FORMALISM OF PRIVACY PRESERVING ACCESS CONTROL

A DISSERTATION SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

2010

By
Naikuo Yang
School of Computer Science

Contents

Word Count 57565

List of Tables

List of Figures

Abstract

There is often a misalignment between requirements for keeping data owners' information private and real data processing practices, and this can lead to violations of privacy. Specifying and implementing appropriate policies to control a user's access to a system and its resource is critical for keeping data owners' information private. Traditionally, policy specification is isolated from requirements analysis, which often results in data processing practices that are not in compliance with data owners' requirements.

This thesis investigates a development scheme that integrates policy specification into requirements analysis and approach design. It suggests that, while we derive specification from requirements analysis, we can also improve requirements and approach design through privacy preservation specification by clarifying ambiguities in the requirements and resolving inconsistencies between requirements and data processing practices. This claim is supported by the requirements analysis and specification of a purpose based access control approach for privacy preservation.

The purpose-based access control method consists of an entity of *purpose*, which expresses requirements for keeping personal information private from a data owner's point of view. The requirements analysis is helped by the specification of the entities, the relationships, the invariants corresponding to the requirements, and the model operations along with proof obligations of their satisfiability. That specification results in a complete purpose based access control model in the case of an intra-organisation scenario. The development scheme has also been applied for privacy preservation in distributed collaborative environments. Distributed computing environments pose further challenges for keeping personal information private. Design considerations are taken for ensuring that personal information is accessed from two or more parties only if agreed privacy policies and privacy preferences are satisfied, and for facilitating privacy policies matching and privacy preference compliance among distributed collaborative organisations.

The work presented in this thesis should be of value to researchers on privacy protection methods, to whom the purpose-based access control model has been made available for privacy property verification, and to researchers on privacy specification, who will be able to incorporate specification into the requirements analysis.

Declaration

No portion of the work referred to in this dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Acknowledgements

This thesis would not have been possible without the help and support of many people over many years. I must begin by thanking my PhD supervisors: Professor Howard Barringer and Dr. Ning Zhang. They taught me how to initiate and develop the ideas into realistic goals. They encouraged and supported me throughout my PhD. They not only taught me how to be a good researcher, but also taught me how to be a better person. I feel very fortunate and appreciative to have them as my supervisors. I would like to thank Dr. Chris Kirkham for agreeing to be my project advisor. I also would like to thank Professor Dov Gabbay and Dr. David Rydeheard for their valuable comments and suggestions on the early version of this thesis.

The work presented in this thesis would not have been possible without the financial support by the UK-China Scholarships for Excellence from the China Scholarships Council (CSC) and the UK Department of Innovation, Universities and Skills (DIUS). Their support is greatly appreciated.

During my time in the School of Computer Science at the University of Manchester, I have always had the good fortune to be exposed to ongoing suggestions from both the members of staff and the other students. I am genuinely appreciative for all their help and assistance. I also wish to thank the colleagues with whom I shared the offices IT301 and KB2.110 and had interesting discussions in common room, among them are: Dr. Juan Antonio, Dr. Joachim Baran, Dr. Ioannis Ntalamagkas, Mr. Elton Ballhysa, Mr. Ye Zhang, and Dr. Zhixin Yu, .

I also would like to thank my wife's parents for supporting me to pursue my PhD overseas. They both left us in 2009. May they rest in peace in heaven.

My grandparents and parents, who raised me to be the person I am today, deserve the most acknowledgement. I thank my sister and my brother for their support throughout this process and beyond. Last but definitely not least, I thank my beloved wife, Hongfen, for understanding, supporting, and loving me, and our lovely daughter, Niuniu, for the happiness she brings to us.

Chapter 1

Introduction

The research in this thesis is aimed at gaining data owners' trust and improving their confidence in the disclosure of their personal information. Privacy is an important issue in information systems, but it is particularly vulnerable in infrastructure systems such as medical and financial information systems. Specifying and implementing appropriate policies to control a data user's access to a system and its resources is critical for protecting data privacy. In order to gain the data owners' trust and improve their confidence, it is important to fulfill the data owners' requirements upon the collection and usage of their personal information, as advocated in this thesis. It is also important to maintain consistency between the privacy preserving promises and the real data processing practices of organisations which collect and process such personal information. However, there has been little reported work in formally specifying privacy policies for information systems [FH01]. Additionally, distributed computing environments pose further challenges on privacy preservation. In this work, for privacy protection approaches, the primary focus is to take the data owners' privacy preferences into consideration, which are often neglected in previous privacy protection approaches; for privacy protection specification, all entities of the privacy protection approach will be formally specified, including privacy requirements. Specifically, the purpose-based access control method detailed in this thesis integrates *purpose* into privacy protection approaches, and provides a complete specification. This work also makes further exploration on the design considerations for privacy preservation in distributed collaborative environments. The objective of this chapter is to establish the context for this research.

This chapter is structured as follows. Section 1.1 introduces the research context of this work. Section 1.2 provides a healthcare scenario to illustrate the role of access

control and privacy requirements specification in protecting data security and privacy. It also briefly summarises privacy protection principles. Section 1.3 briefly introduces current policy specification researches and practices, and overviews VDM - the specification method used in this research. Section 1.4 introduces the motivation for this work and summarises the problem statement. Section 1.5 overviews the work presented in this thesis. Finally, Section 1.6 provides an overview of the remaining chapters.

1.1 Research Context

Privacy, as a legal, social and moral issue, has been a concern of social scientists and lawyers for a long time. It is a complex socio-technical system that requires interdisciplinary research from the domains of sociology, psychology, and computer science [And04]. It is especially a major concern for the deployment of information processing systems in this information age, since new technologies raise new threats to privacy rights. In order to protect privacy, many legislative and technical efforts have been made (e.g. [The95], [Uni96], [Obe01], and [Wor06]). This section gives an introduction to the context of privacy research, including the definition of privacy, privacy concern, privacy violation, privacy protection legislations, and privacy preserving technologies.

1.1.1 What is Privacy?

There are philosophical, legal, societal, and technical notions of privacy. The first real definition of privacy dates back to the 19th century. Samuel Warren and Louis Brandeis gave a definition of *privacy* in their seminal paper *The Right to Privacy* in the Harvard Law Review [WB90]. There's a principle in their privacy definition - "*the right to be let alone*". The definition of privacy was brought up due to the development of new forms of technologies that was coupled with other developments at that time. In the sense of the right to be let alone, Warren and Brandeis viewed the photography used by the yellow press as an attack on personal privacy, since photographic and printing technologies made it easier to share and spread images and text in public.

The definition of privacy given by Alan Westin [Wes70] is common in current use. It defined privacy as "*the claim of individuals, groups and institutions to determine for themselves, when, how and to what extent information about them is communicated to others*". According to this definition, natural person (individuals) as well as legal

organisations (groups and institutions) have the right to privacy.

The contents considered private were taken into account by some other definitions. For instance, Wikipedia defines privacy as "*the ability of an individual or group to seclude themselves or information about themselves and thereby reveal themselves selectively. The boundaries and content of what is considered private differ among cultures and individuals, but share basic common themes*" [Wik]. In this context, information not only relates to the raw data about an individual, e.g. name, age, gender, or address, but also relates to their credentials, e.g. degree certificate or benefit entitlement, and relates to their preferences, e.g. 'only my GP can access my genetic data'.

In general, the concept of privacy can be divided into three aspects [Ros04]:

- *territorial privacy*, which protects the close physical area surrounding a person, i.e. domestic and other environments such as the workplace or public space;
- *privacy of the person*, which protects a person against undue interference, such as physical searches, drug testing or information violating one's moral sense;
- *informational privacy*, which controls whether and how personal data can be gathered, stored, processed or selectively disseminated.

The emphasis of the research reported in this thesis concerns protection of private information of a person. In common speech, the words 'personal' and 'private' are sometimes used interchangeably. In this research, 'private' information means individually identifiable information about an individual, while 'personal' information is used to represent 'information about a person', which may or may not be private, so private information is treated as a subset of personal information. People want to keep their private information secret or confidential. Therefore, privacy in this research means *the right of individuals to have control over their private information*.

1.1.2 Privacy Concerns

Privacy is a major concern for the deployment of information processing systems. With the rapid development of information technology, many new initiatives were promoted, such as online banking, distance learning, teleworking, electronic commerce, road traffic management systems, research networks, healthcare networks and so on. These applications have changed our lives completely, but they have also brought different risks for society [FHS96]. On one hand, these applications provide people with great

convenience. For example, by providing credit card information, one can shop online at home instead of crowding around highstreet stores. In addition, these applications bring quality to services provided by organisations. For instance, the shopping history records available make it possible for retailers to offer their customers the most relevant recommendations. On the other hand, however, behind these attractive benefits, the risks of privacy violation are increasing. For these applications, a growing amount of personal information, such as transaction data, financial data, business data, sensitive medical data, or location data, may be collected, processed, and disseminated, with or without the data owners' consents. Moreover, in a networked society, an increasing amount of transactional data for network services will be available, and may be collected at different sites. So personal information may be collected, stored, and processed in various information processing systems.

The collection, storage, processing, and remote retrieval of vast amount of personal information have become a routine and inexpensive proposition [Cal03]. The easy access to private data increases the temptations for interested parties (individuals, businesses, or governments) to intrude upon people's privacy in unprecedented ways [RBE03]. Therefore, individual privacy is seriously endangered. Barry Barber [Bar97] cites the following problems if things go wrong regarding to privacy preservation:

- Public embarrassment or loss of public confidence
- Danger to personal safety
- Infringement of personal privacy
- Failure to meet legal obligations
- Breach of commercial confidentiality
- Financial loss
- Disruption of activities

Privacy is one of the major issues to be handled in many environments, such as the domains of e-Learning [FWBBP06, Klo06, MR07], e-Government [Sto02, RWB02, WC08, MCW08], e-Commerce [ACR99, Kor02, Acq04], healthcare [MPKB99, And00, RCHS03, BKP04, Hun05, BGA06] , and so on.

1.1.2.1 Privacy Debacles

Massive privacy breaches have made privacy a central concern in the information age [HAF05]. In recent years, news about privacy breaches continues to hit the headlines. Some of them are cited below:

1. Two computer discs holding the personal details of all families in the UK with a child under 16 have gone missing in October 2007. The Child Benefit data on them included names, addresses, dates of birth, National Insurance numbers and, where relevant, bank details of 25 million people [BBC07].

2. The HSBC banking group lost a computer disc with the details of 370,000 customers in March 2008. The customers' details included their names, dates of birth, and their levels of insurance cover [BBC08].

3. Four computer discs containing the details of 17,990 current and former staff were lost in July 2008 when they were sent between Whittington Hospital NHS Trust in north London and McKesson, a firm providing IT payroll services. They contained the names, dates of birth, National Insurance numbers, start dates and pay details of all staff of four NHS Trust organisations. They also contained the addresses of some of these NHS Trust staff [Sim08].

4. Ministry of Defence lost an unencrypted portable hard drive in October 2008, which contained the private details of 100,000 members of Army, Royal Navy and RAF personnel, including the names, addresses, dates of birth, passport and National Insurance numbers, drivers' licence, bank details, and their next of kin details. It also held details of another 1.7 million individuals who had made enquiries about joining the Armed Forces [Hop08].

5. A problem with the security of the ContactPoint database exposed personnel data for 55,000 vulnerable children. ContactPoint's shielding system was supposed to remove all details of the estimated 55,000 vulnerable children - apart from the name, sex and age - from the database, which will be available to children's services workers across the country. However, a flaw in the system meant when certain records were updated, a duplicate was created where the details were not shielded [Fre09].

6. Unencrypted memory sticks and CDs containing names, addresses and dates of birth of 9,000 Barnet school children were stolen in a break-in at a council

employee's house in north London. The data relating to Year 11 pupils from 2007, 2008 and 2009 included information about their educational attainment, entitlement to free school meals and home postcodes [BBC10].

These cases of massive privacy breaches were largely caused by the lost of computer discs. But since nowadays more and more information is being processed by electronic means, people also have concerns over their privacy in the e-environments. Many surveys conducted around the world (e.g. [Pri01, Pri06, Ele08, Inf08, Pri08]) have found consistently high levels of concern about privacy [ACR99].

1.1.2.2 Types of Privacy Violations

Daniel Solove provided a taxonomy of possible privacy violations [Sol06]. He categorised related privacy violations into four groups: *information collection*, *information processing*, *information dissemination* and *invasions*. These groups involve a *data subject*, which is the individual about whom a *data holder* has information. From that individual, various entities (other people, businesses, and the government) collect information. The data holders then process it, including store it, combine it, manipulate it, search it, and use it. The next step might be 'information dissemination', in which the data holders transfer the information or release the information to others. The general progression from information collection to processing to dissemination is the data moving further away from the control of the individual. 'Invasions' involve infringements directly on the individual. Instead of the progression away from the individual, invasions progress toward the individual and do not necessarily involve information handling.

Information collection includes making observations through *surveillance* and seeking information through *interrogation*. Information collection affects privacy by making people uneasy in how the collected information could be used. So it is a violation of privacy even if the collected information is never used. Furthermore, interrogation can place people in the uncomfortable position of having to refuse to answer questions. Information collection should also be controlled to prevent other violations of privacy such as blackmail.

Even information is collected in privacy-respecting ways, it can also be processed in ways that violate privacy. Information processing violations can be grouped into the following forms. *Aggregation* makes information available by combining and analysing separate pieces of information rather than collecting new information. Aggregation enables inferences that would be unavailable otherwise. *Identification* makes

information more available and may alter how a person is treated by linking information with a person by way of an identifier. *Insecurity* makes information more available to those who should not be granted access such as identity thieves, and it can also lead to distortion of data if false data is entered. *Secondary uses* makes information available for purposes for which it was not originally intended. *Exclusion* makes a data subject unable to know what records are kept, to view them, to know how they are used, or to correct them. All these forms of information processing create uncertainty on the part of the data subject. Exclusion causes the uncertainty by keeping information that the data holders have about the data subject secret. The other forms of information processing create the uncertainty by making information available in new, possibly unanticipated ways. Even without more material information misuse, such uncertainty can of itself be a harm since it forces the data subject to live in fear of how his information may be used.

After information is processed, the data holder will typically disseminate it to others for use. Some forms of information dissemination can violate privacy by providing information to inappropriate entities. *Confidentiality* can be breached when a trusted data holder makes unauthorised disclosure of confidential information about a data subject, such as the violation of patient-physician trust relationship. *Disclosure* involves not a violation of trust as with confidentiality, but rather the making of private information known outside the group of individuals who are expected to know it. *Exposure* occurs when embarrassing but trivial information is shared stripping the data subject of his dignity. *Distortion* is the presentation of false information about a person. Distortion not only harms the data subject, whose reputation is damaged, it also harms third parties who are no longer able to accurately judge the subject's character. *Appropriation* is related to distortion. Appropriation associates a person with a cause or product that he did not agree to endorse. Appropriation adversely affects the ability of the person to present himself as he chooses. *Increased accessibility* occurs when a data holder makes previously available information more easily acquirable. It is a threat to privacy as it makes possible uses of the information that were previously too inefficient, and furthermore, potentially encourages unintended secondary uses. *Blackmail* involves the threat of disseminating information unless some demand is met rather than really disseminating information. It allows a person to be dominated and controlled by another. With blackmail, the harm is not in the actual disclosure of the information, but in the control exercised by the one who makes the threat over the data subject. It uses private information to create an inappropriate power relationship.

Invasions represent interference in what is traditionally considered the private sphere of life. There are two forms of invasions. The first involves *physical intrusions* either upon private property (such as trespassing in the home) or upon the body (such as body searches by government officials). The second is *decisional interference*, which is interfering with personal decisions. Some view invasions as violations of other rights such as property and security rights in the case of physical intrusion, or the rights to autonomy and liberty in the case of decisional interference.

Privacy breaches and violations cause concerns about privacy. Privacy concerns might seriously hamper acceptance of information technology [GL04]. People start considering whether it is worth to risk their privacy to gain benefits offered by information technology. Collecting vital private information, such as credit card information and medical records, is often more difficult due to the lack of trust from data owners on information processing organisations. This distrust, caused by privacy concerns of the data owners, becomes a serious obstacle to the widespread adoption of information technology.

1.1.3 Legislative Efforts

Legislative efforts have been made for privacy protection. Many legislative acts aiming at personal data protection were proposed to ease privacy concerns. (1) In the European Union, the *EU Directive on the protection of individuals with regard to the processing of personal data and on the free movement of such data* [The95] was formally adopted in October 1995 by the European Council. The main objective of the Directive is the protection of privacy as a fundamental right, which is increasingly endangered in the networked society. Another objective of the Directive is providing a uniform minimum standard for privacy protection to prevent restrictions on free flow of personal data between EU member states for reasons of privacy protection. Besides, the EU Telecommunications Directive [The06] imposes obligations on carriers and service providers to protect the privacy of users' communication. Its rules severely restrict marketing activities as well as access to billing data. It is also required that Caller ID technology must incorporate an option for per-line blocking of number transmission. Furthermore, information collected in the delivery of communication must be destroyed once the call is completed. The EU Directive on the protection of personal data and the EU Telecommunications Directive aimed at enforcing a relatively high standard of personal data protection. (2) In the United States, there are several legislative acts that require certain organisations to provide protection of personal data,

such as the Health Insurance Portability and Accountability Act (HIPAA) [Uni96] for healthcare organisations, Children's Online Privacy Protection Act (COPPA) [Fed98a] for websites or online services directed to children, and the Gramm Leach Bliley Act (GLBA) [Fed98b] for financial institutions. (3) In Canada, the Privacy Act [Dep09], as a federal legislation, came into effect in 1983. The act sets out rules for how institutions of the federal government deal with personal information of individuals. The Personal Information Protection and Electronic Document Act (PIPEDA) [Off00] governs how private-sector organisations collect, use, and disclose personal information in the course of commercial business. It also contains various provisions to facilitate the use of electronic documents. The PIPEDA was passed to promote consumer trust in electronic commerce. There are also legislative acts that set out regulations for privacy protection in other countries.

Information processing organisations have also started to publish privacy policies to promise fair personal information practices and to acquire information owners' trust and to improve their confidence. However, privacy cannot be sufficiently protected solely by legislations and privacy policies. As stated by Michael Tschantz and Jeannette Wing [TW09], technical approaches to privacy must be part of the basis of creating privacy legislations and in designing privacy regulations. Legislations and policies need to be technically feasible to implement. Privacy protection requirements should also be enforced by information technologies, and privacy preservation should become an important design criterion for information and communication systems [FH01]. Therefore, privacy enhancing technologies, which can technically enforce legal, organisational and individual privacy requirements, have to be designed and implemented.

1.1.4 Privacy Protection Technologies

Privacy protection technologies refer to a variety of technologies that safeguard personal privacy by minimising or eliminating the collection of identifiable data [HB98]. The privacy protection technologies cover a diversity of aspects, such as:

- *Protecting the user identities with anonymity, pseudonomity, unlinkability, and unobservability of users*. The legal principle of necessity of data collecting and processing requires that personal data should only be collected or used for identification purposes when truly necessary. If personal data has to be collected, it should be rendered anonymous or pseudonymous as soon as the purpose for which the data was collected permits that.

- *Protecting the confidentiality, integrity, and availability of personal data using access control mechanisms*. The privacy requirements of *necessity of data processing of personal data of users and data subjects*, which requires that access to personal data is necessary for performing current task, and *purpose binding*, which requires that the purpose of the access should be compliant with the purpose for which the data was collected, can be technically enforced through an appropriate security policy and access control mechanisms.

Researchers in the information security community and the formal specification community have investigated privacy preservation and privacy policy specification from different perspectives. A number of approaches have been proposed in the literature with regard to privacy-preserving data accesses. Some research on privacy preservation approaches discuss technologies that protect user identities by enforcing anonymity [HS04, Poo99, Cha92, RR97, Oli95, Swe02], pseudonymity [KS03, GGK+99, AB08, AF08, LZY06, Cha81], unobservability or unlinkability [AS00, Mal08, EKS02, BFTs04, Odl03, SK03, PW87, CAMN+02]. Such technologies are important means to protect users from traffic analysis and the creation of communication and user profiles. However, to protect private information that has to be collected, processed or transmitted, and to implement privacy requirements such as *purpose binding* and *necessity of data processing of personal data of users and data subjects*, there is also a need for research on privacy technologies based on access control mechanisms. The research reported in this thesis will focus on privacy preservation through access control mechanism.

The definitions mentioned in section 1.1.1 and analysis show that privacy protection is mainly based on individuals' ideas. It depends on the sensitivity of the personal data to be collected, processed, and disseminated. The sensitivity of personal data is not only dependent on how intimate the details are, but it is also influenced by the purpose to access it and the context of use. For usual access controls, such as the access control for confidentiality protection, the information flow from objects with different security levels to subjects with different clearance levels can be used as the decision criteria, and it is fairly straightforward to implement. Since the comfort levels about privacy vary from individual to individual, the decision criteria for privacy protection is more complex than that for usual access control. In this work, data owners' privacy preferences are taken into consideration, and the entity of purpose is integrated into privacy protection approach.

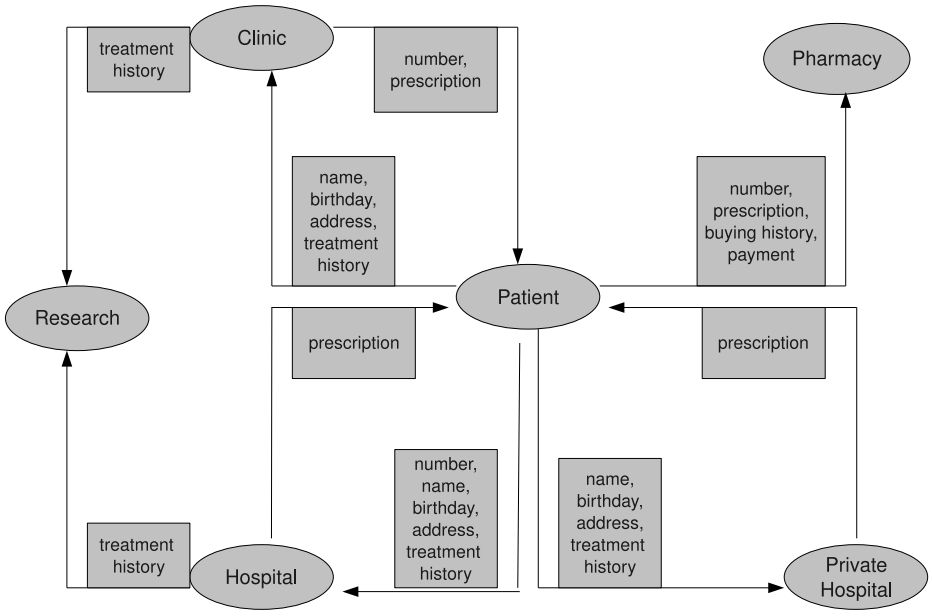

Figure 1.1: Data Accesses in a Medical Care Scenario

1.2 Privacy Protection in Data Access

The discussion about privacy protection is based on accesses to private personal data. In a privacy preserving data processing environment, a data owner should have total control over how his data may be accessed or used. This control encompasses several data processing phases, including the collection, storage, access, and dissemination of the data. In this section, data accesses and privacy requirements are discussed and analysed. In the first part, an exemplar scenario of data accesses is presented. The privacy requirements in this scenario and privacy requirements in general are then analysed, two major privacy protection guidelines are discussed, and a summary of privacy properties is then presented. Some organisations define their own privacy policies following general privacy protection principles. The importance of specification and implementation of these policies is then pointed out.

1.2.1 Data Access and Privacy: A Medicalcare Scenario

We use the following exemplar scenario in medical care to illustrate data accesses for the consideration of privacy preservation.

Scenario - Medical Care: Figure 1.1 describes a typical medical care scenario in the

NHS system. A patient wants to get medication. Firstly he may register with his GP. For the registration, he needs to provide certain personal information, some of which is confidential. For example, he may need to provide his name, date of birth, home address, contact details, and most importantly, his medical history, for the registration. This information will be stored in the Clinic. Once registered, the NHS system will issue a unique identification – the NHS number – to the patient. For every subsequent visit to his GP, the patient's health details, the prescription given, and the treatment history will also be updated and stored in the Clinic.

The patient may also visit other NHS or private hospitals or specialists for further treatments. For NHS hospitals, with the NHS number, doctors in the hospitals can obtain information of the patient. A copy of the medical record may be stored in the hospitals visited by the patient. These hospitals may also add/append more information into the patient's medical record. The patient may also leave separate medical information in private hospitals. In addition, when buying medicine with prescriptions at a pharmacy, the patient may leave some of his information, such as name, date of birth, home address, and payment information to the pharmacy.

1.2.2 Privacy Preservation Requirements

The information contained in medical record reveals some intimate aspects of the patient's life. It may consist of the patient's diagnostic and testing information, his family medical history, genetic information, and history of diseases and treatments. So the patient may treat it as private. Breach of privacy can be damaging to both the patient and the organisations concerned.

In order to protect his information, the patient may specify some privacy preserving requirements. The patient may

- give his consent to clinic and hospitals to collect and store his health information;
- give his consent to pharmacy to collect and store his purchasing information;
- allow his GP to access all his health information, including some highly private information, such as genetic information;
- allow the treatment and medication history to be used for research purpose in an anonymous way without obtaining the patient's consent, such as for the analysis of prevalent features and trend of infectious diseases (but the patient has the option to forbid such access);

- allow the contact information and purchasing history to be used for direct marketing purpose by the pharmacy (but the patient has the option to forbid such service);
- allow the contact information and purchasing history to be shared with third-parties, but this needs patient's consent;
- demand that the payment information, such as credit card information, be used only for payment authentication, and can not be shared with any third-parties.

These requirements express demand for privacy preservation from a data owner's point of view. For data processing organisations, in this case clinics and hospitals, to enforce privacy preservation, these requirements should be taken into account.

1.2.3 General Privacy Properties

The privacy preservation requirements of a patient on data accesses in a medical care scenario are shown above. Similar efforts have also been made in other contexts, such as in e-commerce and e-government environments. Some institutions have brought forward general privacy principles as guidelines for privacy protection. The following are two sets of major privacy preservation principles:

- OECD Guidelines for Data Protection

  The Organisation for Economic Co-operation and Development (OECD) has specified the guidelines on the protection of privacy and transborder flows of personal data [Org80]. The guidelines specify eight privacy principles: *collection limitation, data quality, purpose specification, use limitation, security safeguards, openness, individual participation,* and *accountability*. The OECD guidelines on data protection consider privacy protection in relation to personal data.

- FTC Fair Information Practice Principles

  The US Federal Trade Commission (FTC) brought forward the Fair Information Practice Principles [Fed98c]. They specify five core principles of privacy protection: *(1) Notice/Awareness; (2) Choice/Consent; (3) Access/Participation; (4) Integrity/Security;* and *(5) Enforcement/Redress*. These principles were put forward to ensure that the practices of collection and use of personal information are fair and to provide adequate privacy protection.

Both the OECD and FTC principles provide general privacy requirements that organisations should comply with. Based on the analysis of these principles and other requirements in legislative acts, the most essential privacy principles are summarised in [FH01] as follows.

- Lawfulness and fairness: Private data should be collected and processed in a lawful and fair way;
- Purpose specification and purpose compliance: The purpose for which private data is collected and processed should be specified. The subsequent use of private data is limited to those specified purposes, unless there is an informed consent by the data owner;
- Necessity of data collection and processing: The collection and processing of private data should only be allowed, if it is necessary for the requests falling within the responsibility of the data processing agency;
- Notification and access rights of the data owners: Data owners have the right to notification, and the right to correction, erasure or blocking of incorrect or illegally stored data;
- Security and accuracy: Appropriate technical and organisational mechanisms have to be taken to guarantee the confidentiality, integrity, and availability of private data. Private data has to be kept relevant, accurate, and up to date.

These principles express general privacy requirements that organisations should enforce in their data processing systems, and provide guidelines to data processing organisations for privacy preservation.

1.2.4 Organisational Privacy Policies

Companies and enterprises nowadays gather more and more information about their customers in order to provide more competitive services. It is especially true that companies and enterprises use applications on the web to monitor behaviors of their customers. This results in heightened concern over potential unintended disclosure and misuse of private information. Fortunately, the trend is such that organisations and enterprises are also becoming more serious about respecting the privacy of their

customers. On the one hand, they are required to comply with existing privacy regulations, on the other hand, they also need to take advantage of their privacy practices as an important capital to increase (or at least retain) their market share.

Governments, industries, and independent global consortium have encouraged organisations to define their practices for handling and sharing personal information, including reasonable communication of these policies to data owners. Therefore, in parallel to the proposition of general privacy principles mentioned above, many organisations have also defined their own privacy policies [Mic09, Yah06, Goo09, eBa10, Pay10]. These privacy policies are the major privacy requirements that organisations should enforce in their data processing systems.

An organisational privacy policy often reflects different legal regulations, promises made to customers, as well as more restrictive internal practices of the organisation [KSH03, KSW02]. In general, a privacy policy defines what data is collected, for what purpose the data may be used, whether the organisation provides access to the data, who are the data recipients, how long the data will be retained, and who will be informed under what circumstances [KSW02]. Privacy policies can be viewed from three different perspectives [SHW02]:

1. Preference: the data usage preferences of a particular data owner whose data may be collected by an organisation. For example, a data owner may have a preference such as:

   I do not want my medical record to be used for marketing purposes.

2. Promise: the privacy promises that an organisation advertises to enable a data owner to determine whether the data processing practices of the organisation match his preferences. For example, the following statement shows the privacy promises of a hospital [BHAS04]:

   We will not use your medical record for any purpose other than the primary purpose for which the data was collected.

3. Privacy Practices: the fine-grained access or privacy-control policy that governs the actual usage of the data by users of one or more organisations. Privacy practices are more detailed and restrictive than privacy promises. For example, a privacy practice may be stated as:

   A patient's medical record can be read by their primary doctors, for treatment purposes.

Privacy promises and privacy practices state privacy requirements from organisations' side, while privacy preferences express the requirements from data owners' point of view. Therefore, privacy preservation need to take into account privacy requirements from both organisations and data owners.

1.3 Privacy Policy Specification and VDM

Traditionally, privacy policies are written informally using natural languages, which makes it difficult to determine exactly who is authorised to access which object for what purpose. There is clearly a need to have some formal means to specify privacy rights and obligations that are promised by privacy policy statements and mandated by a number of legislations. With formal specification of privacy policies, the meaning of privacy-preserving requirements can be precisely determined. It is then possible to claim the abidance of the data processing practices with the privacy policies.

Formal methods is a technology that can help by providing foundational formalisations of privacy and practical tools for checking for privacy violations. It can and should be applied to security and privacy preservation [CW+96, TW09]. All the machinery of the formal methods community can help us gain a more rigorous understanding of privacy rights, requirements, and violations. We can use formal models, such as state machines and process algebras, to model the behavior of the system and its threat environment. We can use formal logics and formal languages to state different aspects of privacy, to state desired properties of these systems, to state privacy policies, to reason about when a model satisfies a property or policy, and to detect inconsistencies between different privacy policies. Automated analyses and tools enable us to scale the applicability of these foundational models and logics to realistic systems. We can also see the advantage of formal techniques, and in particular the logic-based ones, which provide validation tools for the agent models [BLPT04], and allow management model that are not application dependent.

1.3.1 P3P - A Policy Specification Method

Access control policies may be specified in formal logic such as Alloy [Jac02] and Authorisation Specification Language (ASL) [JSS97]. To formally specify privacy policies, the World Wide Web Consortium (W3C) proposed the *Platform for Privacy Preferences Project* (P3P) [CLM+02b]. P3P is a notable approach commonly used

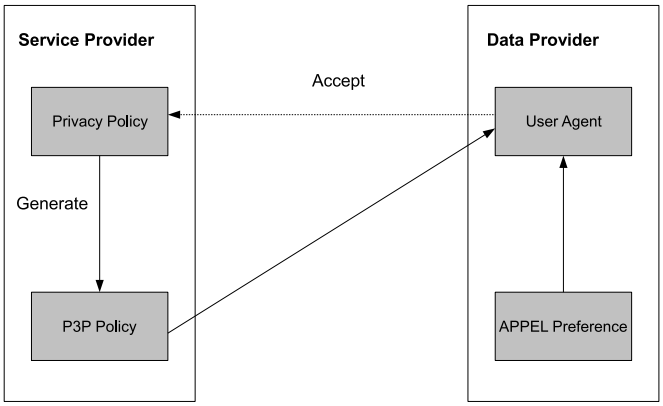

Figure 1.2: The Usage Scenario of P3P and APPEL

for specifying privacy policies of service providers on Internet. The goal is to enable machine-readable privacy disclosures, which could be retrieved automatically by web browsers [BCKM05, CAG02], ensuring that customers are informed about privacy policies before they release personal information to organisations.

