context handler: false

Checks access rights with parameters: per@domain.no ,2 , 21
Access granted from context handler: true
Restrictions:
      Max east, north: 400000.0, 300000.0
      Min east, north: 400000.0, 300000.0

Checks access rights with parameters: per@domain.no, 1, 21
Access granted from context handler: false

Checks access rights with parameters: per@domain.no, 3, 21
Access granted from context handler: true
```

1. Output when context information is requested:

   If true:
   i. Are the restrictions correct?
   ii. Is the user in the situation at the moment?

2. Does the answer match the assigned rights?

Comment: Access to presence information was not granted as it should be. The rest of the test was passed.