Figure 1.2 describes the intended use of P3P and a language designed to interoperate with P3P - APPEL (*A P3P Preference Exchange Language*) [CLM02a]. A service provider publishes a privacy policy detailing its data processing practices, and then generates corresponding P3P policy reflecting this policy, which is in a machine-readable format. When a customer wants to use the service, the P3P policy is then transmitted to the customer's user agent, which is configured using the customer's privacy preferences, expressed in APPEL. If the service provider's data practices promised in the P3P policy conform to the customer's privacy preferences, then the user agent will accept the received policy.

P3P provides a mechanism for ensuring that customers can be informed about privacy policies before they release their personal information, but it does not provide a mechanism for making sure that organisations actually act according to their policies. P3P depicts privacy policies in a standard machine-readable format, but it only gives description of promises rather than technical measures for enforcement of the policies. In addition, without a technical and automated control mechanism in place, an obscure understanding of dataflows of personal data within an organisation may lead the organisation to unintentionally violate their own published privacy policies. The

problem is amplified if personal data is used not only within the organisation that collected the data, but also by other external organisations, such as partner organisations and managing authorities with a legitimate need to access the data.

1.3.2 VDM - A Rigorous Method of Specification

The specification approach chosen for this investigation is VDM (Vienna Development Method), which is based on a formal specification language – the VDM specification language. VDM is described in [Jon90]. In the rigorous method, objects are normally specified in terms of a model. The specification of an approach takes the form of an *operation* (or operations) on a *state* which defines a class or set of valid states. Well-formed conditions, known as *data type invariants*, may be used to limit the defined class further. Operations are specified using pre-condition predicates (predicates on a single initial state) and two-state post-condition predicates (predicates over the initial and final state values). This type of specification aims to be implicit, which means it aims to fix the properties required of the approach without specifying how they are to be achieved. All operations must preserve any data type invariant that may exist. They may change the value of the state as long as the new value is a valid state.

Initial specification should aim to capture abstract concept and avoid implementation detail. By gradually including design, algorithmic, and implementation detail, the development to a program proceeds either by data refinement or by operation decomposition. On the one hand, in data refinement, a new state closer to the implementation is defined, and the operations are redefined on this state. Given a state of the representation, a 'retrieve function' relates the new and more concrete specification to the more abstract specification, showing how the corresponding abstract state can be achieved. At each refinement stage, it is important to construct proofs which show why the refinement adequately models the previous stage. On the other hand, in operation decomposition, the state remains unchanged and the operations are redefined by combining simpler operations with control constructs such as sequence, selection, and iteration. As with the refinement process, a number of proof obligations arise for each of the control constructs used within the decomposition process.

The reasons for choosing this approach are:

- it provides the kind of expressive possibilities required by many applications, such as the engineering of critical systems;

- the satisfiability proof obligations only requires operations which are specified mostly with pre-conditions and post-conditions;
- the satisfiability proof approach is relatively simple, which facilitates the analysis of privacy requirements.

This research discusses the specification of the purpose-based access control method, in both intra-organisational and inter-organisational cases. A link is established between the task of requirements analysis and the design of the state of a VDM state. As well as the specification itself, this research also provides a good example of the development method.

1.4 Motivation and Objective of This Work

This work is motivated by the observation that privacy requirements and data processing practices are often misaligned, which leads to privacy violations. As described in a recently proposed road-map for web privacy by Anton et al. [ABLY07], there still remain vital research problems to be addressed for privacy protection. One major challenge is actual enforcement of privacy policies once the data has been collected. Enterprises and organisations have taken various approaches to protect customer privacy, such as publishing privacy policies on their websites, enabling P3P compliant privacy policies, incorporating privacy seal programs (e.g. TRUSTe [Ben99], BBBOnline [BBB09], CPAWebTrust [Web]), etc. But these approaches cannot truly safeguard customers' privacy because they could not ensure that customer private data is properly handled after it is collected. Enterprises and organisations' actual data processing practices might intentionally or unintentionally violate the privacy policies they published on their websites.

A big step towards enforcing privacy policies in the information system of an organisation is considering them when making decisions over accesses to private data. With that vision, Powers et al. proposed privacy policy rules [PAS02], which comprise of *data type* of data items, *operation* on the data, *data user* who accesses data, *purpose* of data access, *condition* that restricts the accesses, and *obligations* that need to be carried out by the organisation after the access. Traditionally, policy specification is isolated from requirements analysis, which often results in data processing practices that are not in compliance with privacy requirements. As mentioned in Section 1.3, in order to precisely determine the meaning of privacy preserving requirements, to

maintain consistency between privacy protection promises and actual data processing practices, there is a need to formally define, specify and enforce privacy protection requirements.

In conclusion, the objective of this work is to address the problem summarised as follows. *Because privacy policy specification is typically isolated from requirements analysis, the resulting practices often do not comply with privacy requirements. This leads to the development of systems that neither comply with the privacy requirements nor adequately protect the information with which they are entrusted. Software and security engineers need methodological support for specifying privacy policies and ensuring compliance with privacy requirements.*

1.5 Overview of This Work

The purpose-based access control approach presented in this thesis integrates the entity of purpose into privacy preservation approach. Previous work has shown that when specifying a privacy policy, the concept of purpose should also be taken into account for access control. To specify privacy protection approaches, we must examine privacy requirements from the organisations' side to identify users and their interactions with the system, and examine privacy preferences from the data owners' point of view to identify the data to be protected.

As previously mentioned, this thesis focuses on the specification of privacy preservation. There are two major advantages in focusing on specification. Firstly, privacy preservation specifications are machine-enforceable, whereas natural languages policies are not. These specifications are closer to real data processing implementations. Secondly, it is then possible for software and security engineers to specify and analyse policies that meet organisational goals using formal languages.

The first major contribution of this approach is a development scheme introduced in this thesis that ensures compliance between privacy requirements and data practices by integrating policy specification with requirements analysis. We derive access control policies from basic privacy requirements and high level security and privacy guidelines. Because privacy requirements come from these sources, this development scheme helps ensure that a data processing system is actually enforcing privacy policies. We specify the entities in a purpose-based access control model, the invariants corresponding to the privacy requirements, and model operations together with their

proof obligations. This specification is an iterative process. Although we derive specification from requirements analysis and approach design, we can also improve requirements and approach design through privacy preservation specification by clarifying ambiguities in the requirements and resolving inconsistencies between requirements and data processing practices.

Another major contribution of this work is the design considerations for privacy preservation in distributed collaborative environments. This is achieved in two ways. First, we introduce the concept of data composition to overcome limitations of object type in the purpose based access control approach, ensuring that personal information is accessed from two or more parties only if agreed privacy policies and privacy preferences on data compositions are satisfied. Second, we provide mechanisms for facilitating privacy policy matching and privacy preference compliance among distributed collaborative organisations.

1.6 Outline of Thesis

The rest of this thesis is organised as follows:

Chapter 2 provides an overview of related work in privacy preservation approaches, to position the work presented in this thesis.

Chapter 3 presents privacy protection approach specifications. It analyses formalism of privacy preserving approaches in coping with concept of purpose. It illustrates the motivations for integrating privacy protection approaches and specifications, and argues the necessity for purpose based access control approach.

Chapter 4 then details the purpose based access control method. Entities, relationships, and privacy requirements in a single organisation are presented. It outlines the basic framework of our privacy preserving access control model, and presents essential concepts, definitions of the main entities, and the formal specifications of mapping functions and the access granting rules.

Chapter 5 illustrates the use of VDM in the formal specification of the purpose based access control model in a medical care scenario. It shows how an initial specification can be formed and then manipulated in a rigorous way through the careful introduction of design detail in the form of data structure and operations.

Chapter 6 looks at entities and processing phases required for application of the

purpose based access control model to distributed computing environments. Two design considerations are presented: (1) how to ensure that private information is accessed from two or more parties only if agreed privacy policies and privacy preferences are satisfied, and (2) how to facilitate privacy policies matching and privacy preference compliance among distributed collaborative organisations.

Chapter 7 summarises contributions of this thesis and future work that is needed to further refine the method.

Chapter 2

Literature Overview: Privacy Preservation

2.1 Chapter Introduction

The underlying bases of the work reported in this thesis are two research areas: information security and formal specification. Researchers in the information security community and the formal specification community have investigated privacy preservation approaches and privacy preservation specification from various perspectives. To position the work in this thesis, some of the most relevant previous work in both areas is briefly surveyed. An overview of relevant work in privacy preservation approaches is provided in this chapter. Related work on privacy preservation specification will be investigated in next chapter.

2.1.1 Privacy Preserving Access Control Approaches

Privacy preservation methods approach privacy preservation from different perspectives. The most straightforward way is to adopt similar concepts as security levels and clearance levels used for protection of confidentiality. The HPSF method [Obe01], which introduced a concept of privacy sensitive level, falls into this category. Some approaches consider operations on private data, and propose to enforce basic legal privacy requirements. Fischer-Hübner et al [FHO98] extended Task Based Access Control approach [ST94] to enforce privacy requirements such as *purpose binding* and *necessity of data processing*. Since Role Based Access Control (RBAC) method is a

policy neutral and flexible access control technology, some approaches propose extensions to RBAC method for privacy protection, such as the concept of *explicit denial* for permission assignment, or the introduction of consent as a model entity. Some other approaches introduce purpose to express privacy requirements from the data owners' point of view. The rest of this chapter will present these approaches in detail.

2.1.2 Privacy Preserving Approach Design Considerations

The design considerations for a privacy preserving access control approach consist of three aspects:

- Basic privacy requirements

  The approach need to implement basic privacy requirements, such as *necessity of data processing* and *purpose binding*.

- Scalability of management

  For the consideration of efficiency, both object management and subject management in the approach should be scalable.

- Authorisation mechanism

  The approach integrates authorisation into access control mechanism and enforces the privacy policy.

This chapter presents relevant privacy preservation approaches. Based on the analysis of features of these approaches, this chapter will conclude with an evaluation on these approaches against the above mentioned aspects.

2.2 The Hierarchical Privacy-Sensitive Filtering Model

The *Hierarchical Privacy-Sensitive Filtering (HPSF)* model was proposed by Oberholzer [Obe01]. The model was introduced to protect personal information of a patient based on the sensitivity level of a specific data item regarded by the patient. The model uses the concept of *privacy sensitivity level* (PSL). It defines a PSL for every data item or every sensitive data item. The PSL value of a data item indicates how sensitive a data owner is about disclosure of the data item. The higher a PSL value, the more sensitive the data item regarded by the data owner. The model also defines a *user*

privacy-sensitive level (UPSL) for every user requested access to the data items. A user will only be allowed to see the contents of those data items where PSL $\leq$ UPSL.

A patient discloses personal information about himself during a data collecting process. In addition to information a patient normally discloses, the model requires the patient to indicate how sensitive he is about privacy of his personal data. The patient specifies PSL for his data items, indicating the level of personal privacy he consents to, with regard to the use of the data item. Each data item may be assigned with one of the four different privacy sensitivity levels: *non-sensitive (0)*, *sensitive (1)*, *very sensitive (2)*, and *extremely sensitive (3)*, as follows:

- *Non-sensitive (PSL=0)*: Data items that are anonymous and not private. Examples of such data items are gender, city, and language. A user would not be able to identify a patient easily by only viewing non-sensitive data, and therefore the privacy of the patient will be preserved.

- *Sensitive (PSL=1)*: Data items that contain identifying information. The data at this level can be used to link records in different tables.

- *Very sensitive (PSL=2)*: Anonymous data items that are very private. Examples of such data items are religion and race. Maybe a patient with surname at PSL=1 is not sensitive about which city (PSL=0) he lives in, but does not want people to know his religion (PSL=2). Data items of treatment records or prescription for the patient may be given PSL 2 in the model.

- *Extremely sensitive (PSL=3)*: Personal data that are viewed to be extremely private. Extremely sensitive personal data are defined to be so private that the data can not be disclosed to anyone or any third party. An example may be the case where a doctor has diagnosed a patient as HIV positive and the patient does not even want his family to know that.

In addition to roles that are granted to users of the databases, every user that performs an operation on the data of a patient must be assigned with a PSL related to the patient. This PSL will be referred to as a user PSL (UPSL). The UPSL value for a specific user or role indicates the level of access that the user or role will be allowed with regard to the data item. The UPSL of a user will be based on the privacy sensitivity levels or preferences set by the patient or on the default PSLs set by the hospital. The higher a UPSL value, the higher the PSL level of data items the user is given access

to. When two users view a same data item, if their UPSLs are different, they will not necessarily see the same contents.

Users have access to certain data items of a patient where the PSL value of the data items are no higher than the UPSL value of the users. A user with UPSL 3 will have access to data items of a patient at PSL 3 and lower PSLs at 2, 1 and 0. A second user with UPSL 1 will have access to data items at PSL 1 and lower PSL 0 for the patient, but will be denied access to data items at levels PSL 2 and PSL 3 of the same patient. However, if the second user has a UPSL 2 preferred and set by a second patient, the user will then have access to data items at PSL 2 and lower levels belonging to the second patient.

Sometimes, it may be necessary for a user to access a data item of a patient, even if the patient did not give consent for the specific user to access the data item. In such a case where non-consented access seems to be necessary, the user can override the PSL of the data item. The PSL of the data item may be temporarily lowered to the UPSL of the user to allow him to view the contents of the data item. Details of the temporary overriding transaction will be logged in a privacy alert log file, after which the user will be allowed to access the specific data item. This temporary lowering of the PSL of a data item will be viewed as a possible privacy violation until it is cleared when a privacy officer deems that the overriding was indeed necessary. So, the privacy officer must inspect the privacy alert log file at predetermined intervals and investigate all probable violations of personal privacy. In cases where personal privacy violations are suspected, the patient must be informed and the case must be investigated. The patient should also have the right to view all references to probable personal privacy violations pertaining to him.

This concept of privacy-sensitive level is similar to the concepts of security and clearance levels used in the *multi-level security* (MLS) model [BP76], which mainly aims at preserving data confidentiality and integrity. As privacy preservation is mainly based on individual's ideas, the privacy sensitive levels may vary from individual to individual. It is more difficult to define persistent PSLs and UPSLs than to define confidentiality levels. In addition, defining PSLs on the granularity of data items reduces the management scalability. Therefore, using privacy sensitive levels to preserve individual privacy is more complex than using security and clearance levels to preserve confidentiality. Moreover, the non-consented accesses may cause personal privacy violations. Although investigations on alert log files may help to identify privacy violation, the approach cannot prevent it from happening due to the PSL overriding mechanism.

Furthermore, the privacy level approach cannot reflect some important privacy requirements, such as purpose binding.

2.3 The Task-Based Approach

Fischer-Hübner et al [FHO98] proposed another approach to privacy preservation by extending a task-based access control method [ST94] with enforcement of basic legal privacy requirements. In this approach, data can only be accessed in a controlled manner by executing a *task*. A task consists of a set of certified operations, representing the set of "*necessary accesses*" to object classes. Example of tasks in a hospital information system are the tasks such as "*diagnosing*", "*operation*", or "*therapy*". This approach specified a privacy policy based on the tasks that a user is performing. The privacy policy specified in this method is described as follows: *A subject may only have access to personal data, if this access is necessary to perform its current task and only if the subject is authorised to perform this task. The subject may only access data in a controlled manner by performing a (well-formed and certified) transformation procedure, for which the subject's current task is authorised. Besides, the purpose of its current task must correspond to the purposes for which the personal data was obtained or there has to be consent by the data objects.*

In order to specify privacy invariants and to formulate privacy constraints and information flow rules for tasks, the concept of *purpose* is introduced as a model variable. Normally, the rules regulating personal data processing should specify purposes for personal data processing. In this approach, certain purposes are specified when personal data is collected. Moreover, every task is defined to serve certain purposes. For each application, it is then necessary to determine purposes for its tasks, as well as purposes for which personal data is collected.

This approach defines several entities which are listed as follows. A task is looked on as a system state transition. The set of tasks that a subject is authorised to perform is defined as its *authorised task*. The task that is currently performed by a subject is defined as its *current task*. Invariants are used to define relationships between variables within individual states. The privacy invariants should hold in each system state. Constraints are used for specifying state transitions. State transition functions describe changes of state variables. They are divided into general transition functions and privileged transition functions. General transition functions are defined for actions accessing objects and executing transformation procedures, such as *get access, release*

access, create object, delete object, change current task, execute transformation procedure, exit transformation procedure, and so on. Privileged functions are used to administrate access control information, such as tasks, purposes, authorised tasks for a subject, authorised transformation procedures for a task, object classes and their purposes, necessary accesses and consent. A constraint differs from an invariant in that it takes into account the relationships between values in two successive states, that is, before and after each state transition function.

Illegal information flow may occur when a subject reads from a personal data object and then writes the information obtained from it to a non-personal data object. In order to control information flow, the mechanism of program certification on transformation procedure is introduced. Certification mechanism states that a subject can access an object only by executing a certified transformation procedure which is authorised for its current task. A program certification could check that no statement in the transformation procedure, if executed, would cause an information flow violation. In order to check and certify the information flow, the certifier has to have information about object classes of objects and their purposes. Illegal information flow could be prevented by a careful design of transformation procedures and an appropriate definition of necessary accesses.

This approach illustrates how the privacy policy is implemented in an imaginary hospital example. When personal data is collected, the consent of the data owner and certain purposes about the data usage are specified. When a user requests to access data while performing a task, he can access the data only if this access is necessary to perform his current task, and he is authorised to perform the task (*requirement of necessity of data processing*). To enforce privacy, the purpose of the task, currently performed by the user who requests to access personal data, will be checked against the purpose for which the personal data were obtained, or the consent given by the data owner (*requirement of purpose binding*).

The major contribution of this approach is that it has illustrated two important requirements - *necessity of data processing* and *purpose binding* - for privacy preservation, and demonstrated how a privacy policy may be enforced. However, because this approach is based on the tasks a user is performing, and this approach does not provide much support for roles management, the subject management in this approach lacks scalability.

2.4 The Role Based Access Control Approach

To ease the scalability concern over access control in the design of privacy-preserving access control solutions, the *Role Based Access Control* (RBAC) method has received considerable attentions due to its policy neutral and flexible features. Before discussing extensions to RBAC for privacy preservation, including RBAC with explicit denial and privacy-aware RBAC, this section first gives an overview of the Role Based Access Control method.

2.4.1 An Overview of RBAC

The roots of role-based access control can be traced back to the earliest access control systems. The concept of roles has been adopted in access control products in the 1970s and 1980s, such as Resource Access Control Facility (RACF) [SCFY94]. The concept of user group is closely related to the concept of role, so it was used to implement role [Bal90, DHTK93]. Over the years, many researchers have proposed models for RBAC [NO93, NO94, NO99, FBD99]. Role based access control, as formalised in 1992 by David Ferraiolo and Rick Kuhn [FK92], has become the predominant approach for advanced access control because it reduces the security administration cost. A variety of IT vendors, including IBM, Microsoft, Secure Computing, and Siemens, began developing products based on the model [RS98]. RBAC features are also supported in commercial database management systems, such as Informix, Sybase, and Oracle [Not96]. In 2000, the Ferraiolo-Kuhn model was integrated with the framework of Sandhu et al. [SCFY96] to create a unified model for RBAC, published as the NIST RBAC model [SFK00] and adopted as an ANSI/INCITS standard in 2004. Today, most information technology vendors have incorporated RBAC into their product lines, and the technology has found applications in areas ranging from health care to defence [DS99, Cha01], in addition to the mainstream commerce systems for which it was designed. Moreover, role-based systems have already been developed for some time by a variety of organisations [BB89].

2.4.1.1 RBAC Entities

The entities in the RBAC model are defined as follows: A *user* is a person, a *subject* typically refers to a user, but it could be extended to include computer process or autonomous agents, a *permission* is an approval to execute an operation on one or more protected *objects*, an *operation* could be a simple access mode, e.g. read/write/update,

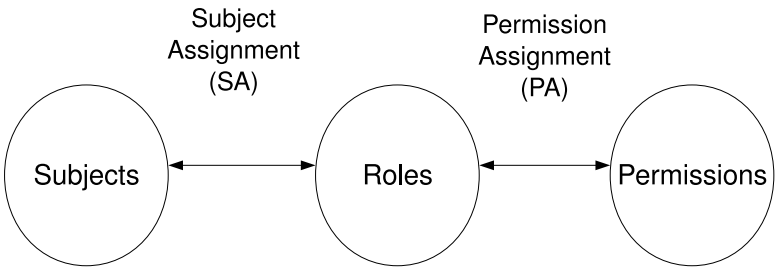

Figure 2.1: A Basic RBAC Model

or a complex access such as a method invocation in an object-oriented system, and a *role* is a named collection of permissions. Suppose $user$, $subject$, $role$, $operation$, and $permission$ denote respectively a set of users, subjects, roles and permissions in a system, the following relationships and functions are defined to specify mappings among users, subjects, permissions, and roles:

$subject\ assignment\ SA \subseteq subject \times role$: a many-to-many mapping from subjects to roles;

$permission\ assignment\ PA \subseteq permission \times role$, a many-to-many mapping between permissions and roles;

$subject$-$user$(s:subject): the user associated with subject s;

$authorised$-$roles$(s:subject): the roles associated with subject s;

$role$-$members$(r:role): the users authorised for role r;

$authorised$-$role$(u:user): the roles associated with user u;

$role$-$operations$(r:role): the operations associated with role r;

$operation$-$objects$(op:operation): the authorised objects associated with the operation op.

Figure 2.1 gives a schematic description of role assignment, including subject assignment and permission assignment, in a basic RBAC model. Figure 2.2 illustrates relationships among users, roles and objects.

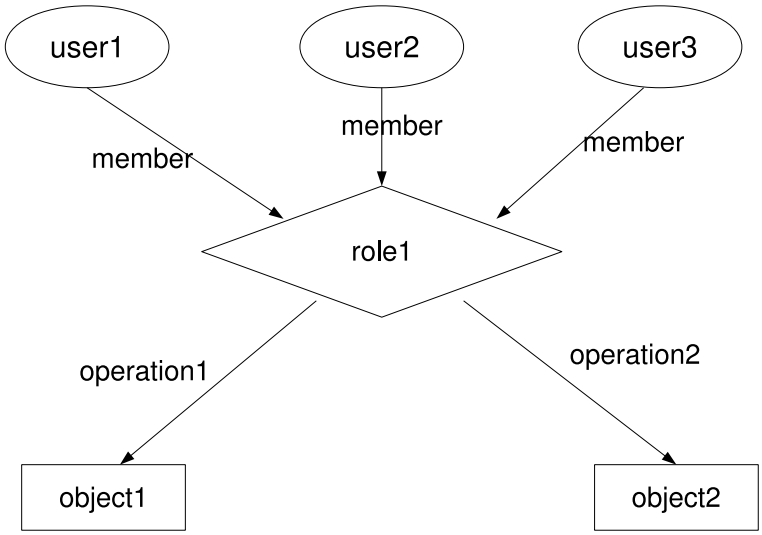

Figure 2.2: RBAC User, Role, and Object Relationship

2.4.1.2 RBAC Properties

In this section, the properties of the role based access control method will be described. They are required and used to define concepts and constraints like *role hierarchies, role authorisation, role activation, operational separation of duty* and *authorised access to objects*. For the research detailed in this thesis, since management scalability is one aspect of the design considerations, it consists of subject management with roles. So properties of the RBAC are detailed as follows.

Property 1: Role Hierarchy

Roles can have overlapping responsibilities and privileges. Role hierarchies are defined to allow roles to "contain" other roles, that is, one role may implicitly include the privileges and constraints that are associated with another role. For example, in a hospital information system, the role "doctor" contains the role "health-care provider" and thereby inherits the privileges from health-care provider. Role hierarchy can be described as:

Role Hierarchy: If a subject is authorised to access a role and that role contains another role, then the subject is also allowed to access the contained role:

$\forall\ s$: $subject$, r_i, r_j: $roles$: $r_j \in authorised\text{-}roles(s) \wedge r_i > r_j \Rightarrow$
$r_i \in authorised\text{-}roles(s)$

Property 2: Static Separation of Duty

The association of a user with a role should be subject to the principles of *least privilege* and *static separation of duty*. The principle of least privilege requires that a user be given no more privileges than necessary to perform his job function. Through the use of RBAC, least privilege can be easily achieved by granting to a role only those operations that need to be performed by the members of the role. The principle of static separation of duty requires that if a user has been authorised as a member of one role, the user can not be authorised as a member of a second conflicting role. For example, in the bank, the roles *teller* and *auditor* are two conflicting roles – they are *mutually exclusive*. Mutually exclusive roles of a given role, denoted as $mutually\text{-}exclusive\text{-}authorisation(r : roles)$, is the list of roles that are mutually exclusive with r.

The static separation of duty property are specified as follows:

Static Separation of Duty: A user is authorised as a member of a role only if that role is not mutually exclusive with any other roles for which the user already possesses membership:

$\forall\, u$: $user$, r_i, r_j: $roles$, $i \neq j$: $u \in role\text{-}members(r_i) \wedge u \in role\text{-}members(r_j)$
$\Rightarrow r_i \notin mutually\text{-}exclusive\text{-}authorisation(r_j)$

Property 3: Cardinality

In the RBAC model, it is possible to restrict the number of users allowed for a role at any given time. For example, only one user should act as a manager or as a department chair at any given time. The number of users allowed for a role and the existing number of users associated with a role are specified by the following two functions:

$membership\text{-}limit(r : roles)$ gives the membership limit (≥ 0) for role r.

$number\text{-}of\text{-}members(r : roles)$ gives the number of existing members of role r.

The *cardinality* property is described as:

Cardinality: The capacity of a role cannot be exceeded by an additional role member:

$\forall r : roles$: $number\text{-}of\text{-}members(r) \leq membership\text{-}limit(r)$

Property 4: Role Authorisation

The RBAC model defines property for role authorisation. The following functions define active roles for a subject which are the roles that the subject is currently using:

$active\text{-}roles(s : subject)$ gives the current list of active roles for a subject s.

A role can be activated by a user, if the user is authorised for the proposed role. This is specified by the following property:

Role Authorisation: A subject can never have an active role that is not authorised for

it:

$\forall s : subject$: $active\text{-}role(s) \subseteq authorised\text{-}role(s)$

Property 5: Role Execution

The following function enables subjects to execute operations:

$exec(s : subject, op : operation)$ is TRUE if and only if subject s can execute operation op, otherwise it is FALSE.

Once it is determined that a role is authorised, the operation can be executed if the role is active. This is described as:

Role Execution: A subject can execute an operation only if the subject is acting within an active role:

$\forall s : subject$, $op : operation$: $exec(s, op) \Rightarrow active\text{-}role(s) \neq \phi$

Property 6: Dynamic Separation of Duty

It is required that the activation of a proposed role is not mutually exclusive with any other active role(s) of the user. This requirement provides administrators with the capability to enforce dynamic separation of duty. In contrast to static separation of duty which places constraint on role authorisations, dynamic separation of duty places constraint on simultaneous activations of roles. For example, a user could be authorised for both the roles Payment Initiator and Payment Authoriser, but can dynamically assume only one of these roles at the same time. The mutually exclusive active role function for the proposed role and the dynamic separation of duty property are specified as:

$mutually\text{-}exclusive\text{-}activation(r : roles)$ gives the list of active roles that are mutually exclusive with the proposed role r.

Dynamic Separation of Duty: A subject can become active in a new role only if the proposed role is not mutually exclusive with any of the roles in which the subject is currently active:

$\forall s : subject$, $r_i, r_j : roles, i \neq j$: $r_i \in active\text{-}roles(s) \wedge r_j \in active\text{-}roles(s)$

$\Rightarrow r_i \notin mutually\text{-}exclusive\text{-}activation(r_j)$

Property 7: Operation Authorisation

The proposed operation has to be authorised for a subject's active role. This is described as the following property:

Operation Authorisation: A subject can execute an operation only if the operation is authorised for the role in which the subject is currently active:

$\forall s : subject, op : operation$: $\exists r : roles$, $exec(s, op) \Rightarrow$

$r \in active\text{-}roles(s) \wedge op \in role\text{-}operations(r)$

Property 8: Operational Separation of Duty

RBAC can be used by a system administrator to enforce the policy of operational separation of duty, which requires that for all operations associated with a particular business function, no single user can be allowed to perform all operations. The operation function and the operational separation of duty property can be specified as:

$function\text{-}operations(f : function)$ gives the set of all operations required for a business function f.

Operational Separation of Duty: For all operations required for a particular business function, no single user should be allowed to have the authorised roles to perform all operations:

$\forall u : user, r : role, f : function, r \in user\text{-}authorised\text{-}roles(u)$:

$\neg(function\text{-}operations(f) \subseteq \cup role\text{-}operations(r))$

Property 9: Object Access Authorisation

Control of access to objects is specified with the following function and property:

$access(s : subject, o : object)$ is TRUE if and only if the subject can access the object, otherwise it is FALSE.

Object Access Authorisation: A subject can access an object, only if the role is part of the subject's current active role set, the role is allowed to perform the operation, and the operation to access the object is authorised:

$\forall s : subject, o : object$: $access(s, o) \Rightarrow \exists r : roles, op : operation$:

$r \in active\text{-}roles(s) \wedge op \in role\text{-}operations(r) \wedge o \in operation\text{-}object(op)$

2.4.1.3 Analysis on Role Based Access Control

RBAC is an access control method that decouples users from privileges by the interpositioning of roles [FCD95, FK92, GI96, AK05]. A role is defined as "a job function within an organisation that describes the authority and responsibility conferred on a user assigned to the role" [SCFY96]. A role is determined by a set of operations that a user or a set of users can perform within an organisation. Authorised operations on objects are allocated to roles by a security administrator. A role should reflect the responsibilities of a position or job description in the context of an organisation, for example, a role in a bank can be *Manager* or *Clerk*. Membership of users in a role is also granted and revoked by the security administrator, on the basis of the users' specific job responsibilities and qualifications. When an individual is assigned with the responsibility to perform a particular job, the security administrator puts him in an appropriate role. He can then exercise the privileges given to that role. Comparing to traditional *Access Control List* (ACL) based approaches [BP76, SHV99], RBAC does

not allow users to be directly associated with privileges, and all privileges are defined in terms of roles. This decoupling lends a greater degree of access control scalability to systems in which accesses must be regulated.

The layer of indirection between users and privileges is a defining feature of the RBAC. It makes the task of authorisation management much easier and scalable than traditional security models. Workflow processes are relatively stable whereas user-task assignments are not, since individual user's job responsibilities change as they move between departments, change jobs etc. Therefore, the ⟨role, privilege⟩ associations typically change less frequently than ⟨user, privilege⟩ associations. Cost, complexity, and potential errors will be reduced by assigning permissions to roles rather than users within large scale systems.

RBAC was later proven to be policy neutral, which means that it is a way for expressing policy rather than embodying a particular security policy. RBAC can be configured to enforce traditional access control policies, such as mandatory access control and discretionary access control [OSM00]. Specifically, lattice-based access control can be realised as a particular instance of systems that support general RBAC [San96]. Furthermore, RBAC supports several well-known security principles: *information hiding*, *least privilege*, *separation of duty*, and *data abstraction*.

RBAC provides a way to model organisational security policies. It is policy neutral in the sense that, by using role hierarchies and constraints, a wide range of security policies can be expressed. Incorporating attributes into RBAC has been proposed for implementing requirements for privacy preservation. Two extensions to RBAC for privacy preservation are presented in the following sections.

2.4.2 RBAC with Explicit Denial

Role assignments in a RBAC system typically adhere to the principle of general denial with explicit consent, i.e. anything that is not explicitly allowed is implicitly denied. Only those users that are assigned to roles are permitted to access objects for which the roles have been assigned with privileges. Through static constraints, users can be prevented from joining roles for which they are not qualified (*prerequisite constraints*) or combinations of roles that are inappropriate (*static separation of duty*) [GI96]. Where selective role activation is permitted, dynamic constraints allow users to belong to multiple roles but ensure that only a subset of those roles may be active at a time (*dynamic separation of duty*) [SZ97].

Reid et al proposed the concept of explicit denial for role assignment [RCHS03].

The authors argue that in a health care context, for consideration of safety and reliability, the opposite form of expressions should be used, that is, denied access should be explicitly stated. Health information networks require a combination of both expression forms. An access policy can then be specified in a way that mirrors how consumers commonly think about who should have access to their health information. Standard RBAC models do not support policy expression in the form of general consent with explicit denial. Therefore, an extension that supports general consent with explicit denial to the RBAC model is proposed. It permits *allow* and *deny* policies to successively qualify each other for inheritance in a role hierarchy. A data owner's consent instructions are expressed by means of allowing and denying hierarchically related roles, which employ a wide range of classifications and granularities. This results in a mechanism for clinicians and patients to easily understand and manage.

Permission of each role is determined by access right factors. Any node in a role hierarchy has one of the three access right factors: *explicitly allowed*, *explicitly denied* or *ambiguous*. Cases for explicit allowance and denial are simple – the node is accorded that access role without relying upon other nodes in the hierarchy. An ambiguous node inherits permission from its child nodes. If any of its immediate child nodes is explicit denial, the ambiguity of the node will be resolved to denial. Otherwise, ambiguous child nodes must be resolved firstly into either allowance or denial. Any emerging denial will be passed to ambiguous parent. The ambiguity of a parent node is resolved to allowance only if none of its child nodes is denial. An ambiguous node whose child nodes are all ambiguous can not be resolved. In this case, the method decrees that ambiguous leaves can be automatically resolved to denial.

The explicit denial mechanism specified practical privacy requirements for consent based health information sharing. It extended RBAC to support access policy expression in the form of general consent qualified by explicit denial. For the consideration of efficiency, it used nesting of explicit denials and consents, which successively qualify each other in a role hierarchy. It focuses on the comprehensiveness of policy expression and the efficiency of permission assignment. However, associating access right factors to permissions reduces independence of the management of different parts in a policy, because access right factors are normally set by data owners when data is collected. Moreover, the lack of a systematic data model decreases data object management scalability.

2.4.3 Privacy-Aware Role-Based Access Control

The concept of *role* in RBAC and *purpose* have a close relation with each other [He03]. When a role is derived from business tasks, a certain set of responsibilities is assigned. Purposes are defined implicitly along with these responsibilities. Since purpose is an important element in a privacy policy statement, it is possible to embody this relationship for privacy preservation. It can be used to specify purpose binding, one of the key privacy requirements.

He et al. [He03] proposed a *Privacy-Aware Role-Based Access Control* (*PARBAC*) method for enforcing privacy policies within an organisation. In this approach, privacy is considered together with security protection and data management technologies. *RBAC*, *Domain-Type Enforcement*, and privacy protection are combined together. Privacy enforcement is supported by combining access control and privacy management. Business purposes and data usage policies are modelled with *Domain-Type-Enforcement*.

PARBAC adopted the expression of a general privacy policy rule as in [PAS02]:

$allow\ [DataUser]$
$to\ perform\ [Operation]\ on\ [DataType]$
$for\ [Purpose]\ provided\ [Condition]$
$carry\ out\ [Obligation]$

The concept of *data type* is important for privacy enforcement. In privacy policy statements, data objects are usually grouped together for consideration. For example, a sample privacy policy rule may be stated as "purchase history can be used for research analysis in an anonymous way". Another sample privacy policy rule may be "contact information cannot be used for marketing purpose". Here, contact information and purchase history are both examples of data types. Name, address, postcode, telephone number, etc. belong to the data type of contact information. Product, price, quantity, etc. belong to the data type of purchase history. Classifying data objects into data types makes the data object management more scalable.

Conditions and obligations are proposed for helping make and enforce authorisation decisions. Some privacy policies state *conditions*, which are prerequisites to be checked when making authorisation decision. For example, a privacy policy may require an organisation to obtain data owner consent before they use personal information for a particular purpose. Here, "*obtaining data owner consent*" is a sample condition. Another example of condition is retention period – how long the data will be kept. The condition here is "under valid data lifetime". Some privacy policies

require additional operations to be executed when enforcing authorisation decision. These additional operations are *obligations*. For example, a sample privacy policy rule "the contact information can be used to complete transaction, but it must be deleted in one week" states a sample obligation – "delete customer data in 7 days". Conditions and obligations improve the expressibility for specifying privacy policies.

The PARBAC method takes purpose check into authorisation process. When a data user requests to access certain data, access control is to be checked firstly. Relevant data policy is retrieved from privacy management system. Access control is checked based on role activations in current session, the subject invoked by these roles, role-subject mapping, subject-domain mapping, and the domain-type access matrix. If the request passes the role/permission check, business purpose is then checked against data purpose, which is retrieved from data policy. If the business purpose is compliant with the data purpose, and if there are additional conditions that need to be qualified, the additional conditions will be checked. If the request can pass this step, the requested access will be granted. Otherwise, access will be denied. If obligations are found in data policy with this access, they need to be executed by obligation execution module. All data access requests should be logged in the audit trail for future auditing, no matter access requests are granted or denied.

PARBAC goes beyond traditional access control models in that it not only provides system security from an organisation's perspective, but also protects privacy from a data owner's point of view. It enables an organisation to act as a trusted custodian to protect data owner's privacy. However, this approach cannot guarantee privacy compliance, because it is built upon putting the trust in organisations who collected data. It relies on the organisation's policy to govern the use of objects. If malicious applications or users changed the policy, privacy enforcement cannot be guaranteed. Moreover, although using the role based access control mechanism as the basis of access control provides a favourable degree of subject management scalability, this approach does not provide a systematic data object model, thus reduces the data object management scalability.

2.5 The Purpose Related Approaches

As we mentioned in earlier sections, privacy policies concern more about the purposes that a data object is used for, rather than the actions that users perform on the data object. Traditional access control approaches cannot readily achieve privacy protection.

The notion of purpose should be added to and play a major role in access control methods for privacy preservation. Observing this, purpose-based access control approaches have been proposed.

This section first gives an overview of the purpose-oriented access control approach, which is the earliest to bring in the concept of purpose along with accesses for access control. This section then discusses purpose-based access control for privacy preservation, including basic purpose-based access control method, and conditional purpose-based access control method with dynamic roles.

2.5.1 Purpose-Oriented Access Control

Yasuda et al proposed the purpose-oriented access control method [YTT98a, YTT98c, YTT97, YTT98b] for object-based systems. According to this approach, a system consists of a collection of objects. These objects are manipulated only through operations supported by themselves. In addition, an operation computed in one entity invokes operations on other entities. Thus, information in one entity will flow to other entities through operation invocations. Operations are classified into four types from the information flow point of view: *non-flow, flow-in, flow-out, flow-in/out*. On receipt of a request op from an object o_2, the receiving object o_1 computes op and then sends back the response to o_2. Here, if the request and the response carry data, the data in o_1 and o_2 are exchanged. An access rule shows how each subject s manipulates an object o by an operation t of o. Access rules are defined according to operation types. They have to satisfy information flow relations among objects. The purpose of s to access o by t is modelled as what operation u of s invokes t to manipulate o. So, in the purpose-oriented access control, an access rule is specified in the form, $(s : u, o : t)$, where u shows the purpose.

This work is the first approach that we have come across in literature so far to combine access rule with the purpose for data access control. However, the purpose discussed in this approach only takes into account the legal operation of a subject on an object. It does not consider the intended usage of an object, so there is no mechanism to check the access purpose against the intended usage. This approach focuses only on information flow in nested invocations between objects rather than privacy preservation. But it inspired further efforts on using the concept of purpose for achieving privacy preserving access control.

2.5.2 Purpose-Based Approach for Privacy Protection

Byun et al proposed a privacy preserving access control approach [BBL05, BL08] based on the notion of purpose. In this approach, purposes are further divided into two categories: *intended purpose* and *access purpose*. An intended purpose is related to a data object, and specifies the intended usage of the data object. An access purpose, on the other hand, is related to data accesses. It specifies the intention for which a given data object is accessed. Intended purposes support both positive and negative privacy policies. An intended purpose consists of two components: *Allowed Intended Purposes* and *Prohibited Intended Purposes*. This structure provides flexibility to the access control model. By using prohibited intended purposes, one can guarantee that data accesses for certain purposes are never allowed. Conflicts may arise between the allowed intended purposes and the prohibited intended purposes for a same data item. These conflicts are resolved by applying the denial-takes-precedence policy, where prohibited intended purposes override allowed intended purposes. In order to simplify the management, purposes are organised according to a hierarchical structure based on principles of generalisation and specialisation, which is appropriate in common business environments.

Purpose check is used in the access control module. A user is required to state his access purpose along with his data requests. The module validates the stated access purpose against the user's authorisation to make sure that the user is indeed allowed for the access purpose. To facilitate the validation process, each user is granted authorisations for a set of access purposes, and an authorisation of an access purpose permits users to access data with the particular purpose. If the validation fails, the request is rejected without being further processed. If the validation succeeds, the module then fetches the requested data objects and checks whether or not the access purpose is compliant with the intended purpose of the data object. In order to check purpose compliance, the module must consider both the intended purpose explicitly associated with the data object and the intended purposes the data object implicitly inherited. A request is accepted if and only if the access purpose of the request is compliant with the effective intended purpose of the requested data objects.

The purpose based privacy preserving approach integrates the concept of purpose into access control. This highlights the importance of purpose in expressing privacy policies. By dividing purposes into intended purpose and access purpose, it improves the scalability of purpose management and user request management and thus makes

access control clearer. Although this method does not provide a systematic data object model, which reduces the scalability of data object management, and it lacks role authorisation mechanism, this work is a good starting point for us to analyse privacy requirements and entities involved in privacy preserving access control method. Then we can specify these entities, the relationships among these entities, and the requirement for privacy preservation.

2.5.3 Conditional Purpose-Based Access Control Approach

In purpose-based access control approach [BBL05, BL08], an intended purpose (IP) is divided into two parts: allowed intended purposes (AIP), which explicitly allows to access the data for the particular purpose, and prohibited intended purpose (PIP), which explicitly prohibits to access the data for the particular purpose. Based on purpose-based access control approach, Kabir et al [KW09, KWB10] proposed an extension by including conditional intended purpose (CIP) to extract information from PIP, which conditionally allows to access the data for the particular purpose. Conditional intended purpose means that data provider allows accessing the data for a particular purpose with some conditions. For example, data provider may consider that his income information can be used for marketing purpose by hiding his personal identification information (e.g. id or name etc.) or his income data can be revealed through generalisation, or only the first letter of name can be used for marketing purpose. Since this method supports conditional purpose and prohibited purpose, it allows data owners to specify that data should be used conditionally or should not be used for a set of purposes. This allows users to use some data with conditions. The data providers are also able to express their own privacy preferences through setting intended purpose with three levels.

The conditional purpose-based access control method utilises RBAC in a dynamic manner to achieve the compliance computation between access purpose and intended purpose. It determines the access purpose and purpose compliance in a manner based on subject attributes and context attributes of the system. Intended purposes are dynamically associated with the requested data objects during the access decision to the well-designed hierarchy of private metadata. This allows more flexible policies. By using CIP and PIP, it can assure that data access for particular purposes are allowed with some conditions or never allowed. Access is allowed only if the access purpose is included in the implementation of the intended purpose, in this case the access purpose is compliant with the intended purpose. The access is accepted with conditions if

the implementation of intended purpose includes the access purpose with conditions, in this case the access purpose is conditionally compliant with the intended purpose. The access is denied if the implementation of the intended purpose does not include the access purpose, in this case the access purpose is not compliant with the intended purpose.

With the introduction of conditional intended purpose, this method provides more options of using private information to help organisations to extract more information from data owners. It extends purpose-based access control approach to a further coverage of privacy preserving in data mining atmosphere. However, as in purpose-based access control approach, this method does not provide a systematic data object model, which reduces the scalability of data object management, and it also lacks role authorisation mechanism.

2.6 The Metadata-Based Approaches

Karjoth et al proposed the *Platform for Enterprise Privacy Practices* (E-P3P) method [KSW03] for privacy-enabled management and exchange of customer data. It falls into the category of metadata-based approaches. In metadata-based approaches, a tag or other metadata is associated with a group of data to govern how to use the data. When a subject requests to access data, the associated metadata must be checked to decide whether the operation is allowed or not. Metadata-based approaches have an important assumption: the enforcement of security and privacy policies depends on a trusted system environment [Ste97].

E-P3P introduces a viable separation of duty between four roles of a privacy system, *data subject*, *data user*, *privacy officer*, and *security officer*, The *data subjects* provide data, give consent, and select opt-in/opt-out choices. The *data users* use collected data by executing tasks of applications. The *privacy officers* design and deploy privacy policies, and the *security offices* design access control policies.

In this approach, authorisation is granted based on both access control level and privacy control level. Access control level is used to control the access of users to system applications, while privacy control level is used to control the access of applications to collected data. A privacy policy language [KS02] is proposed for formalising privacy policy and expressing restrictions on accesses to personal data. The Enterprise Privacy Architecture (EPA) [KSW02] proposed by IBM adopts E-P3P as its core technology.

Powers et al. [PAS02] proposed this approach for enterprise-wide privacy management, and defined it as five steps. These five steps can be summarised as: *define an enterprise privacy policy*, *deploy a policy to the IT system*, *record user consents*, *enforce the privacy policy*, and *generate reports of access history*.

This approach has several advantages. Firstly, the privacy enforcement is built upon access control, so it may be applied to an enterprise's existing applications. Secondly, separating privacy control and access control provides more flexibility. Depending on the efficiency requirement of a system, privacy control can be realised as real-time enforcement or conformance checking. Thirdly, setting up separate privacy office role facilitates privacy management.

However, this metadata-based approach also has its limitations. The privacy enforcement system in this approach does not specify any specific access control models. Because a privacy policy can only be enforced if it is formalised as access control rules, just like security policies, the access control model that a system adopts will affect how a privacy policy can be enforced in the system. Since different systems may adopt different access control models, to achieve efficiency, privacy control should be considered together with access control. However, this work does not propose any concrete method as to how privacy control may be incorporated with access control. Furthermore, this method only provides a framework for a privacy enforcement system. There is no detail as how the purpose of an operation is inferred.

2.7 Features of Privacy Preservation Approaches

Table 2.1 summarises the evaluation of approaches detailed in this chapter for privacy preservation according to design considerations of a privacy preserving access control approach mentioned in Section 2.1. The symbol "$\surd$" in the table denotes that relevant approach has corresponding feature. The result of evaluation shows that none of them is capable of meeting all those requirements.

2.8 Chapter Summary

This chapter summarised relevant work in privacy preserving access control approaches. We positioned the work presented in this thesis in the context of literature. In previous

Approaches	Basic Privacy Requirements		Management Scalability		
	Purpose Binding	Necessity of Data Processing	Object	Subject	Authorisation Mechanism
HPSF			√		√
Task Based	√	√	√		√
Explicit Denial				√	
PARBAC	√			√	√
Purpose Based	√				√
E-P3P	√			√	√

Table 2.1: Features of Approaches to Privacy Preservation

work nobody has incorporated basic privacy requirements, object and subject management scalability, and authorisation mechanism in privacy preserving approach. In this thesis, ensuring completeness of these aspects is an important design principle. The approach presented in this thesis helps bridge the gap between privacy requirements and data processing practices with respect to security and privacy. The next chapter will present related work in specification of privacy protection approaches.

Chapter 3

Privacy Protection Specification

3.1 Chapter Introduction

Privacy requirements are written into privacy policies. To unambiguously enforce a privacy protection approach, the privacy requirements expressed in the privacy policy should be precisely specified and enforced. It has been stated in many previous research efforts that modelling privacy requirements in early stages of system development is essential to privacy enforcement [APS02, AHKS02, BM05, BDMN06, TM01]. This chapter discusses approaches that have been proposed with regard to privacy protection specification.

3.1.1 Privacy Protection Specification Features

Three features need to be taken into consideration for a privacy protection specification approach: policy completeness, expressibility, and enforce ability. Policy completeness is used to illustrate whether the specification approach is capable of expressing various aspects of a privacy policy, such as purpose, authorisation, etc. For this feature, we will check whether the approaches have specified both object part and subject part of a privacy policy. Expressibility is used to check whether the specification tool adopted by the specification approach is capable of specifying the entities, the relationships, and the privacy requirements. This feature is required to investigate privacy properties for analysis and verification purposes. Enforce ability is used to illustrate whether the privacy rules are expressed clearly and are integrated into the access control mechanism.

3.1.2 Privacy Protection Specification Approaches

This chapter first reviews privacy policy specification methods, and then evaluates these methods against privacy protection specification requirements. Privacy policies need to be expressed using some language. We have examined three types of languages: access control policy specification languages, privacy policy specification languages, and formal specification languages. For access control policy specification languages, we investigated *Ponder* [Dam02], *eXtensible Access Control Markup Language* (*XACML*) [Org], *Authorisation Specification Language* (*ASL*) [JSSS01], and *Access Control Unit*(*ACU*) [BDVS01]; for privacy policy specification languages, we investigated *P3P* [CLM+02b] and *EPAL* [AHK+03]; and for formal specification languages, we investigated *KAOS* [DFvL91], *VDM* [Jon90], and *Z* [Spi87]. We also investigated specification approaches for security and trust to draw on the experience of specifying properties of information systems.

Among the many specification approaches for security, a particularly successful one is the *spi calculus* [AG97], which falls into the sub-domain of cryptography. It is a process calculus intended to describe and reason about the behavior of cryptographic protocols. Security properties can be expressed rigorously as statements of behavioural equivalence between processes. Butler et al. [BLPT04] examine the use of formal methods for validating and modeling trust, which is another area of interest in secure systems, and focus upon the formal specication stage of a software prototype development. The research reported in this thesis focuses on the investigation of privacy specification approaches.

In respect of privacy policy specification approaches, P3P specifies elements in privacy policies, *ACU* and *ASL* focus on specification of access control rules, and *Certain Answer Model*, *PARBAC*, *Privacy Enhanced Model*, and *Task Based Privacy Model* provide specifications of privacy preserving approaches. The rest of this chapter will present these specification approaches in detail. This chapter will conclude with an evaluation on previous work against privacy protection specification requirements, and identify the need for the development of a privacy protection approach with the help of specification.

3.2 Platform for Privacy Preferences (P3P)

The *Platform for Privacy Preferences* (P3P) [CLM+02b, Wor06] is a notable approach proposed by the World Wide Web Consortium (W3C), which is mainly used for privacy

Purpose Name	Description
current	Completion and support of activity for which data was provided
admin	Web site and system administration
develop	Research and development
tailoring	One-time tailoring
pseudo-analysis	Pseudonymous analysis
pseudo-decision	Pseudonymous decision
individual-analysis	Individual analysis
individual-decision	Individual decision
contact	Contacting visitors for marketing of services or products
historical	Historical preservation
telemarketing	Telephone marketing
other-purpose	Other uses

Table 3.1: Purposes Defined in P3P1.1 [Wor06]

protection on the Internet (see Section 1.3 on Page 29). It enables a web site to state its privacy policy in a standard machine-readable format. A P3P policy is an XML document that describes the data collection behaviours of a site. P3P provides a base schema for the data collected and a vocabulary to express the purposes, the recipients, and the retention policy. P3P predefines a set of values for its elements. Purpose is an important element in a P3P policy. Table 3.1 shows the 12 purposes that are defined in P3P1.1, which specify purposes for which data is collected or used. Additionally, 23 primary purposes are defined in P3P1.1 to provide a more detailed description of data usage under purpose $current$ in Table 3.1, and the reason for which a recipient is collecting data, as shown in Table 3.2. P3P defines recipients as in Table 3.3, which specify who will receive the collected data.

APPEL (*A P3P Preference Exchange Language*) [CLM02a] is the language designed to interoperate with P3P. It is used to describe collections of preferences on

Purpose Name	Description
account	Account and/or subscription management
arts	Arts and entertainment
browsing	Web browsing
charity	Charitable donations
communicate	Communications services
custom	Customisation
delivery	Delivery
downloads	Software downloads
education	Education
feedback	Responding to user
finmgt	Banking and financial management
gambling	Online gambling
gaming	Online gaming
government	Government services
health	Healthcare services
login	Authentication and authorisation
marketing	Advertising, marketing, and/or promotion
news	News and information
payment	Payment and transaction facilitation
sales	Sales of products or services
search	Search engines
state	State and session management
surveys	Surveys and questionnaires

Table 3.2: Primary Purpose Defined in P3P1.1 [Wor06]

Recipient Name	Description
ours	Ourselves and/or entities acting as our agents or entities for whom we are acting as an agent
delivery	Delivery services possibly following different practices
same	Legal entities following our practices
other-recipient	Legal entities following different practices
unrelated	Unrelated third parties
public	Public fora

Table 3.3: Recipients Defined in P3P1.1 [Wor06]

privacy policies between P3P agents. With this language, a user can express his preferences in a set of preference rules. These rules can then be used by his user agent to make automated decisions about whether to accept machine-readable privacy policies received from P3P enabled web sites.

P3P policies consist of common elements of privacy policies, but these are only promises rather than technical measures for policy enforcements. Web sites using P3P specification may also have to provide some further explanations about policy enforcements in a human-readable format. Furthermore, P3P does not have built-in mechanisms to verify if a given access request complies with the stated privacy policy. It is a privacy transparency mechanism rather than a privacy protection specification.

3.3 Access Control Unit

Bonatti et al developed the *Access Control Unit* (*ACU*) [BDVS01] to specify privacy preserving access control rules. This approach is to enforce access control on data archives. A data archive maintains collected data, called *datasets*. In addition, an archive also maintains a collection of *metadata*, representing information associated with datasets. Data archives need to make their data selectively available to others. *ACU* allows data publishers to specify by whom, how, and under which conditions specific data can be accessed.

The *ACU* is typically used to express access rules. It characterises $subjects$, $actions$, $objects$, and possibly, $conditions$. The action attribute is characterised through operation names. Subjects and objects are specified by stating an identifier firstly,

then specifying a given elementary value in the corresponding domain. Conditions are specified by constraints to be satisfied for the request to be granted. The request made by a data user is characterised by a triple: $\langle user, project, purpose \rangle$. The purpose attribute states the reason for which data are being requested and will be used. A project is a named activity registered at the data publisher, for which different users can be subscribed, and a project may have one or more purposes. Two kinds of access rules - $authorisations$ and $restrictions$ - are specified using *ACU*. Authorisations specify permissions for data accesses. They have the form $\langle subjects \rangle$ CAN $\langle actions \rangle$ $\langle objects \rangle$ [IF $\langle conditions \rangle$], where $subject$, $actions$, and $objects$ identify the requests to which the authorisation applies, and $conditions$ is a boolean expression of conditions whose satisfaction authorises the access. Conditions can also be included in the expressions specifying the $subjects$ and $object$ for the rule. Restrictions specify requirements that must be satisfied for an access to be granted. They have the form $\langle subjects \rangle$ CAN $\langle actions \rangle$ $\langle objects \rangle$ ONLY IF $\langle conditions \rangle$, where $subjects$, $actions$ and $objects$ identify the requests to which the restriction applies, and $conditions$ is a boolean expression of conditions that every request to which the restriction applies must satisfy. For a given request, lack to satisfy any of the conditions in restrictions that apply to it implies that the request will be denied. An access request is considered to be authorised if at lease one of the authorisations that applies to the request is satisfied.

The following are examples of security requirements and corresponding ACU rules to enforce them.

- Everybody can access Free_Datasets.

  Users CAN access Free_Datasets

- Access to datasets not in Free_Datasets allowed *only* to UK citizens.

  Users Can access data WITH NOT dataset IN Free_Datasets ONLY IF user/citizenship=‘UK’

The Access Control Unit component mediates all access requests to datasets/metadata. It evaluates the access requests against access rules. For each request received, the ACU component first determines all the rules that apply to the request. These are rules for which the action field is equal or is an abstraction of the action in the request, and whose subject expressions are satisfied by the subject of the request. This rule collection process is followed by a conditions packing and evaluation process. If

required conditions are satisfied, the access is granted, otherwise, it is denied. Consider the access control rules in example above and a request by user Bob to download dataset1 for Commercial purpose within project Marketing. Suppose that dataset1 is a Free_Datasets. The access will be granted with no condition according to the first rule.

The *ACU* specification is simple and flexible. It can be used to express protection requirements that need to be enforced. However, the access control rules it specified are simply adding a condition attribute to traditional access control rules, so they cannot effectively describe privacy requirements. Moreover, *ACU* is incapable of specifying object and subject structures.

3.4 Authorisation Specification Language

Karjoth and Schunter proposed a formal model, including *purpose*, *obligation*, and distributed administration, for privacy preservation [KS02]. The authorisation specification language (ASL) was developed to specify the access rules. The elements in the model are as follows: $principals$, $data$, $purposes$, $actions\ and\ information\ sharing$, $conditions$, and $obligations$. The data system of the privacy model consists of users, groups, data to be accessed, access purposes, and access modes. The authorisations are divided into two categories: authorisation subjects consisted of users, processes, groups and purposes; and authorisation objects consisted of objects, types, and purposes.

The model specifies different types of rules. These rules are inserted by the security administrator. The cando rules represent direct authorisations. Other rules are defined to represent authorisations derived from the system using logical rules of inference: der-cando rules describe the propagation of information, do rules define conflict resolution strategies and error rules define integrity constraints. The privacy policies are specified by using these authorisation rules.

The following are examples of privacy requirements and corresponding ASL rules to specify them.

- Data of type medications can be read for purpose treatment.

  cando(medications, treatment, +read, [], cpo) ←

- Task diagnosing can act for purpose treatment.

  cando(treatment, diagnosing, +activate, [], cpo) ←

- User Bob is authorised to perform task diagnosing.

  cando(diagnosing, Bob, +execute, [], sso) $\leftarrow$

The Privacy Officer provided the first two authorisation rules; the Security Officer gave the third authorisation. In addition, when a patient accepted the enterprise's privacy policy, he implicitly sanctioned the first authorisation. By adding the following rule, the Privacy Officer strengthens the privacy policy stating that the task above serves exactly one purpose.

error $\leftarrow$ certified(diagnosing, p) $\wedge$ certified(diagnosing, $p^{'}$) $\wedge\ p \neq p^{'}$

The proposed model can be used as the basis for an internal access control system to handle received data with regard to privacy preservation. The data owner providing his personal data has the assurance that the organisation receiving the data will handle it according to stated privacy policy. The organisation can also be sure that its data processing practices are not in conflict with the privacy promises they made. The authorisation specification language used in this approach is capable of expressing privacy policy as authorisation rules. It is simple and expressive. But as some components, such as data objects and subjects, are not formally defined, the specification of this model is incomplete.

3.5 Privacy Model using Certain Answer

Stouppa et al proposed a data privacy model using certain answer [SS06]. In the model, a formal definition of privacy problem is presented based on the notion of *certain answer*. Personal data stored in information system takes the form of a *privacy condition* which is a set of queries. Moreover, the public information is given in terms of a *view instance* and *background knowledge*. A view instance consists of queries and their answers, while background knowledge includes additional facts about the system. Background knowledge is provided for better understanding of the data in the view. The privacy problem is then to decide whether any of the queries in private condition can be inferred from the view instance and the background knowledge. In order to state the privacy problem, the notion of *certain answer* was proposed. The certain answers of a query are those answers that are returned by the query in every 'possible' instance. Data privacy is preserved for a query with respect to the provided public knowledge if there are no non-negative certain answers to the query with respect to that knowledge, that is, if the certain answer to it is either the empty set or negative ('none' or 'no').

ALLOW [*Data User*]

to perform [*Operation*] *on* [*Data Type*]

for [*Purpose*] *provided* [*Condition*]

carry out [*Obligation*]

Figure 3.1: A Privacy Policy Rule in PARBAC

Formal definitions for both ontology and query answering on it are given. The ontology is defined as a set of first-order sentences, and query answering is done through entailment. The formal model of data privacy is presented using certain answers, and shows that the privacy preservation can be modelled by logical entailment. For example, when data privacy is applied on relational databases with conjunctive queries, since background knowledge consists of a relational schema with constraints imposed on it, data privacy for this setting is decidable in polynomial time.

The privacy model using certain answers consists of the formalisation of ontology and the constraints of query answering. The privacy preservation problem here is actually to check whether a given view instance leaks information about the data rather than to control user's manipulation on the data. This only considers the data object part of a privacy policy. Moreover, the privacy model using certain answers cannot express basic privacy requirement, such as purpose binding, which makes the specification of privacy policy incomplete.

3.6 Privacy-Aware RBAC Model

Privacy-Aware Role-Based Access Control (PARBAC) model [He03] extended RBAC model for enforcing privacy policies within an organisation. As described in Section 2.4.3, it is based on RBAC, Domain-Type Enforcement (DTE), business purposes, and data usage policies. The model consists of a number of authorisation components and the relationships among the components. The main authorisation components in the PARBAC model are (1) $User$, (2) $Role$, (3) $Subject$, (4) $Domain$, (5) $Purpose$,

(6) $Object - Type$, and (7) $Object - Policy$. The relationships of components are mapping from a source entity to a target entity with either a many-to-many, many-to-one, one-to-many, or one-to-one relation. The model also consists of some constraints that are used to express security requirements. One of the important modules in PARBAC model is customer privacy preferences management. When customer data are collected online or offline, customers can specify their privacy preferences on how the organisation can use the data. When a data user requests to access certain privacy protected data resource, access control is to be checked firstly and then the corresponding data policy is retrieved from privacy management system. A typical privacy policy rule expressed in the PARBAC model has the format as in figure 3.1.

The PARBAC model lists the components and the relationships among components. There are many partial order relations, such as roles, tasks, purposes, and many-to-many relations. However, these relations and interactions between these relations are not formally defined, which makes the specification of the model incomplete. Moreover, there is no clear way to specify conditions in PARBAC.

3.7 Privacy-Enhanced Access Control Model

Fei Xu et al proposed a privacy enhanced access control model [XHWX09, XCHW09]. In the model, the entities of objects and subjects are defined, and the privacy-concerning subject is introduced into the model. Privacy-concering subjects is a subset of the subjects whose privacy needs to be protected when any of the objects is accessed. Privacy-concerning subjects with respect to an object is a subset of the privacy-concerning subjects whose privacy could be violated when any access to the object is executed. Since an object may contain private information about more than one subject, all such subjects can be the privacy-concerning subjects with respect to the object. Privacy access rights are a set of two-tuples: $\{< right, condition >\}$ in which $right$ represents a specific type of access that a subject can execute on an object and $condition$ is a Boolean expression stating the condition under which the preceding access right can be authorised. When the Boolean expression is true, the access right can be authorised by the system.

The privacy-enhanced access control model is expressed using a three-dimensional access control matrix. It is an enhancement of the traditional two-dimensional access control matrix with the third dimension representing the set of privacy-concerning subjects. Any privacy access right in the matrix indicates the access right that subject has

on object subject to privacy control by privacy-concerning subjects. Each entry in the matrix could be empty or could include one or more privacy access rights that the corresponding subject has on the object with the privacy control by privacy-concerning subject.

The privacy-enhanced access control introduces the notion of privacy-concerning subjects, and integrates privacy as a new dimension into the traditional access control matrix model. This can better describe and support data owner requirements for protecting private information when access control is used for making access decisions. However, this model is still an abstract model and is in a very primitive stage. Many implementation issues need to be considered, such as how to specify purposes in privacy requirements, how to solve inconsistencies among the conditions in the privacy access right, and how to implement the three-dimensional access control matrix in real systems in an effective way, etc.

3.8 Task-Based Privacy Model

Fischer-Hübner et al proposed the task-based privacy model [FHO98]. It is defined as a state machine model. As described in Section 2.3, it consists of *state variables*, *invariants*, *constraints* (*privacy properties*) and *state transition functions*. State variables define the security-relevant model elements within the state machine model. They are needed to formally define the privacy policy and the system states. The state variables defined in the task based privacy model include *subjects*, *objects*, *tasks*, *purposes*, *transformation procedure*, and so on. Invariants define conditions for a system state to meet specific privacy principles. To enforce certain privacy policy, the invariants must be fulfilled in each system state. Constraints are added as properties of state sequences to formulate privacy principles, such as necessity of data processing and purpose binding. State transition functions describe all possible changes of state variables, and are defined for the actions in the data processing system. Privacy policies are then specified as privacy rules using the components in the model. The rules are based on the concept of task. A subject shall be allowed to access an object only by performing a task on the object. The tasks have to be defined for each application. The privacy rules are related to requests and therefore are used as access control information in their access control framework.

The task-based privacy model gives formal definitions of components and relationships in an privacy protection approach. It expresses privacy requirements in privacy

Approaches	Policy Completeness		Expressibility	Enforce Ability
	Object Part	Subject Part		
P3P	√	√		
ACU	√	√	♯	√
ASL	√	√	√	♯
Certain Answers	√		♯	
PARBAC	√	√	♯	♯
PEAC		√		
Task-Based	√	√	♯	√

Table 3.4: Features of Privacy Policy Specification Approaches

policies as formal invariants and constraints on these components. However, since it does not provide a clear structure of data objects and subjects, the formal description of the components is not completed. Moreover, the validation mechanism is not formally specified.

3.9 Features of Privacy Protection Specifications

Table 3.4 illustrates the evaluation of privacy policy specification approaches discussed in this chapter according to features of a specification method mentioned in Section 3.1. The symbol "√" in the table denotes relevant approach has corresponding feature, while the symbol "♯" denotes that relevant approach has corresponding feature, but the specification is incomplete. The result of approaches evaluation shows that none of them meets all those featured requirements.

3.10 Requirements Analysis and Specification

With the background knowledge of information privacy, we have reviewed privacy preserving approaches and privacy protection specifications. Traditionally, privacy protection specification is isolated from requirements analysis. This isolation presents a number of problems.

- Incomplete privacy protection method

  The lack of requirements analysis often leads to incomplete privacy protection approach. For example, without the help of requirements analysis, *Certain Answers* specification only specifies privacy protection from the objects' point of view. This makes it incapable of expressing privacy requirements from the subject's perspective, and therefore leads to the design of an incomplete privacy protection method.

- Insufficient privacy property analysis and verification

  Without the specification of entities, relationships, and privacy requirement, the privacy properties of a proposed privacy preservation approach cannot be analysed in full. This makes it impossible to judge whether the proposed approach has achieved the privacy preservation goal it has targeted.

- Misalignment between privacy promises and practices

  Without a proper specification and authorisation mechanism, the privacy promises of an organisation may not be properly implemented in real data processing practices. For example, since P3P does not provide any technical measure on authorisation, it cannot help verify whether the privacy practices are consistent with published privacy policies.

The privacy violations caused by inconsistency between the privacy requirements and the data processing practices motivated us to investigate privacy protection specification. Based on the above investigations, we argue that privacy protection specification should be developed together with requirements analysis. As to privacy requirements, we believe that data owners' privacy-related decisions are highly individual. Therefore, we argue that privacy protection requires an approach which enables disclosure of private information under data owners' preferences, in order for the data owners to gain the benefits of accessing services and using applications at their desired levels of openness.

This work proposes purpose based access control as the mechanism to achieve information privacy with the consideration of data owners' preferences. Purposes can be divided into intended purposes and access purposes. The intended purposes specify the intended usage of data objects, and the access purposes specify the intentions for which a given data object is accessed. To ensure that a data object is used only for its intended usage, the access purpose should be compliant with the data object's intended

purpose. Privacy requirements are analysed and verified with the help of the VDM specification. The complete specification consists of entities, relationships, invariants, model operations and their proof obligations. In addition, design considerations for successful application of the proposed approach in distributed computing environments are investigated.

3.11 Chapter Summary

This chapter investigated relevant work on privacy protection specification approaches. A privacy protection specification should consider the policy completeness, the expressibility, and the integration with enforcement mechanisms. The privacy protection specification approaches that we investigated are analysed against those three conditions. Based on the review of privacy protection approaches and privacy protection specifications, we identified the need for the development of a privacy protection approach with the help of specification. The purpose-based access control model will be presented in next chapter to illustrate that it meets the requirements set in Section 2.1 and Section 3.1.

Chapter 4

A Purpose-Based Access Control Method

4.1 Chapter Introduction

This chapter presents the purpose based access control method that enforces basic legal privacy requirements, such as *purpose binding* and *necessity of data processing*. The specification is developed along with the requirements analysis. The entities, their relationships, privacy requirements, and model operations and their proof obligations in the privacy protection approach are specified, which form the basis of the Purpose Based Access Control Model.

This chapter is organised as follows: Section 4.2 illustrates the design considerations for the purpose based privacy preserving approach and its specification; Section 4.3 specifies entities in the model, including data objects, roles and users, purposes, and accesses; Section 4.4 specifies privacy requirements required for a privacy protection approach; Section 4.5 describes the model rules, and presents their satisfiability proof obligations; Section 4.6 describes the access control mechanism in this approach; Section 4.7 analyses the features of this approach; and concluding remarks are included in Section 4.8.

4.2 Design Considerations

The requirements for privacy preserving approach and its specification are set in Section 2.1 and Section 3.1 respectively. To meet those requirements, when we develop our approach and the corresponding specification, the aspects in the following three

sections are considered accordingly.

4.2.1 Purpose As A Base Construct

In previous chapters, the concept of purpose has been highlighted. In respect of privacy preserving requirements, data owners' decisions about privacy protection are highly individual. So we argued that in order for the data owners to gain the benefits of accessing services and using applications while at the same time to have control over their personal information at their desired levels of openness, privacy protection requires an approach that reflects data owners' preferences over the disclosure of private information.

The concept of purpose is integrated into our privacy protection approach. Purposes can be divided into intended purposes and access purposes. The intended purposes specify the intended usage of data objects, while the access purposes specify the intentions for which a given data object is accessed. To ensure that a data object is used only for its intended usage, the access purpose should be compliant with the data object's intended purpose. Since purpose closely reflects the comfort levels of an individual about his own information, it is introduced as a basic construct of our privacy protection approach.

4.2.2 VDM Specification

VDM is chosen as the specification approach in this investigation, since it provides adequate expressive possibilities for specification and verification of privacy properties, and relatively simple satisfiability proof method. In VDM, objects are specified in terms of a model, such as data object model, purpose model. The specification of the purpose based access control approach takes the form of *operations* on a *state* which defines a class or set of valid states. Well-formed *privacy invariants* are used to limit the defined class further. Operations are specified using pre-condition predicates and two-state post-condition predicates. All operations preserve any privacy invariant that may apply. Operations may change the value of the state as long as the new value is also a valid state. The link is established between the requirements analysis and the design of the states.

4.2.3 Privacy Protection Principles

In the introduction of this thesis, it lists general privacy protection principles, such as OECD *Guidelines for Data Protection* and FTC *Fair Information Practice Principles*. Based on the analysis of these principles and many other requirements in legislative acts, the thesis lists the most essential privacy protection principles in Section 1.2.3 as summarised in Simone Fischer-Hübner's work, namely: *lawfulness and fairness, purpose specication and purpose compliance, necessity of data collection and processing, notication and access rights of the data owner, security and accuracy*.

The principle of lawfulness and fairness just provides a general moral principle, and the principle of notication and access rights of the data owner, and security and accuracy relate more to real implementation mechanisms of information systems. Since this research aims at preserving privacy using access control mechanism, the specication in this thesis focuses on the aspects of purpose specification and purpose compliance, and necessity of data collection and processing. Furthermore, in the research reported in this thesis, we find that the entities of role attributes and system attributes are sufficient to represent the conditions in our analysis, so we currently won't specify the entity of obligation for the user and role model.

4.3 Model Entities

This section outlines the basic framework of our privacy preserving access control model, and presents essential concepts, denitions of the main entities, and the formal specications of mapping functions and the access granting rules. Three entities are usually used in a basic access control system: *subjects*, *objects*, and *operations*. For the system to be able to perform privacy-preserving access control, entities that can be used by data owners to state their privacy preferences and by the system to enforce the privacy requirements should be included, e.g. the privacy sensitive levels of data objects, the consent of data owner, the intended data usages, and the purposes of data accesses. Since privacy policies are concerned with the purposes for which data objects are used rather than the actions that subjects perform on data objects, traditional access control models can not readily achieve privacy preservation. The notion of purpose should play a major role in access control model in order to preserve privacy. Therefore, we will focus on the entity of purpose in our privacy protection model.

In this section, the entities of our privacy protection model and system state are defined. We will first define a suitable structure for representing data objects, then

Object Types	Content
Registration Data	Administrative and demographic information about a patient
Admission Data	Administrative information about a patient
Treatment History	Treatment record of a patient
Diagnosis	Diagnosis data
Prescription	Prescription information
Treatment Suggestions	Instructions for treatment
Billing Data	Billing information about a patient
Statistics	Statistical data

Table 4.1: Object Types in a Medical Care Scenario

we will present the RBAC model with the extension of conditional role, we will then define the entity of purpose model, and then we will define the entities for accessing data objects, and finally we will specify the system state based on the definitions of entities.

4.3.1 An Object Data Model

An object data model, which gives a suitable structure for representing data objects, is defined in this section. In each organisation, there are a set of data objects. A data object is used to denote a piece of information. In this approach, data objects are organised using object type information. An object type corresponds to a set of data objects that satisfy some common properties. For example, as to the data objects collected from a patients in a medical care environment as mentioned in Section 1.2, which are shown in Table 4.1, we can see that some data objects belong to the object type of registration data, some data objects belong to the object type of treatment history, some data objects belong to the object type of diagnosis, and so on. Data objects are classified into different object types, because it is much easier to define and administer intended usages and necessary accesses for object types instead of defining them individually for each single data object, thus improves the object management scalability. These entities are defined as follows.

Let *Object* denote the set of objects. An object type is a set of objects, so the set of object types can be denote as $Object$-set (this is a VDM set type definition).

Type Attributes
Purpose
Retention
Service opt_in
Service opt_out
...

Attribute Values
Admin, Diagnosing ...
1 day, 1 week, ...
true, false
true, false
...

Table 4.2: An Example of Object Type Attributes and Attribute Values

Next, we define object type attributes and attribute values to specify the properties of object type.

Definition 1 *[Object Type Attributes] denoted as $TypeAttr$, are a set of attributes associated with an object type, and these attributes describe the properties for the collection of, and access to, this type of objects.*

Table 4.2 gives an example of object type attributes in a data model.

Definition 2 *[Object Type Attribute Values] denoted as $AttrValue$, are a set of possible values for the object type attributes.*

Table 4.2 also gives an example of possible attribute values associated with object type attributes.

The object data model is concerned with how data objects are organised and how they are associated with type attributes. The object data model in our system is defined as follows.

Definition 3 *[Object Data Model]*

$$
\begin{aligned}
ObjectDataModel :: object &: Object\text{-set} \\
type &: (Object\text{-set})\text{-set} \\
typeAttr &: TypeAttr\text{-set} \\
attrValue &: AttrValue\text{-set} \\
TypeOf &: Object \rightarrow Object\text{-set} \\
AttrOf &: Object\text{-set} \rightarrow TypeAttr\text{-set} \\
ValueOf &: Object \times TypeAttr \rightarrow AttrValue
\end{aligned}
$$

inv (mk-$ObjectDataModel$(*o, t, ta, av, To, Ao, Vo*)) $\underline{\triangle}$
(dom $To = o \wedge$ rng $To \subseteq t$) $\wedge$
(dom $Ao = t \wedge$ rng $Ao \subseteq ta$-set) $\wedge$
(dom $Vo = o \times t \wedge$ rng $Vo \subseteq av$)
The fields of $ObjectDataModel$ state that

1. *object* is a set of objects
2. *type* is a set of object types
3. *typeAttr* is a set of type attributes
4. *attrValue* is a set of attributes values
5. $TypeOf : Object \rightarrow Object$-set is a total function giving the type associated with each object
6. $AttrOf : Object$-set $\rightarrow TypeAttr$-set is a total function giving the type attributes associated with each type
7. $ValueOf : Object \times TypeAttr \rightarrow AttrValue$ is a total function giving the value of the attributes associated with objects.

Some notes on the syntax of the VDM specification above: (1) $ObjectDataModel$ is a composite type. The composite type is defined as: $Name :: ...$ (2) A composite type has a number of fields, and each such field has a value. (3) The signature of a function is written with the domain and range sets (dom and rng on a function return the domain and range sets respectively) separated by an arrow, as in $TypeOf : Object \rightarrow Object$-set. (4) A Greek delta is combined with the equality sign to give the definition symbol ($\underline{\triangle}$). (5) Data type invariants (inv) are truth-valued functions which can be used to record restrictions on composite types. (6) A $make - function$, as mk-$ObjectDataModel$, when applied to appropriate values for the fields, yields a value of the composite type. A make-function is specific to a type, and its name is formed by prefixing mk- to the name of the type.

The data structure for object data model *OM* in our system, of type $ObjectDataModel$, can then be represented as a tuple $\langle$*object, type, typeAttr, attrValue, TypeOf, AttrOf, ValueOf*$\rangle$.

In this section, we defined the object data model for representing data objects in our system. Next we will define the structure for representing the subjects in our system.

4.3.2 Users and Roles

The purpose-based access control approach of [BBL05, BL08] extends the RBAC model with the concept of conditional role, which is based on role attributes and system attributes. In this section, formal definitions of role attributes, system attributes, and conditional roles are presented.

First, we specify the entities of user and role in the basic RBAC model.

Users are the active entities in a system, e.g. the staff in a medical care scenario.

Let $User$ denote the set of users.

The roles in a system reflect the responsibilities of positions or job descriptions in the context of an organisation, e.g. therapist, registration staff, or billing staff in a medical care scenario.

Let $Role$ denote the set of roles.

A user may be assigned several roles and a role may be assigned to several users.

$UserRole$: $User \leftrightarrow Role$ is the relation between users and roles.

A user may be assigned with many roles, but the user may not exercise all his roles at the same time. The roles that a user is currently exercising are "active" roles.

Active Roles AR: $User \rightarrow Role$-set is a function that returns the roles for which a user is active.

Because the existing role definitions are predefined for access permission assignments, they may not adequately specify the set of users to whom we wish to grant an access purpose. The concept of *conditional role* was then introduced. It is based on the notion of *Role Attributes* and *System Attributes*. Next we specify them accordingly.

Definition 4 *[Role Attributes] denoted as $RoleAttr$, are a set of properties of roles related to the grant of access purpose.*

Every role $r \in Role$ is associated with a set of role attributes, e.g. the specialty of therapists in a medical care scenario.

RoleAttrOf: $Role \rightarrow RoleAttr$-set is a function that returns the set of role attributes of a role.

Let *RoleAttrValue* denote the set of possible role attribute values.

RoleAttrValueOf: $Role \times RoleAttr \rightarrow RoleAttrValue$ is a function giving the value of role attributes associated with a role.

For an access control system, system attributes are used to describe the properties of a system context. For example, the working hours within a hospital is a system attribute.

Definition 5 *[System Attributes] denoted as $SysAttr$, are properties about the context of access control system.*

The values of system attributes specify the conditions of the access control system. Let $SysAttrValue$ denote the set of all possible system attribute values.

SysAttrValueOf: $SysAttr \rightarrow SysAttrValue$ is a function giving the value of the system attributes in a system.

With role attributes and system attributes, we can now define the conditional role.

Definition 6 *[Conditional Role] refers to a role with some conditions attached to it.*

$$CondRole :: r : Role$$
$$cond : RoleAttrValue \times SysAttrValue \rightarrow \mathbb{B}$$

where $\mathbb{B}$ is the boolean set, $\mathbb{B}$ = {true, false}, and $cond$: $RoleAttrValue \times SysAttrValue \rightarrow \mathbb{B}$ is a truth-valued function.

$CR : CondRole$-set is used to hold the set of conditional roles in a system.

Current Conditional Role CCR: *User* $\rightarrow$ *CR* is a function that returns the conditional role the user currently exercises.

Using the entities defined previously, we can now define the role model for our system.

Definition 7 *[Role Model]*

$$\begin{aligned} RoleModel :: role &: Role\text{-set} \\ user &: User\text{-set} \\ UserRole &: User \leftrightarrow Role \\ AR &: User \rightarrow Role\text{-set} \\ roleAttr &: RoleAttr\text{-set} \\ roleAttrValue &: RoleAttrValue\text{-set} \\ RoleAttrValueOf &: Role \times RoleAttr \rightarrow RoleAttrValue \\ sysAttr &: SysAttr\text{-set} \\ sysAttrValue &: SysAttrValue\text{-set} \\ SysAttrValueOf &: SysAttr \rightarrow SysAttrValue \\ CR &: CondRole\text{-set} \\ CCR &: User \rightarrow CR \end{aligned}$$

The data structure for role model *RM* can be represented as a tuple ⟨*role, user, UserRole, AR, roleAttr, roleAttrValue, RoleAttrValueOf, sysAttr, sysAttrValue, SysAttrValueOf, CR, CCR*⟩.

4.3.3 Purpose

Data is collected for certain purposes. For example, for medical care, data may be collected for registration or diagnosing. Each data access also serves a certain purpose. It is necessary to determine purposes for which data is collected and purposes of data accesses. The concept of purpose has been highlighted in previous discussion. In respect of privacy preserving requirements, data owners' decisions about privacy protection are highly individual. So we argued that in order for the data owners to access services and use applications while at the same time to have control over their personal information, privacy protection requires an approach that reflects data owners' preferences over the disclosure of private information. So the concept of purpose is integrated into our privacy protection approach. In this section, the entity of purpose is formally defined.

Definition 8 *[Purpose] denoted as $Purpose$, is the intention of data collection or data access.*

Purposes are organised in a tree structure, which is called purpose tree. Let P_T denote the purpose tree. Each node represents a purpose in $Purpose$, and each edge represents a hierarchical relation between two purposes. These relations are defined below.

In respect of purposes, some are general, and some are special. There are some relationships among them. The purposes are organised into purpose tree according to these relationships. Next we define relationships among purposes.

The nodes in a purpose tree can be classified into general or special according to the relationships among the nodes.

Definition 9 *[Specialisation (Generalisation)] If $p_1,\ p_2$ are two nodes in a purpose tree, then we say p_2 is a specialisation of p_1 (or p_1 is a generalisation of p_2) if there exists a downward path from p_1 to p_2.*

$Specialisation$: $Purpose \times Purpose \rightarrow \mathbb{B}$ is a truth-valued function that characterises the specialisation relation.

$Generalisation$: $Purpose \times Purpose \rightarrow \mathbb{B}$ is a truth-valued function that characterises the generalisation relation.

We have specified the purposes and the relationships among purposes. Next, we specify the purposes according to data processing stages, including data collections

and data accesses. Purposes, depending on their association with objects and subjects, may be categorised into intended purposes or access purposes, respectively.

Definition 10 *[Intended Purpose] is the specified usages for which the data objects are collected.*

Intended purpose specifies the property of data objects.

IP: $object(OM) \cup type(OM) \rightarrow Purpose$-set is a function that returns intended purposes of a data object or type.

Here, $object$ and $type$ are defined in object data model *OM*, *Purpose* is the set of purposes.

Definition 11 *[Access Purpose] is intentions for accessing data objects.*

Access purpose specifies the property of data accesses.

$AuthorisedAccessPurpose\ AAP$: $CR(RM) \rightarrow Purpose$-set is a function that returns authorised access purposes.

The purpose model in our system is defined as follows.

Definition 12 *[Purpose Model]*

$$
\begin{aligned}
PurposeModel :: purpose &: Purpose\text{-set} \\
Specialisation &: Purpose \times Purpose \rightarrow \mathbb{B} \\
Generalisation &: Purpose \times Purpose \rightarrow \mathbb{B} \\
IP &: object(OM) \cup type(OM) \rightarrow Purpose\text{-set} \\
AAP &: CR(RM) \rightarrow Purpose\text{-set}
\end{aligned}
$$

The data structure of purpose model *PM* can be represented as a tuple ⟨*purpose, Specialisation, Generalisation, IP, AAP*⟩.

4.3.4 Requests, Transactions, and Accesses

This section specifies the entities for accessing data objects, namely, requests, transactions and accesses.

Definition 13 *[Request]*

$Request :: obj : object(OM)$
$ap : purpose(PM)$

When a conditional role *cr* wants to access an object obj, it makes a request for the data object with a particular access purpose *ap*. The request is denoted as a 2-tuple $\langle obj, ap \rangle$. For example, the request from a GP to access treatment history for the purpose of diagnosing has the form of $\langle treatment\ history, diagnosing \rangle$.

We use $Req : Request$-set to denote the set of requests in a system.

Current Request CReq: $CR(RM) \rightarrow Req$ is a function that returns the request currently presented.

Definition 14 *[Transactions] denoted as Transaction, are the executions or procedures to perform a request.*

To ensure an object is accessed in a controlled manner, only specified transactions may be allowed. For example, the diagnosing request consists of three transactions: reading treatment history, analysing medical test results, and appending new diagnosis to the treatment history.

Current Transaction CT: $CR(RM) \rightarrow Transaction$ is a function that returns the transaction currently being performed.

Authorised Transactions AT: $Req \rightarrow Transaction$-set is a function returns the authorised transactions for a request.

Next we define entities about accesses in our system. Model entities related to object accesses are access modes, necessary access, and current access.

Definition 15 *[Access Modes] are the modes of accesses performed on data objects.*

Let *AccMode* denote the set of access modes. $AccMode = \{create, read, write, append, delete\}$

$Mode$: $AccMode$-set denotes the set of access modes in a system.

Definition 16 *[Necessary Accesses] are the accesses that are needed to achieve an access purpose.*

For access purpose, it has to be defined in advance what accesses are needed to achieve that access purpose.

$NecAcc :: ap : Purpose$
$tp : type(OM)$

$$trans : Transaction$$
$$x : Mode$$

NA: $NecAcc$-set denotes the set of necessary accesses.

Definition 17 *[Current Accesses] are accesses that a conditional role is performing.*

$$CurAcc :: cr : CR(RM)$$
$$obj : object(OM)$$
$$x : Mode$$

CA: $CurAcc$-set denotes the set of current accesses.

Then, we can define the access model in our system.

Definition 18 *[Access Model]*

$$AccessModel :: Req : Request\text{-set}$$
$$CReq : CR(RM) \rightarrow Req$$
$$Trans : Transation\text{-set}$$
$$CT : CR(RM) \rightarrow Trans$$
$$AT : Req \rightarrow Trans\text{-set}$$
$$Mode: AccMode\text{-set}$$
$$NA : NecAcc\text{-set}$$
$$CA : CurAcc\text{-set}$$

The data structure of access model *AM* can be represented as a tuple ⟨*Req, CReq, Trans, CT, AT, Mode, NA, CA*⟩.

Having defined the entities in our purpose-based access model, the system state can be defined.

4.3.5 The State of System

Entities are specified in terms of models in previous sections. The specification of our approach takes the form of an operation on a state which defines a class or set of valid states. Well-formed data type invariants are used to limit the defined state further. The formalisation of the model consists of the specification of system state. System state consists of the state variables corresponding to the components defined in previous sections: OM, RM, PM, AM.

A state consists of fields list, invariants, and initialisation. The state space, without invariant and initialisation condition as yet, is written as follows:

state PPS of
$OM : ObjectDataModel$
$RM : RoleModel$
$PM : PurposeModel$
$AM : AccessModel$
inv ...
init ...
end

The initialisation condition on the state is defined as:
init $\sigma \triangleq \{\sigma.object(OM) = \{\} \wedge \sigma.type(OM) = \{\} \wedge$
$\sigma.purpose(PM) = \{\} \wedge \sigma.AAP(PM) = \{\mapsto\} \wedge$
$\sigma.Req(AM) = \{\} \wedge \sigma.Trans(AM) = \{\} \wedge$
$\sigma.CA(AM) = \{\} \wedge \sigma.NA(AM) = \{\}\}$

which initiates the entities of object, object type, purpose, request, transaction, current access, and necessary access with empty sets.

In this section, the entities in purpose-based access control model and the system state have been introduced. Then we are able to specify privacy requirements in privacy policy. We will specify the state invariants corresponding to the requirements in next section, and we will also specify the operations in the model.

4.4 Privacy Invariants and Constraints

In this section, a way to specify privacy requirements with the help of Purpose Based Access Control Method is described. We will express privacy requirements in the privacy policy. The following privacy policy was stated in [FHO98]:

> *A subject may only have access to personal data if this access is necessary to perform its current task, and only if the subject is authorised to perform this task. The subject may only access data in a controlled manner by performing a transformation procedure, for which the subject's current task is authorised. In addition, the purpose of its current task must correspond to the purposes for which the personal data was obtained or consent must be given by the data subjects.*

There are two important aspects of data access that should be protected by a privacy-preserving access control system according to this policy: necessity of data accesses

and purpose binding of accesses to data.

Using the entities we defined before, and according to the process of data access, we express privacy requirements in the privacy policy stated above in following invariants (we place the symbol "′" behind a state variable to refer to the variable in the new system state):

We define invariants through the process of data access. First, we define the invariants for the creation of data objects.

(a) Data Collection Invariants

(a1) A data object can be created if and only if it is necessary for the conditional role to fulfill its current request.

Given two successive system states $v, v^{'}$,
$v = (OM, RM, PM, AM)$,
$v^{'} = (OM^{'}, RM^{'}, PM^{'}, AM^{'})$,
$(v, v^{'})$ satisfies privacy invariant-(a1), iff
$\forall\, cr \in CR(RM),\ type_j \in type(OM), ap \in purpose(PM)$:
$obj \notin object(OM) \wedge \langle obj, ap\rangle = CReq(AM)(cr)$
$\wedge \langle ap, type_j, CT(AM)(cr), create\rangle \notin NA(AM)$
$\Longrightarrow obj \notin object(OM^{'}) \vee TypeOf(OM^{'})(obj) \neq type_j$

This invariant specifies the necessity of data object creation.

(a2) A data object may be created if and only if the purpose of a conditional role's current request match the purpose of the object's type.

Given two successive system states $v, v^{'}$,
$v = (OM, RM, PM, AM)$,
$v^{'} = (OM^{'}, RM^{'}, PM^{'}, AM^{'})$,
$(v, v^{'})$ satisfies privacy invariant-(a2), iff
$\forall\, cr \in CR(RM),\ type_j \in type(OM), ap \in$
$purpose(PM)$:
$obj \notin object(OM) \wedge \langle obj, ap\rangle = CReq(AM)(cr)$
$\wedge\, ap \notin IP(PM)(type_j)$
$\Longrightarrow obj \notin object(OM^{'}) \vee TypeOf(OM^{'})(obj) \neq type_j$

This invariant specifies the purpose compliance of data object creation.

Next, we define invariants for the authorisation of conditional role.

(b) Role Authorisation Invariants

These invariants specify the authorisation of conditional role, access purpose and transaction.

(b1) A user's current conditional role has to be authorised.

For a system state $v = (OM, RM, PM, AM)$,

v satisfies privacy invariant-(b1), iff

$\forall\, u \in user(RM),\ \langle r, cond\rangle \in CR(RM)$:

$\langle r, cond\rangle = CCR(RM)(u) \Longrightarrow r \in AR(RM)(u) \wedge cond \Leftrightarrow$ true

(b2) A conditional role's access purpose in its current request has to be authorised for the conditional role.

For a system state v,

$v = (OM, RM, PM, AM)$,

v satisfies privacy invariant-(b2), iff

$\forall\, cr \in CR(RM),\ \langle obj, ap\rangle \in Req(AM)$:

$\langle obj, ap\rangle = CReq(AM)(cr) \Longrightarrow ap \in AAP(PM)(cr)$

(b3) A conditional role's current transaction has to be authorised for the conditional role's current request.

For a system state v,

$v = (OM, RM, PM, AM)$,

v satisfies privacy invariant-(b3), iff

$\forall\, cr \in CR(RM),\ trans \in Trans(AM)$:

$trans = CT(AM)(cr) \Longrightarrow trans \in AT(AM)(CReq(AM)(cr))$

These invariants specify the authorisation of conditional role, access purpose and transaction.

Next, we define invariants for data access and the deletion of data object.

(c) Data Access Constraints

(c1) A conditional role may only have current access to a data object if the access of executing the transaction on the object type is the necessary access for the access purpose.

For a system state v,

$v = (OM, RM, PM, AM)$,

v satisfies privacy invariant-(c1), iff

$\forall\, cr \in CR(RM),\ obj \in object(OM),\ ap \in$

$purpose(PM), x \in Mode(AM)$:

$\langle obj, ap\rangle = CReq(AM)(cr) \wedge \langle cr, obj, x\rangle \in CA(AM) \Rightarrow$

$\langle ap, Typeof(OM)(obj), CT(AM)(cr), x\rangle \in NA(AM)$

This invariant specifies the necessity of data access.

(c2) A conditional role may only have current access to a data object, if the purpose of its current request is compliant to the intended purposes of the type of the object.

For a system state v,

$v = (OM, RM, PM, AM)$,

v satisfies privacy invariant-(c2), iff

$\forall\, cr \in CR(RM),\ obj \in object(OM),\ ap \in$

$purpose(PM), x \in Mode(AM)$:

$\langle obj, ap\rangle = CReq(AM)(cr) \wedge \langle cr, obj, x\rangle \in CA(AM) \Rightarrow$

$ap \in IP(PM)(Typeof(OM)(obj))$

This specifies purpose compliance of data access.

(c3) A conditional role may delete a data object, if and only if it is necessary for its current request.

Given two successive system states $v, v^{'}$,

$v = (OM, RM, PM, AM)$,

$v^{'} = (OM^{'}, RM^{'}, PM^{'}, AM^{'})$,

$(v, v^{'})$ satisfies privacy invariant-(c3), iff

$\forall\, cr \in CR(RM),\ obj \in object(OM),\ ap \in$

$purpose(PM)$:

$obj \in object(OM) \wedge \langle obj, ap\rangle = CReq(AM)(cr)$

$\wedge \langle ap, Typeof(OM)(obj), CT(AM)(cr), delete\rangle \notin NA(AM)$

$\Longrightarrow obj \in Object(OM^{'})$

This specifies the necessity of data object deletion.

(c4) A conditional role may delete a data object, if and only if the purpose of its current request is compliant to the intended purpose of the type of the object.

Given two successive system states $v, v^{'}$,

$v = (OM, RM, PM, AM)$,

$v^{'} = (OM^{'}, RM^{'}, PM^{'}, AM^{'})$,

$(v, v^{'})$ satisfies privacy invariant-(c4), iff

$\forall\, cr \in CR(RM),\ obj \in object(OM),\ ap \in$

$purpose(PM)$:

$obj \in object(OM) \wedge \langle obj, ap\rangle = CReq(AM)(cr)$

$\wedge\, ap \notin IP(PM)(Typeof(OM)(obj))$

$\Longrightarrow obj \in object(OM^{'})$

This specifies purpose compliance of object deletion.

The invariant of the system state of PPS is the conjunction of these expressions, denoted as inv-PPS. A state v is a privacy-oriented state if and only if v satisfies the privacy invariants specified above.

The invariants have been specified in this section. Next we specify the model rules, and give the proof obligations of model rules.

4.5 Model Rules and Proof Obligations

Model operations are specified using pre-condition predicates, which are predicates on a single initial state, and two-state post-condition predicates, which are predicates over the initial and final state values. This type of specification aims to be implicit, which means it aims to fix the properties required of the approach without specifying how they are to be achieved. All operations must preserve any data type invariant that may exist. They may change the value of the state as long as the new value is valid state.

In this section, formal specifications of model rules are given. They specify operations by which the state variables can be changed. The precondition and the postcondition are used to specify the rules. Proof obligations [BFL$^+$94, Jon87] of operations show that the operations are satisfiable.

Rule1: create-object

Conditional role cr requests to create an object obj with the type tp.

This is specified as following:

$create\text{-}object(cr : CR(RM), obj, tp : type(OM))$

ext rd $RM : RoleModel$ rd $PM : PurposeModel$

rd $AM : AccessModel$ wr $OM : ObjectDataModel$

pre $obj \notin object(OM) \wedge \langle obj, ap\rangle = CReq(AM)(cr)$

$\wedge\, ap \in IP(PM)(tp) \wedge \langle ap, tp, CT(AM)(cr), create\rangle \in NA(AM)$

post $OM' = \langle object(OM) \cup \{obj\}, type(OM), typeAttr(OM), attrValue(OM),$

$TypeOf(OM) \cup \{obj \mapsto tp\}, AttrOf(OM), ValueOf(OM)\rangle$

Some notes on the operation specification above: (1) The first line of the operation specification is similar to that of a function. (2) The second part records those entities to which an operation has external (ext access. (3) Variable names are preceded by an indication of whether access is read only (rd) or read and write (wr). (4) The name of each variable is followed by its type. (5) The truth-valued pre-condition (pre) can refer only to the values of the parameters, while the post-condition (post) normally refers to the values of both parameters and result.

The pre-condition of the operation states that obj is not already in the set of objects, and to create object obj in current request is necessary access. The post-condition states that obj in included in the new object data model.

Next, defined symbols representing the operation's precondition and postcondition are introduced.

pre-create-object($cr, obj, tp, OM, RM, PM, AM$) $\stackrel{\text{def}}{=}$

$obj \notin object(OM) \wedge \langle obj, ap\rangle = CReq(AM)(cr) \wedge ap \in IP(PM)(tp)$

$\wedge \langle ap, tp, CT(AM)(cr), create\rangle \in NA(AM)$

post-create-object(obj, tp, OM, OM') $\stackrel{\text{def}}{=}$

$OM' = \langle object(OM) \cup \{obj\}, type(OM), typeAttr(OM), attrValue(OM),$

$TypeOf(OM) \cup \{obj \mapsto tp\}, AttrOf(OM), ValueOf(OM)\rangle$

The following is satisfiability obligation associated with this operation.

Proof Obligation 1: Operation $create\text{-}object$ is satisfiable.

$create\text{-}object\text{-}sat$

$OM : ObjectDataModel;\ RM : RoleModel;$

$PM : PurposeModel;\ AM : AccessModel;\ inv\text{-}PPS;$

pre-create-object($cr, obj, tp, OM, RM, PM, AM$)

$\exists\, obj, tp : type(OM), OM : ObjectDataModel, OM' : ObjectDataType \cdot$

post-create-object(obj, tp, OM, OM') $\wedge\ inv\text{-}PPS'$

Next we give proof for this satisfiability obligation.

from $OM : ObjectDataModel; RM : RoleModel;$

$PM : PurposeModel; AM : AccessModel; inv\text{-}PPS;$

pre-create-object($cr, obj, tp, OM, RM, PM, AM$)

1 $\{obj\} : object(OM')$-set

2 $object(OM) \cup \{obj\} : object(OM')$-set

3 $\{obj \mapsto tp\}$: $object(OM') \rightarrow type(OM')$

4 $TypeOf(OM) \cup \{obj \mapsto tp\}$: $object(OM') \rightarrow type(OM')$

5 from $inv\text{-}PPS$

5.1 $ap \in IP(PM)(tp)$

5.2 $\langle ap, tp, CT(AM)(cr), create\rangle \in NA(AM)$

infer $object(OM') = object(OM) \cup \{obj\} \wedge$

$TypeOf(OM') = TypeOf(OM) \cup \{obj \mapsto tp\}$

6 $\exists\, obj, tp : type(OM), OM : ObjectDataModel,$

$OM' : ObjectDataModel \cdot inv\text{-}PPS'$

infer $\exists obj, tp : type(OM), OM : ObjectDataModel,$
$OM' : ObjectDataType \cdot$
post-create-object$(obj, tp, OM, OM') \wedge inv\text{-}PPS'$

This proof obligation states that there must always be at least one state configuration satisfying the operation's post-condition whenever the system is in some legal state and when the operation's parameters satisfy its pre-condition in that state.

Rule 2: delete-object

Conditional role cr requests to delete an object obj.

This is specified as following:

$delete\text{-}object(cr : CR(RM), obj : object(OM))$

ext rd $RM : RoleModel$ rd $PM : PurposeModel$
wr $AM : AccessModel$ wr $OM : ObjectDataModel$

pre $\langle obj, ap\rangle = CReq(AM)(cr) \wedge ap \in IP(PM)(TypeOf(OM)(obj))$
$\wedge \langle ap, TypeOf(OM)(obj), CT(AM)(cr), delete\rangle \in NA$

post $OM' = \langle object(OM) \setminus \{obj\}, type(OM), typeAttr(OM), attrValue(OM),$
$TypeOf(OM), AttrOf(OM), ValueOf(OM)\rangle$
$\wedge AM' = \langle Req(AM), CReq(AM), Trans(AM), CT(AM), AT(AM),$
$Mode(AM), NA(AM), CA(AM) \setminus \{CA(AM) \cap \{CR \times \{obj\} \times Mode\}\}\rangle$

Then we give satisfiability obligation for this operation.

Proof Obligation 2: $delete\text{-}object$ is satisfiable.

$delete\text{-}object\text{-}sat$
$OM : ObjectDataModel;\ RM : RoleModel;$
$PM : PurposeModel;\ AM : AccessModel;\ inv\text{-}PPS;$
pre-delete-object$(cr, obj, OM, RM, PM, AM)$

$\exists\, obj : object(OM), OM : ObjectDataModel, OM' : ObjectDataModel,$
$AM : AccessModel, AM' : AccessModel \cdot$
post-delete-object$(obj, OM, OM', AM, AM') \wedge inv\text{-}PPS'$
with pre-delete-object$(cr, obj, OM, RM, PM, AM)$ and
post-delete-object(obj, OM, OM', AM, AM') defined as:
pre-delete-object$(cr, obj, OM, RM, PM, AM) \stackrel{def}{=}$
$\langle obj, ap\rangle = CReq(AM)(cr) \wedge ap \in IP(PM)(TypeOf(OM)(obj))$
$\wedge \langle ap, TypeOf(OM)(obj), CT(AM)(cr), delete\rangle \in NA$
post-delete-object$(obj, OM, OM' AM, AM') \stackrel{def}{=}$
$OM' = \langle object(OM) \setminus \{obj\}, type(OM), typeAttr(OM), attrValue(OM),$

$TypeOf(OM)$, $AttriOf(OM)$,$ValueOf(OM)\rangle$
$\wedge\ AM' = \langle Req(AM), CReq(AM), Trans(AM), CT(AM),\ AT(AM),$
$Mode(AM),\ NA(AM), CA(AM) \setminus \{CA(AM) \cap \{CR(RM) \times \{obj\}$
$\times Mode(AM)\}\}\rangle$

The proof of the satisfiability obligation is that of **Rule 1** with the substitution of privacy invariant (c3) and (c4) for privacy invariant (a1) and (a2).

Rule 3: get-access

Conditional role cr requests that access object obj in mode x be enabled, $x \in \{read, write, append\}$.

$get\text{-}access(cr : CR(RM), obj : object(OM), x : Mode)$

ext rd $RM : RoleModel$ rd $PM : PurposeModel$
wr $AM : AccessModel$ rd $OM : ObjectDataModel$

pre $\langle obj, ap\rangle = CReq(AM)(cr) \wedge \langle ap, Typeof(OM)(obj), CT(AM)(cr), x\rangle \in NA(AM) \wedge ap \in IP(Typeof(obj))$

post $AM' = \langle Req(AM), CReq(AM), Trans(AM), CT(AM), AT(AM), Mode(AM), NA(AM), CA(AM) \cup \{\langle cr, obj, x\rangle\}\rangle$

The proof of the satisfiability obligation is that of **Rule 1** with the substitution of privacy invariant (c1) and (c2) for privacy invariant (a1) and (a2).

Rule 4: release-access

Conditional role cr requests that access object obj in mode x be disabled, $x \in \{read, write, append\}$.

$release\text{-}access(cr : CR(RM), obj : object(OM), x : Mode)$

ext rd $RM : RoleModel$ rd $PM : PurposeModel$
wr $AM : AccessModel$ rd $OM : ObjectDataModel$

pre $\langle obj, ap\rangle = CReq(AM)(cr)$

post $AM' = \langle Req(AM), CReq(AM), Trans(AM), CT(AM), AT(AM), Mode(AM), NA(AM), CA(AM) \setminus \{\langle cr, obj, x\rangle\}\rangle$

The proof of the satisfiability obligation is that of **Rule 1** with the substitution of privacy invariant (c1) and (c2) for privacy invariant (a1) and (a2).

Rule 5: execute-transaction

Conditional role cr requests to execute transaction $trans$

$execute\text{-}transaction(cr : CR(RM), trans : Trans(AM))$

ext rd $RM : RoleModel$ wr $AM : AccessModel$

pre $trans \in AT(AM)(CReq(AM)(cr)) \wedge CT(AM)(cr) = Nil$

post $AM' = \langle Req(AM), CReq(AM), Trans(AM), \{trans\}, AT(AM), Mode(AM),$

$NA(AM), CA(AM)\rangle$

The proof of the satisfiability obligation is that of **Rule 1** with the substitution of privacy invariant (c1) and (c2) for privacy invariant (a1) and (a2).

Rule 6: exit-transaction

Conditional role cr requests to exit its current transaction $trans$.

$exit\text{-}transaction(cr : CR(RM), trans : Trans(AM))$

ext rd $RM : RoleModel$ wr $AM : AccessModel$

rd $OM : ObjectDataModel$

pre $\langle obj, ap\rangle = CReq(AM)(cr)$

post $AM' = \langle Req(AM), CReq(AM), Trans(AM), Nil, AT(AM), Mode(AM),$
$NA(AM), CA(AM) \setminus \{CA(AM) \cap \{\{cr\} \times object(OM) \times Mode(AM)\}\}\rangle$

The proof of the satisfiability obligation is that of **Rule 1** with the substitution of privacy invariant (c1) and (c2) for privacy invariant (a1) and (a2).

Rule 7: change-current-request

Conditional role cr requests that its current request be changed to req.

$change\text{-}current\text{-}request(cr : CR(RM), req : Req(AM))$

ext rd $RM : RoleModel$ rd $PM : PurposeModel$

wr $AM : AccessModel$ rd $OM : ObjectDataModel$

pre $\langle obj, ap\rangle = req \wedge ap \in AAP(PM)(cr) \wedge CT(AM)(cr) = Nil$

post $AM' = \langle Req(AM), req, Trans(AM), Nil, AT(AM), Mode(AM),$
$NA(AM), CA(AM)\rangle$

The proof of the satisfiability obligation is that of **Rule 1** with the substitution of privacy invariant (c1) and (c2) for privacy invariant (a1) and (a2).

The following **Rules 8-19** are management operations, and they are performed by privacy officer. Since they don't create, delete, or access objects, they won't change the satisfiability of invariants.

Rule 8: add-NA

Conditional role cr requests to add the tuple $\langle ap_i, type_j, trans_k, x\rangle$ to *NA*.

$add\text{-}NA(cr : CR(RM), \langle ap_i, type_j, trans_k, x\rangle)$

ext rd $RM : RoleModel$ wr $AM : AccessModel$

pre $cr = privacy\text{-}officer \wedge \langle ap_i, type_j, trans_k, x\rangle \notin NA(AM)$

post $AM' = \langle Req(AM), CReq(AM), Trans(AM), CT(AM), AT(AM), Mode(AM),$
$NA(AM) \cup \{\langle ap_i, type_j, trans_k, x\rangle\}, CA(AM)\rangle$

Rule 9: delete-NA

Conditional role cr requests to delete the tuple $(ap_i, type_j, trans_k, x)$ from *NA*.

$delete\text{-}NA(cr : CR(RM), \langle ap_i, type_j, trans_k, x\rangle : NA(AM))$

ext rd $RM : RoleModel$ rd $PM : PurposeModel$

wr $AM : AccessModel$ rd $OM : ObjectDataModel$

pre $cr = privacy\text{-}officer \wedge (\forall\, cr_i \in CR(RM), a \in Mode, ap \in purpose(PM),$
$obj \in object(OM)$: $\langle obj, ap\rangle = CReq(AM)(cr_i) \wedge \langle cr_i, obj, a\rangle \in CA(AM) \Rightarrow$
$\langle ap, TypeOf(OM)(obj), CT(AM)(cr_i), a\rangle \neq \langle ap_i, type_j, trans_k, x\rangle)$

post $AM' = \langle Req(AM), CReq(AM), Trans(AM), CT(AM), AT(AM), Mode(AM),$
$NA(AM) \setminus \{\langle ap_i, type_j, trans_k, x\rangle\}, CA(AM)\rangle$

Rule 10: add-request

Conditional role cr requests to add the tuple $\langle obj, ap\rangle$ to ***Req***.

$add\text{-}request(cr : CR(RM), \langle obj, ap\rangle)$

ext rd $RM : RoleModel$ rd $PM : PurposeModel$

wr $AM : AccessModel$

pre $cr = privacy\text{-}officer \wedge \langle obj, ap\rangle \notin Req(AM)$

post $AM' = \langle Req(AM) \cup \{\langle obj, ap\rangle\}, CReq(AM), Trans(AM), CT(AM),$
$AT(AM) \cup \{\langle obj, ap\rangle \mapsto Nil\}, Mode(AM), NA(AM), CA(AM)\rangle$

Rule 11: delete-request

Conditional role cr requests to delete the tuple $\langle obj, ap\rangle$ from ***Req***.

$delete\text{-}request(cr : CR(RM), \langle obj, ap\rangle : Req(AM))$

ext rd $RM : RoleModel$ rd $PM : PurposeModel$

wr $AM : AccessModel$ rd $OM : ObjectDataModel$

pre $cr = privacy\text{-}officer \wedge (\forall\, cr_i \in CR(RM) : CReq(AM)(cr_i) \neq \langle obj, ap\rangle)$

post $AM' = \langle Req(AM) \setminus \{\langle obj, ap\rangle\}, CReq(AM), Trans(AM), CT(AM),$
$AT(AM), Mode(AM), NA(AM) \setminus \{NA(AM) \cap \{\{ap\} \times \{TypeOf(OM)(obj)\} \times$
$Trans(AM) \times Mode\}\}, CA(AM)\rangle$

Rule 12: add-type

Conditional role cr requests to define an object type tp.

$add\text{-}type(cr : CR(RM), tp)$

ext rd $RM : RoleModel$ rd $PM : PurposeModel$

rd $AM : AccessModel$ wr $OM : ObjectDataModel$

pre $cr = privacy\text{-}officer \wedge tp \notin type(OM)$

post $OM' = \langle object(OM), type(OM) \cup \{tp\}, typeAttribute(OM),$
$attributeValue(OM), TypeOf(OM), AttributeOf(OM), ValueOf(OM)\rangle$

Rule 13: delete-type

Conditional role cr requests to delete tp from $type(OM)$.

$delete\text{-}type(cr : CR(RM), tp : type(OM))$

ext rd $RM : RoleModel$ rd $PM : PurposeModel$

wr $AM : AccessModel$ wr $OM : ObjectDataModel$

pre $cr = privacy\text{-}officer \wedge (\forall\, obj \in object(OM) : TypeOf(obj) \neq tp)$

post $OM' = \langle object(OM), type(OM) \setminus \{tp\}, typeAttribute(OM), attributeValue(OM), TypeOf(OM), AttributeOf(OM), ValueOf(OM)\rangle \wedge AM' = \langle Req(AM), CReq(AM), Trans(AM), CT(AM), AT(AM), Mode(AM), NA(AM) \setminus \{NA(AM) \cap \{purpose(PM) \times \{TypeOf(OM)(obj)\} \times Trans(AM) \times Mode\}\}, CA(AM)\rangle$

Rule 14: add-purpose

Conditional role cr requests to add the purpose p to $purpose(PM)$.

$add\text{-}purpose(cr : CR(RM), p)$

ext rd $RM : RoleModel$ wr $PM : PurposeModel$

pre $cr = privacy\text{-}officer \wedge p \notin purpose(PM)$

post $PM' = \langle purpose(PM) \cup \{p\}, Specialisation(PM), Generalisation(PM), IP(PM), AAP(PM)\rangle$

Rule 15: delete-purpose

Conditional role cr requests to delete the purpose p from $purpose(PM)$.

$delete\text{-}purpose(cr : CR(RM), p : purpose(PM))$

ext rd $RM : RoleModel$ wr $PM : PurposeModel$

rd $AM : AccessModel$ rd $OM : ObjectDataModel$

pre $cr = privacy\text{-}officer \wedge (\forall\, obj \in object(OM) : p \notin IP(PM)(TypeOf(OM)(obj))) \wedge (\forall\, tp \in type(OM) : p \notin IP(PM)(tp)) \wedge (\forall\, \langle obj, ap\rangle \in Req(OM) : ap \neq p)$

post $PM' = \langle purpose(PM) \setminus \{p\}, Specialisation(PM), Generalisation(PM), IP(PM), AAP(PM)\rangle$

Rule 16: add-authorised-transaction

Conditional role cr requests to authorise transaction $trans_j$ to request req_i.

$add\text{-}authorised\text{-}transaction(cr : CR(RM), req_i : Req(AM), trans_j : Trans(AM))$

ext rd $RM : RoleModel$ wr $AM : AccessModel$

pre $cr = privacy\text{-}officer \wedge trans_j \notin AT(AM)(req_i)$

post $AM' = \langle Req(AM), CReq(AM), Trans(AM), CT(AM), AT(AM) \cup \{req_i \mapsto trans_j\}, Mode(AM), NA(AM), CA(AM)\rangle$

Rule 17: delete-authorised-transaction

Conditional role cr requests to revoke transaction $trans_j$ from request req_i.

$delete\text{-}authorised\text{-}transaction(cr : CR(RM), req_i : Req(AM), trans_j : Trans(AM))$

ext rd $RM : RoleModel$ wr $AM : AccessModel$

pre $cr = privacy\text{-}officer \wedge (\forall\, cr_i : CT(AM)(cr_i) \neq trans_j \vee CReq(AM) \neq req_i)$

post $AM' = \langle Req(AM), CReq(AM), Trans(AM), CT(AM), AT(AM) \backslash \{req_i \mapsto trans_j\}, Mode(AM), NA(AM), CA(AM)\rangle$

Rule 18: create-transaction

Conditional role cr requests to add a new transaction $trans_j$.

$create\text{-}transaction(cr : CR(RM), trans_j)$

ext rd $RM : RoleModel$ wr $AM : AccessModel$

pre $cr = privacy\text{-}officer \wedge trans_j \notin Trans(AM)$

post $AM' = \langle Req(AM), CReq(AM), Trans(AM) \cup \{trans_j\}, CT(AM), AT(AM), Mode(AM), NA(AM), CA(AM)\rangle$

Rule 19: delete-transaction

Conditional role cr requests to delete a transaction $trans_j$ from $Trans(AM)$.

$delete\text{-}transaction(cr : CR(RM), trans_j : Trans(AM))$

ext rd $RM : RoleModel$ wr $AM : AccessModel$

pre $cr = privacy\text{-}officer \wedge (\forall\, cr_i : CT(AM)(cr_i) \neq trans_j)$

post $AM' = \langle Req(AM), CReq(AM), Trans(AM) \backslash \{trans_j\}, CT(AM), AT(AM), Mode(AM), NA(AM), CA(AM)\rangle$

We have defined a state v as a privacy-oriented state if it satisfies the privacy invariants $inv\text{-}PPS$. From the specification and satisfiability obligation proofs of the model rules, for the system PPS starting from initial state σ, since σ satisfies $inv\text{-}PPS$, and model Rules 1-19 maintain the invariants $inv\text{-}PPS$, we can conclude that PPS is a privacy-oriented system.

4.6 Access Control Mechanism

The authorisation process in the purpose based access control method is illustrated in Figure 4.1. When a data user requests to access certain data object, access control is to be checked first and corresponding data policy, including data owner's preferences, are retrieved from privacy management system. Access control is checked based on conditional roles activated, the subjects invoked by those roles to access data, and role-subject mapping. If the request passes the role check, then data purposes is to be checked against the access purpose. If access purpose is compliant with the intended

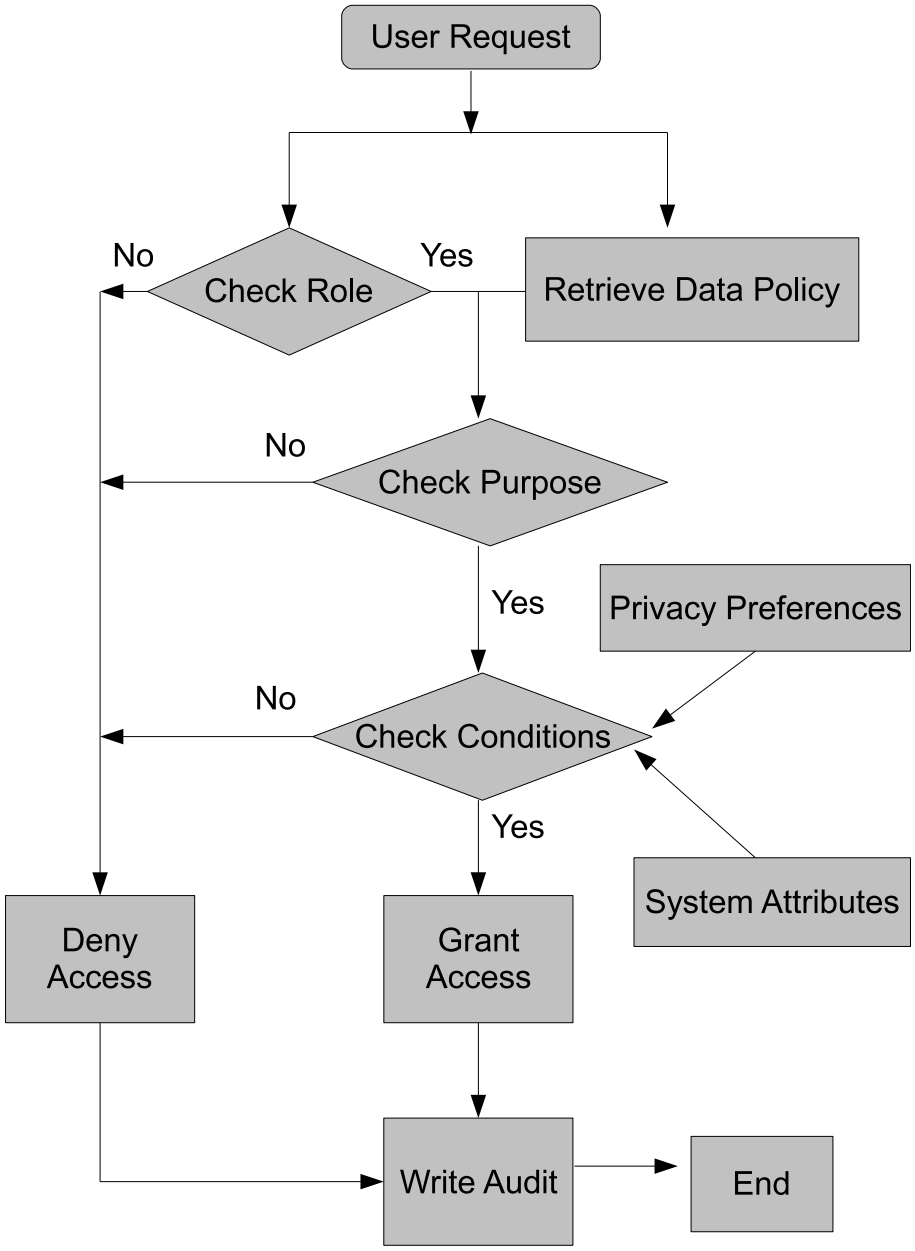

Figure 4.1: Authorisation Process of Purpose Based Access Control

purpose, then conditions are to be checked. If the request passes this procedure, then access is granted. Otherwise, access is denied. All data access requests should be logged in the audit trail.

4.7 Analysis of the Purpose Based Access Control Method

The purpose based access control approach was designed to enforce legal privacy requirements, such as necessity of data processing and purpose binding. In this section, we discuss issues related to the proposed method.

4.7.1 Purpose in Access Request

There is a close relation between the notion of purpose in privacy policies and the notion of role in RBAC. A role in RBAC can be defined as a set of actions and responsibilities associated with a particular activity [SS94]. Purpose in privacy policies is defined as a reason for data collection and use [CLM+02b], and business purposes can be identified through task definition within an organisation's IT systems and applications [PAS02]. Here, a close relation can be observed between what responsibility a user has (can be modeled as role) and its associated tasks for fulfilling the responsibilities (can be modelled as purpose). On the other hand, fulfilling the tasks requires access to data, which is represented by permissions in RBAC. The similarities between them tend to make it difficult to make distinction in some situations. For example, order processing in a store can be modelled as a specific role compared to widely scoped roles (as described in [SdV01]), as well as a purpose for accessing customer record (as described in [KS02]). Roles are usually derived based on organisational positions and responsibilities. But purposes have no relation to organisational structure, but to functions.

Some may argue that the role entity itself can support both notions of role and purpose, by having structural and functional roles respectively. In such an approach, the model needs to deal differently with those two role types by allowing:

- assignment of permissions only to functional roles,
- inherence of functional roles by structural roles,
- assignment of users only to structural roles.

Moreover, comparing to RBAC, the authorisation mechanism use assertion of purpose as a part of access request. That feature is not supported in standard RBAC, as access check is only based on session and requested permission [FK92]. Therefore, the access checking function in the standard RBAC needs to be changed to be aware of the access purpose.

4.7.2 Purpose Management

Access control methods that support privacy protection usually require a user to indicate the purpose of accessing information as one of the access request parameters. The indicated purpose is then used to check for compliance with policies in the system and preferences set by data owners.

The drawback in existing approaches is that users can indicate any purpose for information access without any restriction. Although the indicated purpose is checked against the policy, but that freedom makes system very vulnerable to misuse of data for purposes not really related to a role. That can happen with the existence of a simple error in the policy rules, which is not unlikely in practice considering the presence of role and purpose hierarchies. This problem can be solved in that the user cannot use data for a purpose without first having been authorised for that purpose. Such authorisation is possible only for those purposes assigned to the user's currently active roles. These assignments come from the fact that any role has a restricted set of responsibilities and functionalities, which will define purposes for privacy-sensitive information access.

4.7.3 Expressibility

The method decreases the expression complexity by avoiding rules involving multiple entities, namely roles, purpose, and permission. The core part of privacy policies usually state purposes for and circumstances under which collected data would be used, and the extent of use of personal information may differ based on the purpose of use. For example, health record of a job applicant may be used for the purpose of approval of qualification for a special job requiring certain degree of health, without disclosing details. However, the details may be used for the purpose of treatment of that person as a patient. Expressibility challenge may arise in a scenario where different roles with the same purpose can have different accesses.

Consider the following scenario in a healthcare scenario. The role *senior researcher* can access complete profiles of specific patients with the purpose *research*,

with previous consent. However, the role *research assistant* with the same purpose should only have access to limited profiles. But if purposes are defined fine-grained enough in the system, there would not be an expression problem most of the time. For the aforementioned scenario, although both accesses by roles senior researcher and research assistant have the purpose *research* at high level, they can be categorised into the more fine grained purposes *complete research* and *limited research*, respectively. If purposes are not easy to define in a scenario, the method can cope with the issue by allowing predicates based on conditional role, such as the specialty of roles. Therefore, permissions can be assigned with purposes based on the user's active conditional roles.

4.7.4 Sticky Policies

The sticky policy paradigm [KSW03] is a flexible approach for privacy policies. In that approach, policies are defined and maintained for each data item. Therefore, privacy policies can be different for different instances of the same data type. We argue that it is less probable to be followed by organisations. The main drawback is that the organisations would loose its centralised control over access control policies once the policy is stuck to the data. That is not preferred since the access control policy may require changes due to revision of high-level policies, or frequent improvement of the access control policy itself. Moreover, the storage and processing of access control policies will be very expensive in the case of using sticky policy approach.

4.8 Chapter Summary

This chapter described our basic privacy preserving access control method. It uses purpose to combine authorisation and data accesses. The framework presented in this approach captures the essential features such as referenced by Byun et al [BBL05, BL08] and Fischer-Hubner et al [FHO98].

The entities of the model were formally specified, including a data object model, users and roles, purposes, and request, transactions, and accesses. With these entities it is possible to present invariants corresponding to privacy requirements. These invariants are defined through the data processing stages, which consists of data collection, role authorisation, and data accesses. In order to complete the access control model, specifications of operations and their proof obligations have also been investigated. We have shown how this is achieved using VDM.

Chapter 5

A Case Study of Privacy Protection Specification

5.1 Chapter Introduction

The specification of a privacy preserving approach in a medical care scenario is presented in this chapter. The purpose-based privacy model proposed in Chapter 4 is used to specify the privacy protection approach. Real world medical care systems are generally large and overly complex. Designing privacy-preserving approaches for such systems is a challenging task. In this chapter, we present a privacy-preserving approach for personal information management system in an imaginary hospital scenario. The specification covers the privacy protection for patients throughout the medical record handling process, while complying with most aspects of healthcare practice. The basic problem analysis is provided to the extent that the scenario is simplified and part of personal data processing in a hospital environment is specified. Nonetheless the material presented here should provide a good indication of how formal specification techniques could be integrated into traditional requirements analysis approach.

This chapter is organised as follows: Section 5.2 describes the informal requirements for the data handling process; Section 5.3 presents an analysis of the requirements using entity relationship modeling and the various operations are also identified; from this an outline specification structure is derived in Section 5.4, and the analysis of the specification against privacy requirements is presented; in Section 5.5, a comparison with two other specification case studies is presented; and concluding remarks are included in Section 5.6.

5.2 Data Handling Process

In this medical care scenario, we consider a patient's medical record handling for the medical treatment in a hospital. Patient's personal data is processed following the subsequent steps:

- Patient admission
- Diagnosis and treatment suggestions by an examination specialist
- Treatment by a therapy or transfer to another medical treatment centre or care ward
- Patient discharge
- Transfer of billing information to the patient's health insurance company

From the system's point of view, during and after the patient's stay in hospital, with the consideration of privacy requirements, the patient's data is processed by the following steps:

- An admission staff member (with the access purpose "admission") creates a file for the patient's admission data with *create*, collects administrative, social, demographic and insurance information about the patient and writes it into the patient's admission data file by using an editor. He creates a treatment history file with the operation *create* and appends the action "admission" to the treatment history.
- The examination specialist (with the access purpose "diagnosing") creates a diagnosis file and uses the editor to write and change his diagnosis. Then he creates a treatment suggestion file for this patient and writes into it with an editor, changing it when necessary. Finally, he appends his actions to the treatment history file. If necessary, he can transfer a patient to another specialist, medical treatment centre or care ward (with the access purpose "treatment transfer" or "care transfer"). For a patient transfer he can transfer diagnosis and treatment suggestions data by using the transfer programs.
- The therapist (with the access purpose "therapy") reads the treatment suggestions, treats the patient and appends his treatment actions to the treatment history.

- When the treatment has been completed, the patient is discharged by an admission staff member, who makes last entries to the patient's admission data and treatment history data by using the append editor.
- At last, a billing staff member (with the access purpose "billing") reads the treatment history and creates and edits the billing data file, which he transfers to the patient's medical insurance company by using the data transfer program.
- With the purpose "statistical analysis", diagnoses and treatment histories data can be read by the researcher's statistic program in order to calculate statistical data. However, access to these files for research purposes is only permitted if the patients in question have given their consents. Statistical data files can only be created, changed and deleted by users performing the current transaction statistical analysis.

5.3 Requirements Analysis and Entity Identification

In this section, we undertake an analysis on the requirements in order to get a feel for the structure of the problem and to extract entities. Requirements analysis provides a framework for examining the given requirements and extracting entities and the relationships that exist among those entities. Having identified entities, their attributes and relationships, the operations required are tabulated. When producing a VDM specification, we are required to design a state model which captures the essential entities and entity relationships. Operations are then specified using the state model.

5.3.1 Privacy Requirements

In the imaginary hospital scenario, the privacy model can protect personal patient data by enforcing the privacy principals of *necessity of data processing* and *purpose binding*. In particular, the personal data processing of patient-related data in the areas of medical treatment, administration and care should be separated as far as possible. For example, administration personnel should not have access to medical data and doctors (physicians) should not have access to billing data. Diagnosis data and treatment suggestions may be transferred to another medical treatment centre or care ward. Researchers may use patient data for statistical (research) purposes, if the patients have given their consent. Besides, billing information is transferred to the patient's insurance companies.

Admin	**MedTreat**	**Care**	**Research**
Registration	Diagnosing	Care Transfer	Statistical Analysis
Admission	Prescribing		
Billing	Therapy		
Data transfer	Treatment Transfer		

Table 5.1: Purposes in a Medical Care Scenario

5.3.2 Entity Identification

The organisation of a hospital is divided into the areas of medical treatment, care, administration and research. $Object$ denotes the set of data objects in the system, which are then categorised into object types of *registration data*, *admission data*, *treatment history*, *diagnosis*, *prescription*, *treatment suggestions*, *billing data*, and *statistical data*. Object types are denoted as $Object$-set. Attributes associated with object types and their values are denoted as $TypeAttr$ and $AttrValue$ respectively.

Appropriate purposes for separating the main areas in a hospital are: *administration* (admin), *medical treatment* (medTreat), *care* (care) and *research* (research). These main purposes can be further divided into sub-purposes, as shown in Table 5.1. The set of purposes is denoted as $Purpose$. Table 5.2 illustrates intended purpose for data objects in the medical care scenario.

Users in the hospital, denoted as $User$, are assigned roles which are denoted as $Role$. Roles in the system include *registration staff*, *examination specialist*, *therapist*, *billing staff*, and *researcher*. Roles are authorised for certain access purposes. Table 5.3 illustrates the roles and their authorised access purposes in a medical care scenario.

Personal data objects are processed by the transactions as in Table 5.4, which is denoted as $Transaction$. Then the next step is the definition of authorised transactions for access purposes, as in Table 5.5. Necessary accesses (denoted as NA) are then defined as in Table 5.6. Possible access modes are *create*, *read*, *write*, *append*, and *delete*.

Transctions describe changes of state variables that may take place. They are divided into general transactions, which execute objects accesses, and privileged transactions, which administrate and define access control information. Privileged transactions must be managed by a *privacy officer*.

Object Types	**Intended Purposes**	**Content**
Registration Data	registration	Demographic information about a patient
Admission Data	admission	Administrative, social, and insurance information about a patient
Treatment History	admission, medTreat	Treatment history
Diagnosis	diagnosing	Diagnosis data
Prescription	prescribing, billing	Prescription data
Treatment Suggestions	diagnosing, therapy	Instructions for surgeons and therapists
Billing Data	billing	Billing information about a patient and his medical treatment
Statistics	statistical analysis	Statistical data

Table 5.2: Object Types and Intended Purposes in a Medical Care Scenario

Roles	**Access Purposes**
Registration staff	registration, admission
Examination specialist	diagnosing, therapy, prescribing, treatment transfer
Therapist	therapy, treatment transfer, care transfer
Billing staff	billing, data transfer
Researcher	statistical analysis

Table 5.3: Roles and Access Purposes in a Medical Care Scenario

Transaction	**Usage**
Create	Creation of personal data file of a specified type
Append	Appending text to an existing file
Edit	Reading and writing a text file
Display	Reading a text file and displaying it on the screen
Delete	Deletion of a file
Transfer	Data transfer by interprocess communication
Statistic	Reading files, calculating and writing statistics

Table 5.4: Transactions in a Medical Care Scenario

5.4 System Specification

In Section 5.3 we examined the requirements using the entity relationship analysis approach. This yielded a number of entities and associated attributes as well as a list of operations. That analysis provides a framework for producing a formal specification of the information system. We specify the set of entities and identify entity models. The entity models are derived from the analysis on the entities and the entity relationships already undertaken in Section 5.3. We will also list the operations identified in previous section.

5.4.1 Specification of Entities

With the reference to Section 4.3, the entity models are specified as follows:

$$\begin{aligned} ObjectDataModel :: object &: Object\text{-set} \\ type &: (Object\text{-set})\text{-set} \\ typeAttr &: TypeAttr\text{-set} \\ attrValue &: AttrValue\text{-set} \\ TypeOf &: Object \rightarrow Object\text{-set} \\ AttrOf &: Object\text{-set} \rightarrow TypeAttr\text{-set} \\ ValueOf &: Object \times TypeAttr \rightarrow AttrValue \end{aligned}$$

The object data model *OM* in our system has the type $ObjectDataModel$

$$\begin{aligned} RoleModel :: role &: Role\text{-set} \\ user &: User\text{-set} \\ UserRole &: User \leftrightarrow Role \end{aligned}$$

Access Purpose	**Authorised Transactions**
registration	Create, Edit
diagnosing	Create, Append, Edit, Display
therapy	Display, append
treatment transfer	Transfer
prescribing	Append, Display
care transfer	Transfer
admission	Create, Edit
billing	Edit, Display
data transfer	Transfer
statistical analysis	Create, Edit, Statistic

Table 5.5: Authorised Transactions for Access Purposes in a Medical Care Scenario

$$AR : User \rightarrow Role\text{-set}$$
$$roleAttr : RoleAttr\text{-set}$$
$$roleAttrValue : RoleAttrValue\text{-set}$$
$$RoleAttrValueOf : Role \times RoleAttr \rightarrow RoleAttrValue$$
$$sysAttr : SysAttr\text{-set}$$
$$sysAttrValue : SysAttrValue\text{-set}$$
$$SysAttrValueOf : SysAttr \rightarrow SysAttrValue$$
$$CR : CondRole\text{-set}$$
$$CCR : User \rightarrow CR$$

The role model *RM* in our system has the type $RoleModel$.

$$PurposeModel :: purpose : Purpose\text{-set}$$
$$Specialisation : Purpose \times Purpose \rightarrow \mathbb{B}$$
$$Generalisation : Purpose \times Purpose \rightarrow \mathbb{B}$$
$$IP : object(OM) \cup type(OM) \rightarrow Purpose\text{-set}$$
$$AAP : CR(RM) \rightarrow Purpose\text{-set}$$

The purpose model *PM* in our system has the type $PurposeModel$.

$$AccessModel :: Req : Request\text{-set}$$
$$CReq : CR(RM) \rightarrow Req$$
$$Trans : Transation\text{-set}$$

Access Purpose	Object Type	Transactions	Accesses
admission	Admission data	Create	create
″	″	Edit	read, write, append
″	Treatment history	Create	create
″	″	Append	append
diagnosing	Diagnosis	Create	create
″	″	Edit	read, write, append
″	″	Display	read
″	Treatment history	Append	append
″	Treatment suggestions	Create	create
″	″	Edit	read, write, append
therapy	Treatment history	Display	read
″	Treatment suggestions	Append	append
treatment transfer	Diagnosis	Transfer	read, write, append
″	Treatment suggestions	Transfer	read, write, append
prescribing	Precription	Create	create
″	″	Append	append
care transfer	Diagnosis	Transfer	read, write, append
″	Treatment suggestions	Transfer	read, write, append
billing	Treatment history	Display	read
″	Billing data	Create	create
″	″	Edit	read, write, append
data transfer	Billing data	Transfer	read, write, append
statistics	Statistics	Create	create
″	″	Edit	read, write, append
″	″	Delete	delete
″	Diagnosis	Statistic	read
″	Treatment history	Statistic	read

Table 5.6: Necessary Accesses in a Medical Care Scenario

$$CT : CR(RM) \rightarrow Trans$$
$$AT : Req \rightarrow Trans\text{-set}$$
$$Mode: AccMode\text{-set}$$
$$NA : NecAcc\text{-set}$$
$$CA : CurAcc\text{-set}$$

The access model *AM* in our system has the type $AccessModel$.

5.4.2 Medical Care System and Operations

The medical care system state can be specified as:

state MCS of
$OM : ObjectDataModel$
$RM : RoleModel$
$PM : PurposeModel$
$AM : AccessModel$
inv ...
init ...
end

where the initialisation condition on the state is:

init $\sigma \triangleq \{\sigma.object(OM) = \{\} \wedge \sigma.type(OM) = \{\} \wedge$
$\sigma.purpose(PM) = \{\} \wedge \sigma.AAP(PM) = \{\mapsto\} \wedge$
$\sigma.Req(AM) = \{\} \wedge \sigma.Trans(AM) = \{\} \wedge$
$\sigma.CA(AM) = \{\} \wedge \sigma.NA(AM) = \{\}\}$

General operations regarding medical record include: $create\text{-}admission\text{-}data$, $edit\text{-}admission\text{-}data$, $create\text{-}treatment\text{-}history$, $append\text{-}treatment\text{-}history$, $create\text{-}diagnosis$, $edit\text{-}diagnosis$, $display\text{-}diagnosis$, $create\text{-}treatment\text{-}suggestion$, $edit\text{-}treatment\text{-}suggestion$, $transfer\text{-}diagnosis$, $transfer\text{-}treatment\text{-}suggestions$, $create\text{-}billing\text{-}data$, $edit\text{-}billing\text{-}data$, $create\text{-}statistics$, $edit\text{-}statistics$, $delete\text{-}statistics$, $execute\text{-}transaction$, $exit\text{-}transaction$, and $change\text{-}current\text{-}request$.

Privileged operations include: $add\text{-}NA$, $delete\text{-}NA$, $add\text{-}request$, $delete\text{-}request$, $add\text{-}type$, $delete\text{-}type$, $add\text{-}purpose$, $delete\text{-}purpose$, $create\text{-}transaction$, $delete\text{-}transaction$, $add\text{-}authorised\text{-}transaction$, and $delete\text{-}authorised\text{-}transaction$.

The operations listed above are instances of model rules specified in the purpose based access control model in Chapter 4. For example, $create\text{-}admission\text{-}data$ is a particular instance of Rule 1 $create\text{-}object$ for object type $admission\text{-}data$. It is then specified as:

$create\text{-}admission\text{-}data(cr : CR(RM), obj, admission\text{-}data: type(OM))$

ext rd $RM : RoleModel$ rd $PM : PurposeModel$

rd $AM : AccessModel$ wr $OM : ObjectDataModel$

pre $obj \notin object(OM) \wedge \langle obj, ap\rangle = CReq(AM)(cr)$

$\wedge\ ap \in IP(PM)(admission\text{-}data)$

$\wedge\ \langle ap, admission\text{-}data, CT(AM)(cr), create\rangle \in NA(AM)$

post $OM' = \langle object(OM) \cup \{obj\}, type(OM), typeAttr(OM), attrValue(OM),$

$TypeOf(OM) \cup \{obj \mapsto admission\text{-}data\}, AttrOf(OM), ValueOf(OM)\rangle$

The determined operations as instantiations of more generic model rules specified in Chapter 4 are listed in Table 5.7. Note that operations in the third row are about data objects access, and they consist of instantiations of both $get\text{-}access$ and $release\text{-}access$.

A state v is a privacy-oriented state if it satisfies the privacy invariants $inv\text{-}PPS$ in Section 4.4. From the specification and analysis on the relationships between operations of MCS and the model rules, for the system MCS starting from initial state σ, since σ satisfies $inv\text{-}PPS$, and model Rules 1-19 maintain the invariants $inv\text{-}PPS$, we can conclude that MCS is a privacy-oriented system.

5.5 Comparisons

In Chapter 3, we have analysed some related work on privacy protection specification based on access control. But there are not many researches provided case studies to illustrate the application of specifications. In this section we compare our case study with two other case studies provided by related work - PARBAC [He03] and task-based access control [FHO98] approaches.

PARBAC and task-based access control approaches both provided case studies in healthcare sector. The case study of PARBAC illustrated a simplified scenario in an online drug store. It briefly described the data collected by the store and the privacy policy of the store, and then listed model entities, including users, roles, domains, subjects, tasks, purposes, and entity relationship mappings among entities. Based on the authorisation process, it then examined two example requests from data users and analysed whether these requests will be granted or denied access. The case study of task based access control method described privacy protection in a hospital. It listed some entities in the scenario, including purposes, possible tasks, object classes of personal patient data, transformation procedures, and necessary accesses.

Operations	**Model Rules**
create-admission-data, *create-treatment-history*, *create-treatment-suggestion*, *create-billing-data*, *create-diagnosis*, *create-statistics*	*create-object*
delete-statistics	*delete-object*
edit-diagnosis, *display-diagnosis*, *edit-treatment-suggestion*, *transfer-diagnosis*, *transfer-treatment-suggestions*, *edit-billing-data*, *edit-statistics*	*get-access*, *release-access*
execute-transaction	*execute-transaction*
exit-transaction	*exit-transaction*
change-current-request	*change-current-request*
add-NA	*add-NA*
delete-NA	*delete-NA*
add-request	*add-request*
delete-request	*delete-request*
add-type	*add-type*
delete-type	*delete-type*
add-purpose	*add-purpose*
delete-purpose	*delete-purpose*
add-authorised-transaction	*add-authorised-transaction*
delete-authorised-transaction	*delete-authorised-transaction*
create-transaction	*create-transaction*
delete-transaction	*delete-transaction*

Table 5.7: Operations as instantiations of Model Rules

Our work goes beyond these two case studies by specifying the privacy protection for patients throughout the medical record handling process while complying with most aspects of healthcare practice. The two case studies given by PARBAC and task based method are both very elementary. For example, PARBAC just lists some model entities and two example requests, while task-based method only gives a simple description of possible entities in a hospital scenario. They are descriptions rather than illustrations of the application of specifications. Our case study covers the informal requirements for the data handling process, and by analysing the requirements, identifies the entities and their relationships, and the various operations. The analysis on the specification and operations against privacy requirements is also presented in our work. This analysis links the exemplar scenario to the generic specification of purpose-based access control method, thus makes the case study a good illustration of the application of purpose-based access control method in medical care scenario.

5.6 Chapter Summary

This chapter presented a case study of formal specification in the development of a privacy preserving system in a medical care scenario. In the simplified hospital scenario, the basic problem analysis was provided and personal data handling process was specified. The entities were identified and specified, and the specification and operations were analysed against privacy requirements. Comparison was made against case studies in two other privacy protection specification work. The case study provided a good indication of how formal specification techniques could be integrated into traditional requirements analysis approach, and showed how an initial specification can be formed and then manipulated in a rigorous way through the careful introduction of design detail in the form of data structure and operations.

Chapter 6

Privacy in Distributed Collaborative Computing

6.1 Chapter Introduction

Previously in Chapter 4, we have presented a purpose based access control model. It consists of the following entities: *data objects*, *users and roles*, *purposes*, and *data accesses*. The privacy policy defines a set of rules. Each rule regulates a set of data users about accessing data objects for certain purposes. These rules are analysed and specified into invariants of the model. The model lays out a mechanism to protect data owner's privacy within an organisation.

We analyse the design considerations for privacy preservation in distributed collaborative environments in this chapter. Distributed computing environments pose further challenges. The first consideration is to overcome limitations of object type in the purpose based access control approach. The key problem addressed for this part is how to ensure that personal information is accessed from two or more parties only if agreed privacy policies and privacy preferences are satisfied. The second consideration is to provide mechanisms for facilitating privacy policies matching and privacy preference compliance among distributed collaborative organisations.

The chapter is organised as follows: Section 6.2 provides a motivating scenario for privacy protection in distributed computing environment; Section 6.3 analyses model entities according to the structure of the model; Section 6.4 presents notions relevant to the discussion; Section 6.5 discusses related work; Section 6.6 specifies the purpose based access control model for distributed collaborative organisations; and Section 6.7 presents privacy policies matching in a federation.

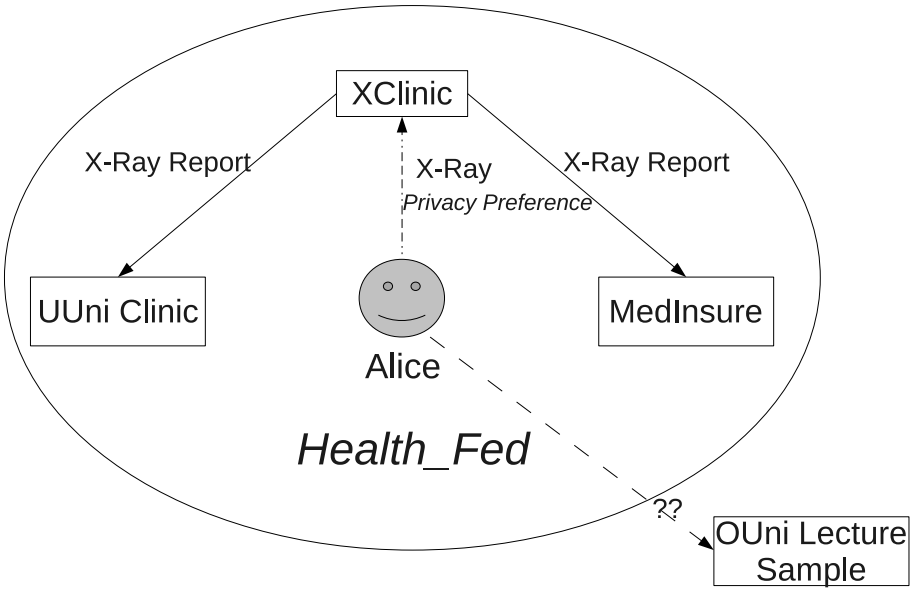

Figure 6.1: A Motivating Healthcare Scenario

6.2 Motivating Scenario

Health care providers and payers conduct many transactions electronically. They maintain large volumes of confidential health information along with other sensitive personal and financial data. In this arena an individual's personal information includes his/her medical records, which is made up of (often disjoint) treatment records from different health institutions. Privacy of medical records and medical-related personal information requires particular attention. To analyse the specific challenges and requirements of this environment, we consider the example of an online federation of hospitals and organisations collaborating with each other, named *Health_Fed*. We assume each federated organisation being composed by entity collecting data owners' information and interacting with data owners and other federated organisations through negotiations. In particular, we refer to the scenario of a data owner Alice who is a student of University UUni, as shown in Fig 6.1.

We start from Alice getting an X-Ray performed at a clinic XClinic, part of the *Health_Fed* federation. The resulting X-Ray report is stored with the privacy preferences of Alice at XClinic itself. XClinic collects medical records of its patients according to some privacy policies publicly available. Alice's report (along with her privacy

preferences) is subsequently sent to her insurance company MedInsure, for filing her claim. *Health_Fed* promotes privacy practices harmonisation within the various institutions by providing templates for possible policies describing different approaches to data practices. Both XClinic and MedInsure specify policies using such templates. As such upon transmission of data between the two entities, MedInsure can easily verify whether its applied privacy policy is subsumed by the XClinic one. At a later date, UUni Health Clinic requests the X-Ray information from XClinic for a routine checkup and update of Alice's health information. UUni Health Clinic has all health related information of Alice. After several weeks Alice visits another university, OUni, and finds an X-Ray as a study sample in one of their biology lectures. Even though in the sample, the personal identifying information, such as name, and identification number, have been removed from the record provided with the X-Ray, such record still provides medical data, such as abnormalities seen in the x-rays, and supporting general data, such as gender, age, race, height, weight. Alice finds that this information perfectly fits her.

Therefore, how can Alice make sure that her privacy preference with respect to the X-Ray was not violated as this information was shared among the different institutions? Can Alice know which entity has managed her own data and according to which privacy practices?

6.3 Model Entities Analysis

Model entities is required to be reappraised to apply the established framework of purpose based access control approach to distributed computing environments. In this section, we will examine the design considerations for entities according to the structure of the model.

6.3.1 Data Composition – Data Objects

Data type is proposed for data objects management. In an information processing system, the entity of a **data object** is used to denote a piece of information. Nowadays, in order to provide more competitive services, companies, enterprises and other organisations are gathering more and more personal information about their customers. Since there are many separate information units in a system, the number of data objects could be very big, which makes the management of individual data objects a complex

and time-consuming task. Considering that some data objects in a certain context may satisfy a series of common properties, to ease the management burden, these data objects are classified into object types. Here, an **object type** denotes a set of data objects sharing certain common properties. For example, data objects *Alice*, *Bob*, and *Charlie* are all persons' names, so they belong to object type *person name*. This kind of object types, which do not have any sub-type in the considered domain, are defined as *basic* object types. An object type can also be comprised of multiple data objects of other object types. This kind of object types are defined as *complex* object types. For example, in a scenario of medical care, depending on the stages in which the information is collected, and the purposes for which the information will be used, data objects denoting information collected from a patient can be grouped into different *complex* object types. For instance, the data objects, with member data objects of object types *name*, *age*, *symptom*, and *treatment history*, may form an object type *medical record*; the data objects, with member data objects of object types *name*, *billing address*, *amount*, and *payment method*, may form another object type *payment information*. Classifying data objects into object types allows us to specify and administer intended usages and necessary accesses in terms of object types instead of individual data objects, and therefore makes the management of data objects scalable. Observing this, we specified the entity of object type in our purpose based access control VDM model [YBZ08].

Data composition is used for data objects management in distributed environment. For a distributed environment, different parties will collect, use, and disseminate personal information of individuals. This chapter focuses on collaborative distributed organisations. In this environment, data objects from different *complex* object types need to be brought together for certain considerations, and to be managed under a set of common access control rules. This raises a number of challenges. On one hand, this information is necessary to speed up and facilitate user authentication and access control, and thus enhance usability of personal information. On the other hand, this information need to be protected because they may convey sensitive information that individuals may not be willing to share unless specific conditions are met. Only using object type can not handle this kind of situations. It is straightforward trying to group these data objects into a collected form. We define data composition to denote this collected form. Access control rules for these data objects can then apply to the data composition.

The design consideration about data composition is to overcome the limitations of object type in the purpose based access control approach, particularly its inability

to handle combinations of data objects belonging to different *complex* object types (in the following parts of this report, when we say object type, we refer in particular to *complex* object type). The key problem addressed for this part is how to ensure that personal information is accessed from two or more parties only if agreed privacy policies and privacy preferences are satisfied. Privacy policies and data owner's preferences thus need to be cross-checked among these parties. To this extent, the concept of **exclusive composition** is introduced. Exclusive composition is a data composition whose member data objects are exclusive to each other, which means a data user can not access all of its member data objects. It is specially used to prevent accesses to certain data object combinations. Exclusive compositions are usually set according to the data owner's preferences.

6.3.2 Role Assignment - Users and Roles

In the purpose based access control model, as to the entities of users and roles, it extends the *Role Based Access Control* (RBAC) model with the concept of conditional role, which is based on role attributes and system attributes. Role attributes are a set of properties related to the grant of access purpose to a role. Every role is associated with a set of role attributes, e.g. for the role of therapists in a medical care scenario, specialty is an attribute of the role therapist. System attributes are properties about the context of an access control system. The values of system attributes specify the conditions of the access control system. For example, working hours is a system attribute to be taken into account in a hospital access control system. The consideration of role assignment, which is about assigning roles to users, is important to protect private information. The role assignment invariants were specified in the purpose based access control model.

Proper role assignments become increasingly complex for privacy protection in distributed environment. The following examples illustrate the data processing scenarios based on which we discuss role assignment. Firstly, consider the following scenario in a medical care and scientific research environment: Assume that Bob is a user practicing in a hospital. As a doctor, he has sufficient reasons to access the records of the patients whom he treats. He must be entitled to access all parts (at least most) of the patient records, including patient names, patients' family information, patients' medical history, because he is responsible for these patients. Although he might be able to access information of all the patients he treats, he must not be allowed to access records of patients whom he does not treat. If Bob is also a staff in the medical department in a university, and he is a researcher on a specific disease. When accessing patient

information in the hospital as a medical researcher, he must be allowed to access the medical information of all the patients who have the specific symptom and have given consent that their medical information can be used for medical research, which include the patients Bob as a doctor does not treat. Some parts of the patient records, for example, the patient name and family information are not needed for the research purpose. Therefore, Bob as a researcher must not be allowed to have access to that information. In this example, the roles in two organisations, doctor in the hospital and researcher in the medical department, are set up according to the responsibility of their positions. As a researcher, he is not allowed to access information like name and family information of the patient's medical information, however, as a responsible doctor, he is allowed to have access to such information. When assigning these two roles to a same user, since the doctor has access to more information than the researcher does, the requirement that researcher is not allowed to have access to certain information can not be fulfilled. Possible problem in the above example is not about the role set up, but is about the role assignment to user.

Consider another example of a scenario in financial environment: The mortgage applying process consists of a number of steps, including preparation of application, asking for credit rating, and issuing mortgage. Normally, the preparation of application and issuing mortgage take place in the lender organisation, and credit rating is provided by a third party organisation. For this task to be fulfilled, each step must be successfully completed. The separation of duty principle states that two or more different people should be responsible for the completion of a task, which is to prevent fraud by separating a task into subparts and spreading responsibility of each subpart over several people. Therefore, for the mortgage application task to be fulfilled according to the principle of separation of duty, each step, especially the steps of credit rating and mortgage approval, must be performed by different users. Some previous work has explored the support for separation of duty principle within an organisation, but applying separation of duty principle in each single organisation cannot prevent a user from getting a role for providing credit rating from one organisation and a role for mortgage approval from another organisation.

The problems of role assignment conflicts and breach of separation of duty may arise in distributed environment. The design consideration for this part is to analyse the requirements on role assignment for privacy preservation in distributed environment, and propose solutions to address the issue of role assignment conflicts and to support separation of duty principle.

6.3.3 Purpose Assignment - Purposes

The privacy policy defines a set of rules. Each rule can allow a set of data users (role) to perform actions on data objects for certain purposes. These rules can be written into invariants of the model. Such a privacy policy holds good for single organisation; however, when it comes to privacy preserving among organisations, it requires to consider more issues, and it needs extra information to achieve that.

6.3.3.1 Inter-Organisational Policies

Based on the medical care example in Section 1.2.1, consider the following scenario: the patient goes to a hospital, and he may also go to a local clinic. His information, including medical record, is stored in the clinic as well. For preserving privacy of patient, the clinic also publishes its privacy policy. Now, both the hospital and the clinic have certain privacy policies on the patient's information.

Purpose based access control approach can provide privacy preservation for each organisation according to their privacy policies. All accesses within an organisation should adhere to its privacy policy and it could happen without much problem. However, when considering privacy policies of these two organisations together, one possible problem that may arise is purpose conflict. Let us consider an example of the genetic information of a patient. This involves the patient, the clinic, and the hospital. The patient's GP in the clinic may have to obtain the patient's genetic information for processing the diagnosis of certain disease. For diagnosing certain disease of the patient referred by GP, the doctor in the hospital asks for access to the genetic information as well.

The rules in privacy policies for access genetic information for both clinic and hospital are shown as follows:

- Privacy Policy of Clinic:

  For the diagnosis of certain disease, only the responsible GP may access the patient's genetic information.

- Privacy Policy of Hospital:

  For the diagnosis of certain disease, the responsible doctor may access the patient's genetic information.

Next, we analyse the possible purpose assignment conflicts in this scenario.

6.3.3.2 Privacy Policy Conflicts

Conflicts arise when two organisational privacy policies of the two organisations exchanging data doesn't match with each other. Referring to the privacy policy of the hospital, it allows access to the patient's genetic information by responsible doctor whereas the privacy policy of the clinic prohibits it. For inter-organisation environment, organisations have to conform to the rules when they interact and exchange data even though the policies are different. For purpose based access control approach, in privacy policy, the entities "access" and "purpose" explain that access will be taken on data object that is collected for the specified purpose(s). An "allow" from the evaluation authority means the requester is able to perform the access on the given data object. We give the example of privacy policy rule for obtaining the patient's genetic information in the clinic:

- Only the responsible GP is allowed to access the patient's genetic information for the purpose of "diagnosing" for certain disease.

So far there is no problem. Now, for this disease, the GP refer this patient to the hospital. The diagnosing from the doctor in the hospital has no idea about the privacy policy rule of the clinic on the patient's genetic information. The data object is medical record and the purpose is certain disease diagnosing. The doctor wouldn't be able to perform diagnosing without obtaining the medical record. The clinic has no knowledge of the process of the diagnosing, which is performed to conform to the hospital's privacy policy. So the patient's information obtained is treated by the hospital as its own data conforming to the hospital's own privacy policy. Therefore, enabling the responsible doctor to access the patient's genetic information leads to a conflict.

6.3.4 Information Flow - Data Access

According to the principle of purpose compliance, a user may access an object, if the purpose of its current request is contained in the set of purposes for which data object of the object type is obtained.

We talked about data composition in Section 6.3.1. A data object consists of several attributes. In changing some attributes of a data object, a user may refer to attributes from other data objects. In this case, information flow occurs. For example, when a patient registers with a hospital, his medical history is given and filled in registration

information. The medical history may be referred by a doctor and put in the patient's medical record.

Information flow happens when there is data exchange, it is necessary for some operation of the system. However, illegal information flow may occur if there is problem with process design. The following scenario shows that illegal information flow may happy unintentionally. Suppose in a process design, a doctor while performing a request could read medical record and write sensitive data from medical record to medical history, which is part of registration information. Consequently, the receptionist that is performing a request could read data from medical history, which was written to registration information. Hence, the principle of purpose binding could be violated.

6.4 Preliminary Notions

Our approach relies on the several important notions of data composition, Chinese Wall policy, privacy policy and privacy preference, and federated identity management. In what follows we provide background information about these notions that is relevant for subsequent discussion.

6.4.1 Data Composition and Object Type

Data object type and data composition approach data objects from different perspectives. Object type concerns data objects that appear within one context, while data composition focuses on combinations of data objects from different object types. In this research, data composition especially concerns preventing accesses to certain combinations of data objects. Individual member data objects of a data composition can come from different object types within one organisation, or from different organisations. For example, some data objects of a data owner are managed by one organisation, and other data objects of the same owner are managed by another organisation. If the owner wants accesses to some of these data objects to be controlled by a same policy, then those data objects can be formed into a data composition. The access control system needs to provide mechanism to control this kind of accesses.

The following scenario illustrates the necessity of introducing data composition. Following the example in Section 6.3.1, for an insurance company to fulfill its business purpose, it may need some member data objects from object type *medical record*, and some member data objects from *payment information*. Only using object type can not

sufficiently handle this kind of combination. To ease the management of accesses, these data objects should be grouped into a data composition. Access control policies and privacy preferences can then be defined on this data composition.

Revisit the scenario mentioned earlier: a user has accessed the data objects of one data owner's *name* and *address* in the *registration information*, and the user wants to access the data objects of the data owner's *address* and *phone number* in the *contact information*. According to the data owner's preferences, which are specified for these data objects by the data owner, the user is allowed to access either *name* or *phone number*, but not both, i.e. he is not allowed to know which name is associated with which phone number. Suppose the user has got the *name* and *address* pair by issuing a request, to which he has been granted access, and suppose in a subsequent session he can perform a request to get the *address* and *phone number* pair, to which he is also granted access. Now he can combine the results from these two requests to deduce the *name* and *phone number* pair to some accuracy.

This example shows that when a data owner's information spans more than one object types, simply restricting accesses to data objects or object types by separate policies, and returning results on policy satisfaction, is not sufficient to guarantee the prevention of accesses to certain combinations of data objects. In our previous model, using object type does not provide a mechanism to control the total set of information disclosed to a user over time, so this research proposes the concept of data composition.

In our purpose based access control model [YBZ08], the concept of *purpose* takes the data owner's preference over data usage into consideration, which is an important feature for privacy preservation. Purposes are divided into two categories: intended purposes and access purposes. The former is related to data objects and object types, and the latter is related to data accesses. The intended purposes specify the intended usage of a data object, and the access purposes specify the intention of an access request to a given data object. Privacy preserving access control is to ensure that data objects can only be used for their intended purposes, and the access purposes should be compliant with the intended purposes. This property is checked for object types in the proposed model. For data composition, this property also needs to be maintained.

6.4.2 The Chinese Wall Policy

The purpose-based access control approach restricts accesses to individual data objects and data object types. As we discussed in previous section, because the granularity of data object combinations may vary for some specific services, the privacy preserving

access control mechanism ought to provide support for data composition. Exclusive composition is specially proposed to prevent accesses to certain data object combinations. It tries to control the total set of information disclosed to a party over time. In this sense, there is a similarity between our approach and the Chinese Wall security policy.

The Chinese Wall policy [BN89, Kes92] is a well known information control policy. The policy states that an access is only allowed if the requested information is not in conflict with any other information that is already held. In the policy, entities to be secured are grouped according to some common properties. These groups are then placed into different conflict of interest classes. At the start, any entity in any group may be accessed by the target user. As soon as an entity in a particular group has been accessed, access to entities in other groups belonging to the same conflict of interest class is disallowed. Only entities in the same group as the first entity accessed can further be requested. All other entities belonging to groups in other conflict of interest classes can still be accessed. Further restrictions are then added as subjects access entities, until at some point, a subject will only have access to those entities already accessed previously.

The Chinese Wall policy can be used to specify control over information when conflicts of interest arise. It is particularly applicable to commercial security systems. As an example, this policy could be used to govern access to information by a consultant in a consulting firm. In principle, a consultant has access to all information in the firm. As soon as the consultant has read some information about an insurance firm *I*, which is a client of this firm, access to the information relating to client insurance firms other than *I* should be revoked. This is to prevent possible conflicts of interest and decisions based on confidential information about competitors of firm *I*, who may also be clients of the consulting firm. The consultant should still have access to other information about firm *I*. He still should have access to information that do not contain information about insurance firms.

6.4.3 Privacy Policy and Privacy Preference

Privacy policy states who the *recipients* will be for the *data*, the *purpose* for which this data will be used, and how long the data will be *retained*. Data in a privacy policy can be represented at different levels of granularity. They can refer to aggregate data, or they can refer to more specific pieces of information, such as, name or address. The current vocabulary adopted by the P3P standard is not adequate for automatically and

efficiently matching policies. We need to operate on a more articulated terminology, using which we can compare and relate different values assigned to a same element of the policy. In our work, the data object refers to the smallest granularity data. They are then grouped into object types and data compositions. In particular, it is important to extend and define semantic relationships among elements in the *purpose* element. To achieve this goal, we consider the hierarchy supporting the specification of privacy preferences set by users.

Privacy preference is an important entity for a privacy enforcement system. When data objects are collected online or offline, data owners can specify their privacy preferences on how the organisation can use the data objects. In particular, data owners may set exclusive composition in their privacy preferences. Data owners may change their privacy preferences at any time after the data has been collected. In addition, data objects may be grouped together and the group of data objects can be handled under a single rule. Privacy preference management is to link these preferences with actual data objects with which they are associated. We assume each group of data objects is associated with a privacy preference tag. Whenever a user requests to access a group of data objects or some specific data objects, we can retrieve the corresponding privacy preferences from the privacy management system.

6.4.4 Federated Identity Management

An emerging approach for protecting personal information while at the same time enhancing information usability is to focus on inter-organisation management of identity information. This is referred to as *federated identity management*. Digital identity is the digital representation of the information known about an individual or organisation. The goal of a federated approach to digital identity management is to provide protected environments enabling personal information sharing. To date several ongoing initiatives [Lib09, OT02, SWI09] are developing protocols and platforms for federated management of digital identities.

Although federating identities greatly simplify the task of collecting and distributing personal information in the federation, no satisfying mechanisms are currently provided to protect data owners' privacy and for privacy policy matching in collaborative environments. As organisations in a federation correspond to independent entities, they may adopt privacy practices that are not homogeneous. Uncontrolled personal information sharing may result in privacy breaches and threats like identity theft, and in the lack of compliance with respect to the privacy policies advertised by various

service providers and the privacy preferences set by data owners.

A suitable solution to the problem of privacy in a federated environment should satisfy an important requirement, which is to provide mechanisms for facilitating privacy policies matching and harmonisation among federated organisations. Such mechanisms would make it possible to determine whether or not the transfer of personal information from one organisation to another would violate the privacy policies stated by the former. Notice that allowing an organisation to transfer personal information to another organisation is important in order to maximise user convenience. Privacy conscious users may in fact have their own preferences concerning the use of their personal information.

In this chapter, we address this requirement by developing an approach that supports the privacy controlled sharing of personal information and the harmonisation of privacy policies based on the notion of *subsumption*. Subsumption is used on policies defined over equal or similar class of data in order to determine if they are in conflict or if one implies the other. To facilitate policy harmonisation in a federation, we assume some predefined policy templates to be available for policy specification. The federated organisations may either exploit the templates or may specify customised policies describing their own practices.

We base our approach on a rich privacy vocabulary rather than on the vocabulary provided by the Platform for Privacy Preference (P3P) method [Wor07]. We employ Enterprise Privacy Authorization Language (EPAL) [AHK$^+$03] vocabulary hierarchies to address the limited expressive power of the original P3P vocabulary. Moreover, we make use of data composition to establish a common vocabulary for attributes, credentials, and data produced and exchanged across the federation. The use of data composition makes it possible for interacting parties to automatically detect semantic relationships among different attributes and reason about policy subsumption.

Federated organisations manage and collect personal information. As such a data owner will register at his/her own local organisation and then he/she will submit other personal information while interacting with organisations to gain access to specific services or data. As no centralised identity provider exists, such information is not stored at a unique server but is distributed among various respective organisations the data owner has visited. Federated organisations, besides interacting with data owners to provide them with services, also interact among each other in order to support the federated management of digital identities. Federated organisations exchange personal information to automatically authorise data owners to access services and resources

and so to avoid requiring multiple submissions of these attributes and credentials from data owners.

6.5 Related Work

In this section, we review related work in respect of data composition, role assignment, and identity federation.

6.5.1 Data Objects - Data Composition

Yolanta Beresnevichiene specified object based separation of duty constraints based on the Chinese Wall policy [Ber03]. In this method, the corporative information is organised in a hierarchical structure with three levels:

1. *objects*, which are individual information items;
2. *domains*, which group objects together under some selected attributes;
3. *conflict of interest classes*, which group domains together based on conflict of interests. Domains within the same class are conflicting.

The Chinese Wall policy then states: access to the object being requested is granted, only if the object is either in the same domain as an object which has already been accessed by that user, or belongs to an entirely different conflict of interest class.

As in the example in Section 6.4.2, the domains are defined by individual firms, and the conflict of interest classes are their sectors of business. Some firms could belong to more than one conflict of interest class if their activities span several business sectors. For example, if a consultant accesses information about a firm *F*, which runs business in both insurance and banking sectors, then accesses to other insurance firms and other banks are revoked from that consultant.

In Beresnevichiene's work, the separation of duty constraints specify preventing accesses to objects in other domains within the same conflict of interest class. There are also other approaches [LO02, PBS05] taking Chinese Wall policy into consideration for security protection. These approaches all try to prevent accesses to certain data combinations. However, they concern more about avoiding conflict of interest. They group the same type of information into a conflict of interest class. Then, in each conflict of interest class, a user can only access one instance of a type. In the context of

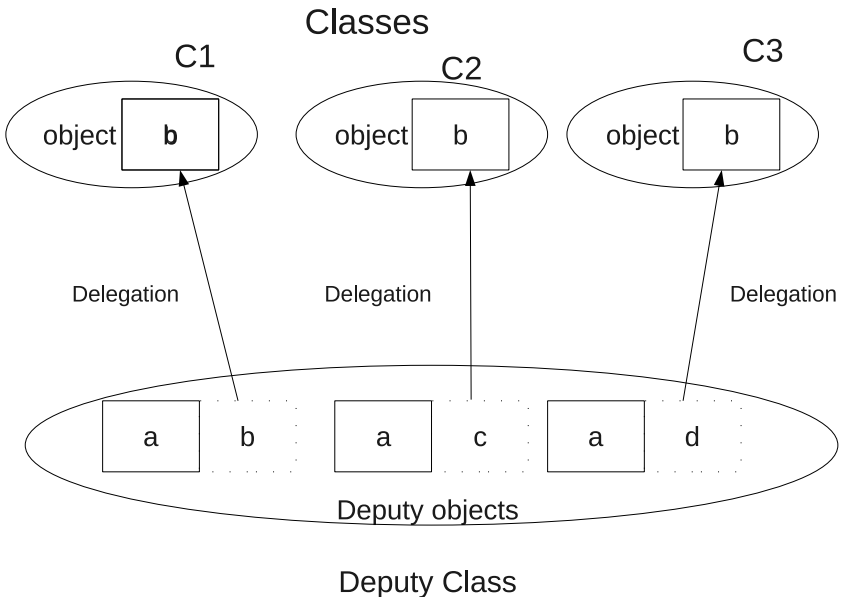

Figure 6.2: Deputy Mechanism

privacy protection, the data objects are individual's private information items. For the same kind of data objects, rather than grouping them into conflict of interest classes, they are grouped into logical groupings, such as the aforementioned object types of *registration information* and *contact information*. Privacy policies are defined to control accesses to data objects of an object type. Exclusive compositions are defined by the individual arbitrarily to prevent any given party from learning too much about him. In this sense, even data objects belonging to different conflict of interest classes may be treated exclusive to each other. Moreover, the Chinese Wall policy cannot enforce the privacy requirement of purpose binding.

Yi Ren et al proposed a privacy data model based on deputy mechanism [RLTY07]. The privacy data model in this approach consists of two components: (1) atomic privacy object model, which is used to protect privacy of individual; (2) composite privacy object model, which is used to protect privacy among individuals. Composite privacy object can be created by combining atomic privacy data according to some special combination semantics. Deputy is an inheritance mechanism. It is based on links from deputy objects to their source objects and restricted by switching operations. A deputy object inherits attributes and methods from several source classes, as shown in Figure 6.2. If the switching operations are not defined for some attributes and methods

of source classes, then the deputy classes cannot inherit them. The deputy mechanism aims to restrict accesses to certain combinations of data objects. However, it concerns relationships among individual objects when group these objects into composite objects. Since relationships among data objects may vary in different contexts, and data owner's preferences can be set arbitrarily, only using deputy mechanism to form composite objects can not handle the scope of data combinations set by data owners' preferences.

The discussion above has shown that, the Chinese Wall policy cannot sufficiently address privacy aspects of data combination. Neither of the approaches is capable of enforcing purpose binding requirement. If a data owner wants some of his data objects to be controlled according to his preferences, the existing privacy preserving approaches can not provide sufficient support for those data combinations defined by his preferences. Therefore, exclusive composition is proposed for privacy preservation of this kind of data combinations. The formal specification of constraint and its implementation will be presented in the following sections.

6.5.2 Users and Roles - Role Assignment

In systems implementing RBAC, access policies are normally determined at an organisational level. In an organisation, in order to specify privacy policy, some attributes are assigned to role conditions and system conditions. The ability to support inter-organisational access policies is not a support feature. Research has been undertaken to extend RBAC framework to support distributed access control features.

In [BLM01], the authors stated that the standard RBAC approach is not suitable when individual exceptions to default access policies need to be supported. The approach they proposed involves storing exceptions to the default policy with the affected records themselves. This is not entirely consistent with the role based approach which stipulates that all permissions are held by roles. As the exceptions are effectively negative permissions, an authorisation model was proposed in [LLT00] for distributed health care environment of the UK National Health Service, in which access privileges can be both granted and denied through the use of positive or negative confidentiality permissions. Four different confidentiality permission types that have a fixed hierarchical precedence are described in the model. Higher order confidentiality permission types can override lower order types. Similar approach was proposed in [RCHS03]. This kind of approaches allow flexible policy expressions since the privileges held by a role can be allowed or denied to other roles, and they also support individual exception

to default access policies, which is an important feature to role attributes. Although these approaches are stated to be suitable for distributed use, their implementations in distributed environment need further exploration.

Mavridis et al [MPKB99] presented a security policy defined on the basis of RBAC components and supports both mandatory and discretionary features. The access control mechanisms used in their method are hyper node hierarchies, which separate entities into different levels in hierarchy according to their security importance. Then user roles, data sets, and user locations are arranged into different levels in hierarchies, and access decisions are made according to the three dimension access matrix. This method extends privacy protection with the concept of user location control. Location is concerned mainly for user roles. But arranging roles into different levels and restricting location of role use can not prevent role conflicts from different organisations.

Ni et al [NLBL08, NTBL07] proposed privacy aware role based access control models, which extend RBAC model in order to provide support for expressing privacy policies, taking into account features like purposes, conditions, and obligations. In this approach, privacy policies are expressed as permission assignments, which consist of additional components representing privacy related information, including the intended purpose, along with the conditions under which the permission can be given, and the obligations that are to be finally performed. Conflicts between two privacy aware role based access control permission assignments are discussed. On the one hand, it only takes some trivial system conditions into consideration for conflict detection; on the other hand, it only considers conflicts of roles within one organisation.

The concept of context is proposed to express different types of extra conditions or constraints that control activation of rules expressed in the access control policy. Different kinds of contexts were investigated in [CM03], including *temporal context*, *spatial context, user-declared context*, *prerequisite context,* and *provisional context*. Context was introduced to the basic RBAC model in [SH05, WLF08]. They proposed to capture security relevant context of the environment in which access requests are made, and further exploration of context information in distributed environment is needed.

Separation of duty is a well-known principle of computer security. However, it is difficult to express and effectively enforce this principle in security systems. Within role-based access control models, separation of duty has been implemented with mutual exclusion of roles [FBD99, SCFY96]. The term ‘mutual exclusion’ usually has the meaning that some form of conflict exists between pair of roles. Users are allowed to

access only one role from this pair. In [Ber03] the author further explored a systematic framework for applying separation of duty policies in the RBAC model. Constraints were presented to enable application of different separation of duty policy variations. The constraints deal with user assignment to roles, and with permission distribution among roles, including role hierarchies. However, these work didn't consider inter-organisational issues. In [Ber03], it mentioned extending their approach to distributed environment, but only authentication issue was analysed.

6.5.3 Identity Federation

Our work is partly motivated from the existing initiatives related to federated digital identity management. Their goal is to provide a controlled and protected environment for managing identities of federated users. In this section, we explore the most relevant federated digital identity management initiatives.

There are several emerging standards for identity federation like Liberty Alliance [Lib09] and WS-Federation [OT02]. Because these projects are similar, we describe Liberty Alliance in more detail. The Liberty Alliance project is based on Security Assertion Markup Language (SAML) and provides open standards for single sign-on (SSO) with decentralised authentication. SSO allows a user to sign-on once at a Liberty-enabled site in order to be seamlessly signed-on when navigating to another site without the need to authenticate again. This group of Liberty-enabled sites is a part of what is called a *circle of trust*, which is a federation of service providers and identity providers having business relationships based on the Liberty architecture. The identity provider is a Liberty-enabled entity that creates, maintains and manages identity information of users and gives this information to service providers.

Shibboleth [Shi09] is an initiative by universities that are members of Internet2. The goal of such initiative is to develop and deploy new middleware technologies that can facilitate inter-organisational collaboration and access to digital contents. It uses the concept of federation of user attributes. When a user at an institution tries to use a resource at another, Shibboleth sends attributes about the user to the remote destination, rather than making the user log into that destination, thus enabling a seamless access. The receiver can check whether the attributes satisfy its own policies.

Our approaches differs with respect to these approaches in that we do not rely on a central identity provider providing all user attributes. Personal information in our framework is distributed within different federated organisations, each of which can effectively be an identity provider. Besides, these federated identity management

systems do not provide a mechanism for local and global matching.

6.6 Purpose Based Approach for Distributed Environment

This section describes the implementation of purpose based access control approach on privacy protection in distributed collaborative environment.

6.6.1 Model Entities

In this section, the entities of our model and the system state are defined. Firstly, the data structure for representing data objects is specified. Then, the user and role model is presented. The entity of purpose, and the entities for accessing data objects, are then defined. Finally, the system state based on the definitions of these entities is specified.

6.6.1.1 Data Object Model with Data Composition

This section describes a data object model to provide a suitable structure for representing data objects, with the extension of data composition.

In order to introduce the concept of data composition into our previous approach, we need to re-examine the entities in the data object model. Recall the exemplar scenario described in Section 6.4.1, a user can not access the data objects of both a data owner's *name* and his *phone number*. In this case, the data objects of *name* and *phone number* form a data composition. They are *exclusive* to each other. In other words, if a user has accessed one data object in a composition, he should not be allowed to access other data objects in the same composition.

To specify the constraint on data composition, it is necessary to analyse the information representation in the information processing system:

- Data Objects, which are individual information units;
- Object Types, which are sets of data objects that appear in the same context and satisfy some common properties;
- Data Compositions, which are sets of data objects that are combined together for specific consideration or service. This form of data combination is more general than object types.

Let *Object* denote the set of data objects. Object type is a set of data objects, so an object type can be expressed as $Object$-set.

Next, we define object type attributes and attribute values to specify the properties of object types.

Definition 1 (Object Type Attributes) *denoted as $TypeAttr$, are a set of attributes associated with an object type, which describe the properties about the collection of, and access to, this type of data objects.*

Definition 2 (Object Type Attribute Values) *denoted as $AttrValue$, are a set of possible values of the object type attributes.*

In the sense of representing data objects that appear in the same context, a data composition is similar to an object type. But besides that, data compositions are also capable of representing data combinations with different types of data objects. This work especially focuses on data compositions whose member data objects are exclusive to each other, which are defined as *exclusive compositions*.

Let $ExComp$ denote a set of exclusive compositions.

Each data object $object \in Object$ is associated with two functions: $TypeOf$, which returns the object type of a given data object; and $ExCompOf$, which returns the exclusive composition sets of the given data object.

Definition 3 (Access History) *denoted as $AccHis$, is a set of objects a user has accessed.*

The data object model in our system is then specified as:

Definition 4 (Object Data Model)

$$
\begin{aligned}
ObjectDataModel :: object &: Object\text{-set} \\
type &: (Object\text{-set})\text{-set} \\
exComp &: (Object\text{-set})\text{-set} \\
typeAttr &: TypeAttr\text{-set} \\
attrValue &: AttrValue\text{-set} \\
accHis &: AccHis\text{-set} \\
TypeOf &: Object \rightarrow Object\text{-set} \\
AttrOf &: Object\text{-set} \rightarrow TypeAttr\text{-set} \\
ValueOf &: Object \times TypeAttr \rightarrow AttrValue
\end{aligned}
$$

$ExCompOf : Object \rightarrow (Object\text{-set})\text{-set}$

inv mk-$ObjectDataModel$(*o, t, ec, ta, av, ah, To, Ao, Vo, Eo*) $\triangleq$

(dom $To = o \wedge$ rng $To \subseteq t$) $\wedge$

(dom $Ao = t \wedge$ rng $Ao \subseteq ta$-set) $\wedge$

(dom $Vo = o \times t \wedge$ rng $Vo \subseteq av$) $\wedge$

(dom $Eo = o \wedge$ rng $Eo \subseteq o$-set-set)

The invariant for $ObjectDataModel$ states that

1. *object* is a set of data objects in a system
2. *type* is a set of object types
3. *typeAttr* is a set of type attributes
4. *exComp* is a set of exclusive compositions
5. *attrValue* is a set of attributes values
6. *accHis* is a set of data objects that have been accessed by a user
7. $TypeOf : Object \rightarrow Object$-set is a total function giving the type associated with a data object
8. $AttrOf : Object$-set $\rightarrow TypeAttr$-set is a total function giving the type attributes associated with each type
9. $ValueOf : Object \times TypeAttr \rightarrow AttrValue$ is a total function giving the value of the attributes associated with objects.
10. $ExCompOf : Object \rightarrow (Object$-set$)$-set is a function that returns the set of exclusive data compositions of a data object.

The object data model *OM* in our system, of type $ObjectDataModel$, can then be represented as a tuple $\langle$*object, type, exComp, typeAttr, attrValue, AccHis, TypeOf, AttrOf, ValueOf, ExComOf*$\rangle$.

In this section, we defined the data object model to represent data objects in our system. Next we specify the structure for representing the subjects.

6.6.1.2 Users and Roles

The purpose-based access control approach [BBL05, BL08] extends the RBAC model with the concept of conditional role. It is based on role attributes and system attributes. In this part, the formal definitions of role attributes, system attributes, and conditional role are presented.

Firstly, we specify the entities of user and role in the basic RBAC model.

Users are the active entities in a system, e.g. the staff in a medical care scenario.

Let *User* denote the set of users.

The roles in a system reflect the responsibilities of positions or job descriptions in the context of an organisation, e.g. therapist, registration staff, or billing staff in a medical care scenario.

Let *Role* denote the set of roles.

UserRole: $User \leftrightarrow Role$ is the relation between users and roles.

A user may be assigned with many roles, but the user may not exercise all his roles at the same time. The roles that a user is currently exercising are "active" roles.

Active Roles AR: $User \rightarrow Role$-set is a function that returns the roles for which a user is active.

Existing role definitions are predefined for access permission assignments, so they do not adequately specify the set of users to whom we wish to grant an access purpose. The concept of *conditional role* is then introduced. It is based on the notion of *Role Attributes* and *System Attributes*. They are then specified accordingly.

Definition 5 (Role Attributes) *denoted as $RoleAttr$, are a set of properties related to the grant of access purpose to a role.*

Every role $r \in Role$ is associated with a set of role attributes, e.g. the specialty of therapists in a medical care scenario.

RoleAttrOf: $Role \rightarrow RoleAttr$-set is a function that returns the set of role attributes of a role.

Let *RoleAttrValue* denote the set of possible role attribute values.

RoleAttrValueOf: $Role \times RoleAttr \rightarrow RoleAttrValue$ is a function giving the value of role attributes associated with a role.

For an access control system, system attributes are used to describe the properties of a system context. For example, the working hours within a hospital is a system attribute.

Definition 6 (System Attributes) *denoted as $SysAttr$, are properties about the context of access control system.*

The values of system attributes specify the conditions of the access control system. Let $SysAttrValue$ denote the set of all possible system attribute values.

SysAttrValueOf: $SysAttr \rightarrow SysAttrValue$ is a function that returns the value of the system attributes in a system.

With role attributes and system attributes, we can now define conditional role.

Definition 7 (Conditional Role) *refers to a role with some conditions attached to it.*

$CondRole :: r : Role$

$cond : RoleAttrValue \times SysAttrValue \rightarrow \mathbb{B}$

where $\mathbb{B}$ is the boolean set, and $cond$ is a truth-valued function.

$CR : CondRole$-set is used to denote the set of conditional roles.

Current Conditional Role CCR: *User* $\rightarrow$ *CR* is a function that returns the conditional role a user is currently exercising.

The data objects that a user has accessed need to be kept in the user's access history.

$Accessed : User \times Object$-set $\rightarrow AccHis$ is a function which returns the set of the data objects a user has accessed.

The user and role model is then specified as:

Definition 8 (Role Model)

$$
\begin{aligned}
RoleModel :: role &: Role\text{-set} \\
user &: User\text{-set} \\
UserRole &: User \leftrightarrow Role \\
AR &: User \rightarrow Role\text{-set} \\
roleAttr &: RoleAttr\text{-set} \\
roleAttrValue &: RoleAttrValue\text{-set} \\
RoleAttrValueOf&: Role \times RoleAttr \rightarrow \\
&\quad RoleAttrValue \\
sysAttr &: SysAttr\text{-set} \\
sysAttrValue &: SysAttrValue\text{-set} \\
SysAttrValueOf &: SysAttr \rightarrow SysAttrValue \\
CR &: CondRole\text{-set} \\
CCR &: User \rightarrow CR
\end{aligned}
$$

$$Accessed : User \times Object\text{-set} \rightarrow AccHis$$

The role model *RM* can be represented as a tuple ⟨*role, user, UserRole, AR, roleAttr, roleAttrValue, RoleAttrValueOf, sysAttr, sysAttrValue, SysAttrValueOf, CR, CCR, Accessed*⟩.

6.6.1.3 Purpose

Data is collected for certain purposes. For example, in medical care environment, data may be collected for registration or diagnosing. Each data access also serves a purpose. It is necessary to determine purposes for which data is collected and for data accesses. The entity of purpose is specified in this section.

Definition 9 (Purpose) *denoted as $Purpose$, is the intention of data collection or data access.*

Purposes are organised in a tree structure, which is called purpose tree. Let P_T denote the purpose tree. Each node represents a purpose in $Purpose$, and each edge represents a hierarchical relation between two purposes.

There are certain relationships among purposes. The purposes are organised into purpose tree according to these relationships. Next we define relationships among purposes.

The nodes in a purpose tree can be classified into general or special according to the relationships among the nodes.

Definition 10 (Specialisation (Generalisation)) *If p_1, p_2 are two nodes in a purpose tree, then we say p_2 is a specialisation of p_1 (or p_1 is a generalisation of p_2) if there exists a downward path from p_1 to p_2.*

$Specialisation$: $Purpose \times Purpose \rightarrow \mathbb{B}$ is a truth-valued function that characterises the specialisation relation. $Generalisation$ relation can be defined accordingly.

Purposes, depending on their association with objects and subjects, may be called intended purposes or access purposes, respectively.

Definition 11 (Intended Purpose) *is the specified usages for which the data objects are collected.*

Intended purpose specifies the property of data objects.

IP: $object(OM) \cup type(OM) \rightarrow Purpose$-set is a function that returns intended purposes of a data object or type. Here, *object* and *type* are defined in object data model *OM*, *Purpose* is the set of purposes.

Definition 12 (Access Purpose) *is intentions for accessing data objects.*

Access purpose specifies the property of data accesses.

$AuthorisedAccessPurpose\ AAP$: $CR(RM) \rightarrow Purpose$-set is a function that returns authorised access purposes.

The purpose model is then specified as:

Definition 13 (Purpose Model)

$$
\begin{aligned}
PurposeModel :: purpose &: Purpose\text{-set} \\
Specialisation &: Purpose \times Purpose \rightarrow \mathbb{B} \\
Generalisation &: Purpose \times Purpose \rightarrow \mathbb{B} \\
IP : object(OM) \cup type(OM) &\rightarrow Purpose\text{-set} \\
AAP &: CR(RM) \rightarrow Purpose\text{-set}
\end{aligned}
$$

The purpose model *PM* can be represented as a tuple ⟨*purpose, Specialisation, Generalisation, IP, AAP*⟩.

6.6.1.4 Requests, Transactions, and Accesses

This section specifies the entities for accessing data objects, namely, requests, transactions and accesses.

Definition 14 (Request)

$$
\begin{aligned}
Request :: obj &: object(OM) \\
ap &: purpose(PM)
\end{aligned}
$$

When a conditional role *cr* wants to access an object obj, it makes a request for the data object, with a particular access purpose *ap*. The request is denoted as a 2-tuple $\langle obj, ap \rangle$. For example, the request from a GP to access treatment history for the purpose of diagnosing has the form of $\langle treatment\ history, diagnosing \rangle$.

We use $Req : Request$-set to denote the set of requests in a system.

Current Request CReq: $CR(RM) \rightarrow Req$ is a function that returns the request currently presented.

Definition 15 (Transactions) *denoted as Transaction, are the executions or procedures to perform a request.*

To ensure an object is accessed in a controlled manner, only specified transactions may be allowed. For example, the diagnosing request consists of three transactions: reading treatment history, analysing medical test results, and appending new diagnosis to the treatment history.

Current Transaction CT: $CR(RM) \rightarrow Transaction$ is a function that returns the transaction currently performed.

Authorised Transactions AT: $Req \rightarrow Transaction$-set is a function returns the authorised transactions for a request.

Next we define entities about accesses in our system. Model entities related to object accesses are access modes, necessary access, and current access.

Definition 16 (Access Modes) *are the modes of accesses performed on data objects.*

Let *AccMode* denote the set of access modes. $AccMode$ = $\{create, read, write, append, delete\}$

$Mode$: $AccMode$-set denotes the set of access modes in a system.

Definition 17 (Necessary Accesses) *are the accesses that are needed to achieve an access purpose.*

For access purpose, it has to be defined in advance what accesses are needed to achieve that access purpose.

$NecAcc :: ap : Purpose$
$tp : type(OM)$
$trans : Transaction$
$x : Mode$

NA: $NecAcc$-set denotes the set of necessary accesses.

Definition 18 (Current Accesses) *are accesses that a conditional role is performing.*

$CurAcc :: cr : CR(RM)$
$obj : object(OM)$
$x : Mode$

CA: $CurAcc$-set denotes the set of current accesses.

The access model is then specified as:

Definition 19 (Access Model)

$AccessModel :: Req : Request$-set
$CReq : CR(RM) \rightarrow Req$
$Trans : Transation$-set
$CT : CR(RM) \rightarrow Trans$
$AT : Req \rightarrow Trans$-set
$Mode$: $AccMode$-set
$NA : NecAcc$-set
$CA : CurAcc$-set

The access model *AM* can be represented as a tuple ⟨*Req, CReq, Trans, CT, AT, Mode, NA, CA*⟩.

Having defined the entities in our purpose-based access model, the system state can be defined.

6.6.2 The State of System

The formalisation of the model consists of the specification of system state. System state consists of the state variables corresponding to the components defined in previous sections: OM, RM, PM, AM.

The state space, without invariant and initialisation condition as yet, is written as follows:

state $DPPS$ of
$OM : ObjectDataModel$
$RM : RoleModel$
$PM : PurposeModel$
$AM : AccessModel$
inv ...
init ...
end

The initialisation condition is defined as:

init $\sigma \triangleq \{\sigma.object(OM) = \{\} \wedge \sigma.type(OM) = \{\} \wedge$
$\sigma.purpose(PM) = \{\} \wedge \sigma.AAP(PM) = \{\mapsto\} \wedge$
$\sigma.Req(AM) = \{\} \wedge \sigma.Trans(AM) = \{\} \wedge$
$\sigma.CA(AM) = \{\} \wedge \sigma.NA(AM) = \{\}\}$

The entities in purpose based access control model and the system state have been introduced in this section. We are then able to specify privacy requirements in privacy

policy. The state invariants corresponding to the requirements are specified in the next section.

6.6.3 Privacy Constraints

In Chapter 4, we specified invariants through the data processing stages, including creation of data objects,role authorisation, and data access. This section presents constraints for distributed environments.

6.6.3.1 Data Composition Constraint

The basis of data composition constraints is that users are not allowed to access all data objects in an exclusive composition. In other words, the data object the user is requesting is not in conflict with any other data object that he has already accessed. The requirement can be assured with the following constraints:

(d) Data Composition Constraint

(d1) A user may only have current access to a data object, if this data object is not in the same exclusive data composition as other data objects the user has already accessed.

Given two successive system states v, $v^{'}$,
$v = (OM, RM, PM, AM)$,
$v^{'} = (OM^{'}, RM^{'}, PM^{'}, AM^{'})$,
$(v, v^{'})$ satisfies data composition constraint, iff
$\forall\, user \in User(RM)$, $cr = CCR(RM)(user)$, $\forall\, obj_j \in Accessed(cr)(OM)$,
$obj_i \in object(OM)$, $ap \in purpose(PM)$:
$\langle obj_i, ap\rangle = CReq(AM)(cr) \wedge \langle ap, type_i, CT(AM)(cr), x\rangle \in NA(AM)$
$\wedge\, obj_i \in Accessed(user)(OM^{'})$
$\Longrightarrow ExCompOf(obj_i) \cap ExCompOf(obj_j) = \phi$

6.6.3.2 Role Assignment Constraints

When we talk about privacy preservation in distributed environment, role attributed and system context information should also be captured. In this sense, roles are distinguished into global roles and local roles. A global role is granted access purposes which fulfill part of a task consists of cooperation from several organisations. The role for credit rating takes part in the process of mortgage application, which needs cooperation from lender organisation, so this role is a global role. Local roles are roles that

are granted access purposes to fulfill all its purposes within its residing organisation. Roles in previous RBAC research fall in this category.

The problems we spotted in section 6.3.2 and the 'mutual exclusion' have similar purpose, in the sense that information held in objects is distributed among different conflicting groups, and the rules are specified on how these groups are then assigned to the users. Constraints on data composition states that when a user makes an access to a data object, if the data object has a pre-condition saying that some other conflicting data attributes in a data composition should not already be in the possession of the user. Similar to that, the constraints on role assignment can be stated as a user should not be assigned a role if the role falls in the exclusive role set of a role the user has already be assigned to. As we are considering role assignment in distributed environment, we specify constraints on global roles.

$ExRole : Role \times Role \rightarrow \mathbb{B}$ is a truth-valued function that characterises the exclusive relation between two global roles.

Definition 20 (Exclusive Global Role) *denoted as $ExGR$, is the set of mutually exclusive global role pairs (gr_i, gr_j) $i \neq j$, that hold the relation $ExRole(gr_i, gr_j)$.*

The following constraint should be satisfied in global role authorisations.

Global Role Assignment Constraint

(e1) A user is authorised to a global role if the global role is not mutually exclusive with another role to which the user is already authorised.

$\forall\ gr_a, gr_b \in Role, ExRole(gr_a, gr_b) \Longrightarrow UserRole(gr_a) \cap UserRole(gr_b) = \phi$

In order to define constraints that indicate role assignment conflicts, we consider properties of separation of duty. To identify the transactions the separation of duty principle is applied on, the concept of *critical combination* is introduced.

Definition 21 (Critical Combination) *defines the set of transactions where each transaction should be performed by a different user.*

$CriCom : Transaction \rightarrow Trans$-set is a function that returns the critical combination of a given transaction.

A constraint on global roles restricting transactions according to the separation of duty principle is given as:

(e2) A global role has permission to a transaction only if this transaction is not in a critical combination with another transaction already allocated to this role.

$\forall\ gr \in Role, \forall\, trans_i\ \in Trans(gr),$

$trans = CT(gr) \Longrightarrow trans_i \notin CriCom(trans)$

6.6.3.3 Purpose Assignment Constraint

To handle conflicts, we can add a list of restrictions to the "access" and "purpose". For example, the two organisations, clinic and hospital, clinic is the sender and the hospital is the recipient, and the recipient should obey the restrictions that are imposed by clinic. Restrictions can be defined as "restrict the information obtained from my organisation (clinic or hospital) from being used for a specific sets of the following purposes and accesses".

The originator has the freedom to enforce restrictions that the recipient should adhere to. Here is the privacy policy rule for obtaining the patient's genetic information in the clinic with restrictions:

- Only the responsible GP is allowed to access the patient's genetic information for the purpose of "diagnosing" of certain disease. Other parties have been restricted from using the genetic data for any purpose.

When the hospital and the clinic initiate the conversation, their privacy policies are exchanged and the hospital imports all the "restricted access" and "restricted purpose" into its privacy policy rules. Now the policy rule of the hospital becomes:

- Given the referred patient's medical record received for the purpose of "certain disease diagnosing", the responsible doctor can access the genetic information provided that the patient information is not owned by clinic.

6.6.3.4 Information Flow Constraints

Illegal information flow happens because purposes of the objects involved in the information flow doesn't match each other. In order to prevent such illegal information flow, we should take into account the purposes of objects with the accesses of 'write', because that is the case when information 'flows'. In the example stated in Section 6.3.4, the receptionist can read medical record in medical history, which he has legal access. But he doesn't has access to medical record. The illegal information flow from medical record to medical history occurs because the intended purpose of medical record doesn't include the access purpose 'registration'. This kind of illegal information flow occurs in single organisational case. Suppose $obj1$ and $obj2$ reside in different organisations, the information flow may also occur. It shows that illegal information flow will happen in the case of multiple organisations as well.

Information flow constraint can be specified as:

Information Flow Constraints

(f1) In any state, if a conditional role has simultaneous read access to object $obj1$ and write or append access to object $obj2$, then:

$IP(PM)(Typeof(obj1)) \supseteq IP(PM)(Typeof(obj2))$

6.6.4 Proof Obligation

Model rules of a model specify operations by which the state variables can be changed. The precondition and the postcondition of model rules are used to specify the changes of status. In this section, an example of a formal specifications of model rule and its proof obligation showing that the operation is satisfiable, is given.

Rule 3: get-access

Conditional role cr requests that access object obj in mode x be enabled, $x \in \{read, write, append\}$. This is specified as following:

$get\text{-}access(cr : CR(RM), obj, x : Mode)$

ext rd $RM : RoleModel$ rd $PM : PurposeModel$

wr AM:$AccessModel$ rd $OM : ObjectDataModel$

pre $CCR(user) = cr \wedge \langle obj, ap\rangle = CReq(AM)(cr)$
$\wedge \langle ap, TypeOf(obj), CT(AM)(cr), x\rangle \in NA(AM)$
$\wedge\ ap \in IP(TypeOf(obj))$
$\wedge\ (\forall obj_i \in Accessed(user)(OM)\cdot$
$ExCompOf(obj) \cap ExCompOf(obj_j) = \phi)$

post $AM' = \langle Req(AM), CReq(AM), Trans(AM),$
$CT(AM), AT(AM), Mode(AM), NA(AM),$
$CA(AM) \cup \{\langle cr, obj, x\rangle\}\rangle$

The precondition states that a conditional role requests the access to object obj in current request with the access purpose ap, and the access is necessary. The postcondition then states that the access to obj is enabled for this role.

Then, symbols are defined to represent this operation's precondition and postcondition.

$pre\text{-}get\text{-}access(cr, obj, x, OM, RM, PM, AM) \stackrel{\text{def}}{=}$
$CCR(user) = cr \wedge \langle obj, ap\rangle = CReq(AM)(cr) \wedge$
$\langle ap, TypeOf(obj), CT(AM)(cr), x\rangle \in NA(AM)$
$\wedge\ ap \in IP(TypeOf(obj))$
$\wedge\ (\forall obj_i \in Accessed(user)(OM)\cdot$
$ExCompOf(obj) \cap ExCompOf(obj_i) = \phi)$

$post\text{-}get\text{-}access(cr, obj, x, AM, AM') \stackrel{\text{def}}{=}$

$AM' = \langle Req(AM), CReq(AM), Trans(AM),$
$CT(AM), AT(AM), Mode(AM), NA(AM),$
$CA(AM) \cup \{\langle cr, obj, x\rangle\}\rangle$

The satisfiability obligation associated with the operation $get\text{-}access$ is given as follows:

Proof Obligation 3: Operation $get\text{-}access$ is satisfiable.

$get\text{-}access\text{-}sat$
$OM : ObjectDataModel; PM : PurposeModel;$
$RM : RoleModel; AM : AccessModel; inv\text{-}DPPS;$
$pre\text{-}get\text{-}access(cr, obj, x, OM, RM, PM, AM)$

$\exists\, obj, x, AM : AccessModel, AM' : AccessModel$
$\cdot\, post\text{-}get\text{-}access(cr, obj, x, AM, AM') \wedge inv\text{-}DPPS'$

Next we prove this satisfiability obligation.

from $OM : ObjectDataModel; RM : RoleModel;$
$PM : PurposeModel; AM : AccessModel;$
$inv\text{-}DPPS; pre\text{-}get\text{-}access(cr, obj, x, OM,$
$RM, PM, AM)$

1 $\{obj\}$:$object(OM')$-set

2 x:$Mode(AM')$

3 $\langle cr, obj, x\rangle$:$CA(AM')$

4 from $ap \in IP(PM)(tp)$

4.1 $\langle ap, tp, CT(AM)(cr), x\rangle \in NA(AM)$

4.2 $\forall obj_i \in Accessed(user)(OM)\cdot$
$ExCompOf(obj) \cap ExCompOf(obj_i) = \phi)$

infer $CA(AM) = CA(AM) \cup \{\langle cr, obj, x\rangle\}$

5 from $inv\text{-}DPPS$

5.1 $\forall obj_i \in Accessed(user)(OM)\cdot$
$ExCompOf(obj) \cap ExCompOf(obj_i) = \phi)$

infer $\exists\, obj, x : Mode(AM)$
$AM : AccessModel, AM' : AccessModel$
$\cdot\, inv\text{-}DPPS'$

infer $\exists\, obj, x : Mode(AM),$
$AM : AccessModel, AM' : AccessModel$

$\cdot$ *post-get-access*(cr, obj, x, AM, AM')
$\wedge$ *inv-DPPS*$'$

This proof obligation states that, whenever the system is in some legal state of privacy preservation, and the operation's parameters satisfy its precondition in that state, there must always be at least one state configuration satisfying the operation's postcondition.

6.6.5 Access Control Mechanism

To implement exclusive composition in an access control system, we propose the access log mechanism. Through this mechanism, accesses to data objects in an exclusive composition will be logged. The allocation of access will be constrained over the course of time.

The exclusive compositions are defined in pre-conditions of the system attributes. Every time a user makes an access to a data object, the $\langle user, access, object \rangle$ tuple is logged. So for any data object, if there is a pre-condition stating that it should be accessed only if other data objects in conflict are not already in possession of the user. The policy then specifies this rule. With the presence of such a rule, the access control system then gets the access history of the user, checks it for any violations. Only then would it grant or deny access.

Data composition can be in a single organisation, when individual data objects in the data composition are in the same organisation. When the data owner wants his information units in different organisations to be protected under certain policies, data composition can be formed by data objects in multiple organisations. So defined policy rules applies to data composition from more than one organisations.

The authorisation process in the purpose based access control model is illustrated as follows. When a data user requests to access certain data object, access control is to be checked first and corresponding data policy, including data owner's preferences, are retrieved from privacy management system. Access control is checked based on conditional roles activated, including global and local roles, subjects invoked by those roles to access data, role-subject mapping. If the request passes the role check, then data purposes is to be checked against the access purpose. If access purpose is compliant with the intended purpose, then conditions are to be checked, including exclusive data composition of requested data object. If the request passes this procedure, then access is granted. Otherwise, access is denied. All data access requests should be logged in the audit trail.

6.7 Matching Privacy Policies in a Federation

In a collaborative distributed federation, federated organisations (FOs) can exchange personal information to automatically authorise data owners without asking them to submit the same information multiple times. Further, in a federated environment, FOs may need to access information to perform internal activities, such as evaluation of the health state of a patient or determination of patient eligibility to a given exam in a medicalcare environment. We also notice here that sharing information may provide important benefits to the participants themselves. For example, by sharing medical records of a patient, a physician may have available all information concerning the patient and therefore perform a more informed diagnosis.

To enable secure information sharing across FOs, we must assert that the privacy policies of all the FOs that receive information pertaining to a given individual comply with privacy preferences of this individual. In a federation system, a compliance check can be executed between two FOs when one FO (referred to as FO1) requests one or more personal information units from another FO (referred to as FO2). Instead of matching policies against data owner preferences, FO2 can more easily verify whether or not its policies subsume those of FO1. Subsumption reasoning is used on policies defined over equal or similar class of data in order to determine if they conflict.

To enhance flexibility and facilitate the task of policy specification of FOs, we consider two different ways of expressing privacy policies: using *policy templates* or specifying *customised policies*. It is assumed that some policy templates to be pre-defined and available for privacy policy specification. It is also assumed that privacy policy templates to be defined by the FOs as preliminary agreement of possible practices of the entities. A FO may choose to use one of the available templates or may specify its own customised privacy policies. Similarly, data owners can specify privacy requirements according to available specific pre-defined templates, or they can specify their own requirements.

6.7.1 Policy Templates

As aforementioned, FOs can simplify the task of policy specification by using policy templates. Each template has a predefined set of values and is standardised across the federation. Each FO can choose a template T_i from the set $\{T_1, \ldots, T_n\}$ of available templates. The templates in such set are totally ordered based on the strictness

Element	**Value**
Purpose	*current*
Access	*all*
Recipient	*ours*
Retention	*stated-purpose, legal-requirement*

Table 6.1: Exemplar Strict Policy

Element	**Value**
Purpose	*current, contact*
Access	*all*
Recipient	*ours, same*
Retention	*stated-purpose, legal-requirement*

Table 6.2: Exemplar Moderate Policy

approach that will be followed for data disclosure. Specifically, templates are in descending order, then T_i defines practices that are stricter than those defined by policy template T_k, if $k > i$. In other words, T_k subsumes T_i. To simplify the process of policy expression, templates can be used to specify privacy practices for object types, data compositions, attributes or user credentials. In order for information to be released between two FOs, the associated policies must be compliant. Here, by compliant policies we mean that if information is being released from FO2 to FO1, then privacy policy enforced by FO1 should be equal or stricter than the policy applied by FO2.

As suggested by [Lib03], an example of set of policy templates ordered according the strictness is: $\{$*Strict, Cautious, Moderate, Flexible, Casual*$\}$. Adopting the notation adopted by [Lib03] for the P3P syntax, we provide examples of such policy templates in Tables 6.1, 6.2, 6.3.

The policy illustrated in Table 6.1 is that the data may be used only for the current activity and cannot be shared with others. Element *Recipient* is set to *ours*, meaning that the owner has full access to data and *Retention* element states that data is kept for only as long as the purpose required or as mandated by law.

The policy template shown in Table 6.2 is a possible moderate policy. It states that the data it refers to may be used for this activity and can be shared with others having the same business practices. The *Purpose* element states that data owner can be contacted with suggestions concerning processing. As in the example of Table 6.1,

Element	Value
Purpose	*current, contact, other-purpose*
Access	*none*
Recipient	*ours, unrelated, other-recipient*
Retention	*indefinitely*

Table 6.3: Exemplar Casual Policy

data owner has full access to data. Data is kept only as long as purpose requires or according to the length mandated by law.

In Table 6.3 a template for a casual policy is given. Such policy is stated as: data may be used for any activity, as stated by the *Purpose* element. This data may be shared with any unrelated entity irrespective of their policies. Data owner can be contacted with suggestions concerning processing. Data owner may not be able to access or correct data. Finally, as specified by the *Retention* element, data may be kept indefinitely.

If both federated organisations use pre-defined policy templates, policy comparison is straightforward: pre-defined policy templates are totally sorted based on the requirements that need to be met in order to release data. Policy subsumption performs local matching from the perspective of a federated organisation FO1, which is servicing a request for an information unit *I* from another federated organisation FO2. Note that both organisations are using policy templates totally sorted in descending order, therefore, T_k subsumes T_i if $k > i$. Assume that templates $\{T_k, T_i\}$ represent $\{$*Pol1*, *Pol2*$\}$ respectfully, policy subsumption can be performed by checking whether $k > i$.

It is important to note that the definition of policy templates is to be agreed upon by federated organisations. When all entities in a federation use the policy template approach, it is simple to perform policy matching. However, policy templates inherently lack flexibility, and limit the range of preferences and intentions that data owners and FOs can express.

6.7.2 Customised Privacy Policies

Customised privacy policies are designed by FO or data owner and can arbitrarily create rules that describe how data will be managed. These policies give FOs or data owners a flexible and expressive method for defining their privacy preferences and

Element	**Value**
Purpose	*current, pseudo-analysis*
Access	*all*
Recipient	*ours*
Retention	*stated-purpose, legal-requirement*

Table 6.4: Exemplar Customised Policy

practices. However, customised policies are more difficult to specify, match and, typically, to enforce. Moreover, this flexibility increases the difficulty of policy matching. It is fair to assume that federated organisations may refer to similar terms with different names. In order to determine the relationship between two different terms while performing local matching, we assume that federated terminology is used.

The framework for performing local matching between customised FO policies is identical to the method described in Section 6.7.1. However, determining the relative policy strictness is a more articulated process. To evaluate the relationship between two given policies, Pol_r and Pol_s, associated respectively with the requester and the data holder, it is necessary to analyse the *purposes* and *recipients* for all data being requested from Pol_r. Therefore, every data element that is being requested by holder of Pol_r is evaluated to determine whether the requester's intended use of the data element is subsumed by those in Pol_s. Each *purpose* in Pol_r pertaining to this data element is then examined, checking if it is a subset of the purposes pertaining to the same data element from Pol_s. The same comparisons are then performed for the *retention* and *recipient* elements of the policies. Comparison of purposes, recipients, and retentions are based on the semantic hierarchical nature of our data composition. Finally, if the purposes, retentions, and recipients of the requesting policy are all subsumed by the servicing policy, a positive result is returned. Otherwise the result is negative.

An example of a possible customised policy is specified in Table 6.4, which can not be expressed following policy template approach. The policy states that data may be used only for this activity and cannot be shared with others. Statistical records may be kept only with identifying information removed. Data is kept only as long as purpose required or as mandated by law.

6.8 Chapter Summary

This chapter described our design considerations for privacy protection in distributed collaborative environments. Two design considerations were provided. The first one is to ensure that personal information is accessed from two or more parties only if agreed privacy policies and privacy preferences are satisfied. The second consideration is to facilitate privacy policies matching and privacy preference compliance.

The entities of the model were analysed according to the requirements in distributed environment. The purpose based access control approach is revised for distributed environment. With these analyses, data composition is proposed, and privacy policies matching method is also provided.

Chapter 7

Conclusions

This thesis presents the purpose-based access control method, which was designed to improve requirements analysis for the development of privacy protection approach. This work was motivated by the fact that access control policy specification was without systematic procedural support, resulting in systems that are vulnerable to privacy breaches. Additionally, policy specification was isolated from requirements analysis for privacy protection approach design. This could lead to access policies that are not in compliance with privacy requirements. The purpose-based access control approach integrates policy specification into privacy protection approaches. It provides prescriptive procedural guidance and support for specifying access control policies from privacy requirements.

The research reported in this thesis was developed while performing analysis on privacy protection approaches. The privacy protection approach was developed by integrating policy specification and requirements analysis. Through this process, a purpose based access control method was designed and specified. This approach is also used for the analysis of design considerations for privacy preservation in distributed collaborative environments.

The chapter is organised as follows: Section 7.1 provides a synopsis of each chapter; Section 7.2 summarises the contributions of this work; Section 7.3 discusses plans for future work; and Section 7.4 concludes the thesis.

7.1 Chapter Synopsis

Chapter 1 introduced and articulated the problem addressed in this work. Specifying complete and correct policies that control users' access to a system and its resource

is important for the protection of data privacy in information system. Traditionally, the access control policies specification lacked systematic support and was isolated from requirements analysis, resulting in access policies that are not compliant with privacy requirements. This misalignment between privacy requirements and practices motivated the research presented in this thesis, a systematic method for specifying privacy requirements for information system.

Chapter 2 provided an overview of related work in privacy preservation, and Chapter 3 presented privacy protection approach specifications. This positioned the work presented in this thesis. It illustrated the motivations for integrating the privacy protection approach and specification, and argued the necessity for purpose based access control approach. The purpose based access control approach builds upon this prior work by integrating policy specification and requirements analysis, and integrating the concept of purpose into privacy protection approach.

Chapter 4 detailed the purpose based access control method. The entities, the relationships, and privacy requirements in a single organisation were presented using VDM. It outlined the basic framework of our privacy preserving access control model, and presented essential concepts, definitions of the main entities, and formal specifications of mapping functions and access granting rules. To complete the access control method, specifications of operations and their proof obligations have also been investigated.

Chapter 5 illustrated the use of formal specification in the development of a privacy preserving personal information management system in a medical care scenario. The basic problem analysis was provided in that the scenario is simplified and part of personal data processing was specified. It provided a good indication of how formal specification techniques could be integrated into traditional requirements analysis approach, and showed how an initial specification can be formed and then manipulated in a rigorous way through the careful introduction of design detail in the form of data structure and operations.

Chapter 6 analysed design considerations for privacy preservation in distributed collaborative environments. Two design considerations are presented: (1) data composition was proposed to ensure that personal information is accessed from two or more parties only if agreed privacy policies and privacy preferences are satisfied, and (2) policy subsumption mechanism was used to facilitate privacy policies matching and privacy preference compliance among distributed collaborative organisations.

7.2 Summary of Major Results

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

7.2.1 Identification of Purpose Based Access Control

An important contribution of the thesis is the identification of the limitations of existing approaches in supporting data owners to achieve better privacy preservation in data sharing computing environments and our proposal of purpose based access control method. In particular, the thesis presented the following results concerning the identification of purpose based access control method for privacy preservation:

- Investigated a number of approaches to privacy preservation that have taken access control mechanism for privacy preservation, and showed that these approaches failed to enable users to efficiently and effectively adjust the level of openness according to their changing desire for privacy in different situations.

- Studied work on privacy policies for privacy preservation, demonstrated that privacy preservation is not about setting rules and enforcing them but rather should also taking data owners' preferences into account, and showed that existing privacy preserving approaches failed to express privacy requirements from data owner's perspective.

- Combined the analysis on general privacy preserving guidelines and privacy policies to demonstrate that privacy requirements concern more about the purposes that a data object is used for, rather than the actions that users perform on the data object, so the notion of purpose should play a major role in access control methods for privacy preservation.

- Motivated the need for purpose based access control method for privacy preservation and defined the desired end result of privacy preservation as "to ensure that a data object is used only for its intended usage, the access purpose should be compliant with the data object's intended purpose".

7.2.2 Purpose Based Access Control Method for Privacy Preservation

The second major contribution of this work is the introduction of the purpose based access control method for privacy preservation in a intra-organisational case, which is comprised of:

- a process description that details the steps for privacy requirements analysis;
- a complete model that details entities, relationships, and privacy requirements;
- a specification mechanism for privacy property analysis and proof obligations.

Privacy protection specification is typically isolated from requirements analysis. It is often conducted without methodological support or systematic guidance. The purpose based access control method and its specification offers two main advantages not currently available:

- integrating specification with requirements analysis for privacy protection approach design;
- specification mechanism for privacy property analysis and verification.

The development scheme introduced in the thesis that integrates policy specification with requirements analysis is significant. By integrating policy specification with privacy protection approach, purpose based access control method provides a basic framework for ensuring compliance between privacy requirements and data processing practices. The impact of this compliance is significant. One of the problems that plague organisations is the degree of confidence they have in claiming that their information systems are enforcing privacy policies. This problem also hampers the acceptance of information technology due to the lack of trust from data owners on information processing organisations.

The purpose based access control method is a promising step in the right direction towards gaining data owners' trust and improving their confidence in the disclosure of their personal information. The results concerning the purpose based access control method consist of:

- First, access control policies were derived from basic privacy requirements, high level security and privacy policies, and data owners' preferences on the usage

of their private information. Because privacy requirements come from these sources, this development scheme helps ensure that a data processing system is actually enforcing privacy policies and privacy preferences.

- Second, the entities in the purpose-based access control model, the invariants corresponding to the privacy requirements, and the model operations and their proof obligations were specified. The requirements analysis and specification is an iterative process. We derive specification from requirements analysis and approach design, and by clarifying ambiguities in the requirements and resolving inconsistencies between the requirements and data processing practices, we also improve requirements and design during privacy protection specification,

7.2.3 Design Considerations for Privacy Preservation in Distributed Collaborative Environments

In this thesis we analysed the further challenges posed by distributed computing environments on privacy preservation, including information of an individual spanning over several organisations and thus governed by different privacy policies. The work reported in this thesis presented the design considerations for privacy preservation in distributed collaborative environments. The detailed results concerning design considerations in distributed collaborative environments are:

- Data composition was proposed to overcome the limitations of object type in the purpose based access control approach, ensuring that personal information is accessed from two or more parties only if agreed privacy policies and privacy preferences are satisfied. When a data owner's information spans more than one organisation, simply restricting accesses to data objects by separate policies in different organisations and returning results on policy satisfaction, cannot guarantee the prevention of accesses to certain combinations of data objects. Exclusive compositions were defined as data compositions that hold data objects exclusive to each other, and were used to prevent accesses to certain data object combinations.

- Policy subsumption mechanism was proposed for facilitating privacy policies matching and privacy preference compliance among distributed collaborative organisations. To enable information sharing across collaborative organisations, the privacy policies of all organisations that receive information pertaining to a

given individual should comply with privacy preferences of the individual. The checking mechanisms for privacy policies matching and privacy preference compliance were defined for privacy requirements expressed using policy templates or customised policies.

7.3 Future Work

The work presented in this thesis addresses some of the fundamental problems with privacy protection approach specification; however, our approach does have its limitations, and work remains to be done in these areas. This section discusses some of the main limitations of the purpose based access control approach and provides an overview of areas of future interest and work.

7.3.1 Improving the Purpose Based Access Control Method

The main limitations of the purpose based access control method are:

- obligation has not been investigated in privacy protection approach;
- the method does not provide much support for role assignments for distributed environment; and
- formal analysis of the policy subsumption mechanism is not available at this time.

The first area for future work involves extensions to the purpose based access control method. Plans for extending purpose based access control method can be summarised as pursuing the following directions:

- investigate other entities for privacy protection approach, such as obligation;
- analyse design consideration on role assignments for distributed environment; and
- provide formal analysis to policy subsumption mechanism to support automatic reasoning.

In this thesis, we found role attributes and system attributes were sufficient to represent the conditions in our analysis. However, we have yet to investigate the other entity for privacy protection: obligation. Our plans for extending privacy protection approach include investigating this entity.

Role-based access control (RBAC) is widely used in many applications to simplify authorisation management. Defining roles for a complex organisation can be very complex. Although purpose based access control approach supports role assignment management in intra-organisational case, it does not provide much support on role assignments in distributed environments. Role definition and management should be part of the purpose based access control method for distributed privacy protection. A structured role assignment management is needed.

The work presented in this thesis focuses on the identification and specification of privacy requirements for information systems. The support for automatic reasoning of policy matching and preference compliance is still limited. For example, the policy subsumption mechanisms can only match template policies. Additionally, to evaluate whether distributed privacy protection approach achieves its goal, we need formal analysis on the whole approach, including policy matching. Our plan in this direction is to extend the expressibility of privacy policies, and develop formal algorithms for automatic policy matching.

The above discussions are aimed at extending the purpose based access control method to enrich the method and make it more useful. Broader areas of future work are also under consideration.

7.3.2 Using the Method in Legacy systems

The purpose based access control method can be used during the development of new data processing systems. However, there are many legacy systems in place in most organisations and this raises additional questions that must be addressed: Can the purpose based access control method be used as a checking method to check whether a legacy system is enforcing the high-level security and privacy policies? Can the purpose based access control approach be used to check whether two legacy systems with different high-level policies are in compliance with one another? Answers to these questions have practical significance. For example, when an privacy act was introduced, organisations being regulated wanted to know whether their legacy systems were in compliance with the new regulation. In another example, if two organisations

merge, and both organisations have legacy systems in place that are enforcing different policies, can the purpose based access control method help them ensure that their legacy systems are in compliance with one another? We believe that the subsumption mechanism support offered by the design considerations for privacy preservation in distributed collaborative environments is a promising solution on this matter.

7.4 Concluding Remarks

Privacy has become a growing concern in data sharing computing environments. People selectively disclose private information to get certain services, while at the same time they want to remain in control of their privacy. Data processing organisations not only need to follow general guidelines for privacy preservation, they also need to take into account privacy requirements from data owners' points of view.

Privacy protection approach specification is a complex process and involves extensive requirements analysis. In this thesis, we presented the purpose based access control method that provides procedural guidance for this purpose. The purpose based access control model can be used to implement a privacy-enhancing access control system, which integrates privacy requirements from data owner's point of view, and enforces basic privacy requirements. Additionally, the use of specification in privacy protection approach design improves the requirements analysis and the compliance between privacy requirements and data practices by clarifying ambiguities in the requirements. Finally, the design considerations for privacy protection in distributed collaborative organisations provides a foundation for ongoing research on privacy protection approaches for distributed environments.

Appendix A

Access Control with Exclusive Data Compositions

Using exclusive data composition to control access to private information items is well integrated with purpose based access control method. To analyse exclusive data composition, we illustrate the grouping of private information based on information categories described below as defined in Section 3.2 of [Wor07]:

Physical Contact Information ⟨**physical**/⟩: Information such as telephone number or a delivery address that makes it possible for an individual to be contacted or physically located.

Online Contact Information ⟨**online**/⟩: Information such as email address or home pages that allows an individual to be contacted or located on the Internet.

Unique Identifiers: ⟨**uniqueid**/⟩ Non-financial and non-government identifiers, generally issued by a Web site or service in order to uniquely and consistently identify or recognise the individual.

Purchase Information ⟨**purchase**/⟩: Information actively generated through the purchase of a product or service, including information about the method of payment.

Financial Information ⟨**financial**/⟩: Information items about an individual's finance situation, including his account status and activity information, such as account balance, payment or overdraft history, and information about an individual's purchase or use of financial instruments, including credit or debit card information.

Computer Information ⟨**computer**/⟩: Information items such as the IP address, domain name, browser type or operating system that relating to the computer system that is being used to access the network by the individual .

Navigation and Click-stream Data ⟨**navigation**/⟩: Information such as the referrer

page and how long users stay on each page that is passively generated by browsing the Web site.

Interactive Data ⟨**interactive**/⟩: Data such as server logs of account activity or queries to a search engine that is actively generated from or reflecting past interactions with the target service provider through its site.

Demographic and Socioeconomic Data ⟨**demographic**/⟩: Data about an individual's characteristics, such as gender, age, income, postal code, or geographic region

Content: ⟨**content**/⟩ Information, such as the body of an email, bulletin board posts, or IRC chat transcript that is contained in the body of a communication.

State Management Mechanisms ⟨**state**/⟩: Information such as HTTP cookies used for maintaining a stateful session with a user or automatically recognising users who have visited a particular site or accessed particular content previously.

Political Information ⟨**political**/⟩: Information about a user's religious and political linkup, such as membership in or affiliation with groups such as religious organisations, trade unions, professional associations, or political parties.

Health Information ⟨**health**/⟩: Information about an individual's state of health, including physical or mental health, use or inquiry into health care services or products, or purchase of health care services or products.

Preference Data ⟨**preference**/⟩: Information about a user's individual taste, such as favorite color, musical tastes or hobbies.

Location Data ⟨**location**/⟩: Information such as GPS position data that can be used to determine an individual's current physical location and track them as their location changes.

Government-issued Identifiers ⟨**government**/⟩: Identifiers such as a social security or passport number issued by a government to consistently identify the individual.

Other ⟨**other-category**/⟩: Other types of data not covered by the above categories.

It distinguished the **Computer**, **Navigation**, **Interactive** and **Content** categories. The Computer category includes information about the user's computer including IP address and software configuration, while Navigation data describes actual user behavior related to browsing. When an IP address is stored with information related to browsing activity, both the Computer and the Navigation category should be used. Interactive Data is data actively solicited to provide some useful services at a site beyond browsing. Content is information exchanged on a site for the purposes of communication.

The categories described in P3P give data owners and their user agents additional

Categories	Description	Exclusive Composition
physical/	Physical Contact Information	ExComp11 ExComp12 ExComp13
online/	Online Contact Information	ExComp21 ExComp22 ExComp23
uniqueid/	Unique Identifiers	ExComp31 ExComp32 ExComp33
purchase/	Purchase Information	ExComp01
financial/	Financial Information	ExComp01
computer/	Computer Information	Allow01
navigation/	Navigation and Clickstream Data	Allow02
interactive/	Interactive Data	Allow03
demographic/	Demegraphics and Socioeconomic Data	ExComp11 ExComp21 ExComp31
content/	Content	N/A
state/	State Management Mechanisms	N/A
political/	Political Information	ExComp12 ExComp22 ExComp32
health/	Health Information	ExComp13 ExComp23 Excomp33
preference/	Preference Data	N/A
location/	Location Data	N/A
government/	Government-issued Identifiers	N/A
other-category/	Other	N/A

Table A.1: An Exemplar Definition of Exclusive Data Compositions

clues as to what type of information is requested from a service. When forming data compositions using the defined categories, it is important to determine what exclusive data compositions these groups will be divided into. As the comfort levels of privacy preserving vary from one person to another, decisions on this would be ideally left to the data owners. An example of such a partition is given in Table A.1. In this example, the data owner is willing to let the organisation collect computer information, navigation and click-stream data, and interactive data regardless of which other categories of information is required, so these categories are placed into their own exclusive data composition separately. The data owner would not want any organisation to access both his purchase information and financial information, but access to any one of these categories is allowed, therefore his financial information and purchase information are placed in the same exclusive data composition. Health information, political information, and demographics and socioeconomic data can be collected by an organisation, as long as he can not be identified. If an organisation can identify the data owner, then it is allowed to have his physical and online contact information, as well as unique identifiers. To achieve this, exclusive data composition ExComp11 through to ExComp33 are defined. These exclusive data compositions are added to the health, political and demographics categories to prevent simultaneous access to this information with identifying information. The partition defined above is one possibility, different individual may have a similar but possibly different decision.

The data composition consisting of exclusive data objects are defined in the pre-conditions of the system attributes. Every time a user makes an access to a data object, the $\langle user, access, target \rangle$ tuple is logged. So if any data object has a pre-condition saying that it should be accessed only if some other data objects in conflict are not already in the possession of the user, the policy would specify this rule. With the presence of such a rule, the access control system would get the access history of the user and check it for any violations and only then would it grant or deny access. For example, the rules for the partitioning above are as follows:

1. Deny access if both physical/ and demographic/ are required
2. Deny access if both physical/ and health/ are required
3. Deny access if both physical/ and political/ are required
4. Deny access if both online/ and demographic/ are required
5. Deny access if both online/ and health/ are required

6. Deny access if both online/ and political/ are required
7. Deny access if both uniqueid/ and demographic/ are required
8. Deny access if both uniqueid/ and health/ are required
9. Deny access if both uniqueid/ and political/ are required

The following exemplar policy rules would demonstrate the implementation of this type of control in more details. Consider there are three data objects $name$, $address$, and phone $number$, suppose that the data owner doesn't want the pair ($name$, $number$) to be held by any organisation at the same time, then this is written into:
Precondition: exclusive data composition ($name$, $number$)
Policy rule for data object $name$:
if (user: *A*, previous action: *access*, target: *number*)
because
$ExCompOf(number) \cap ExCompOf(name) = \{(name, number)\}$,
then
disallow(user: *A*, action: *access*, target: *name*)
else
allow(user: *A*, action: *access*, target: *name*)
Policy rule for data object $number$:
if (user *A*, previous action: *access*, target: *name*)
because
$ExCompOf(name) \cap ExCompOf(number) = \{(name, number)\}$,
then
disallow(user: *A*, action: *access*, target: *number*)
else
allow(user: *A*, action: *access*, target: *number*)

The condition part in the above implementation may be a composite consisting of conditions that specify the exclusive data compositions for a particular data object.

The data composition can happen in a single organisation, when individual data objects or object types in a data composition are in the same organisation, and the data composition can also be formed by data objects in multiple organisations, when the data owner wants his information units in different organisations to be protected under certain policies. So the discussion above can be naturally extended to distributed scenarios.

Bibliography

[AB08] Nabil Ajam and Ahmed Bouabdallah. Privacy improvement through pseudonymity in parlay x for location based services. *International Conference on Networking*, 0:713–718, 2008.

[ABLY07] Annie I. Antón, Elisa Bertino, Ninghui Li, and Ting Yu. A roadmap for comprehensive online privacy policy management. *Communications of the ACM*, 50(7):109–116, 2007.

[Acq04] Alessandro Acquisti. Privacy in electronic commerce and the economics of immediate gratification. In *Proceedings of the 5th ACM conference on Electronic commerce (EC-04)*, pages 21–29, New York, 2004. ACM Press.

[ACR99] Mark Ackerman, Lorie F. Cranor, and Joseph Reagle. Privacy in E-Commerce: Examining user scenarios and privacy preferences. In *Proceedings of the ACM Conference on Electronic Commerce (EC-99)*, pages 1–8, New York, November 1999. ACM Press.

[AF08] Bandar S. Alhaqbani and Colin J. Fidge. Privacy-preserving electronic health record linkage using pseudonym identifiers. In *10th International Conference on e-health Networking, Applications and Services*, pages 108–117, 2008.

[AG97] Martín Abadi and Andrew D. Gordon. A calculus for cryptographic protocols: The Spi calculus. In *Fourth ACM Conference on Computer and Communications Security*, pages 36–47, 1997.

[AHK+03] Paul Ashley, Satoshi Hada, Günter Karjoth, Calvin Powers, and Matthias Schunter. Enterprise privacy authorization language (EPAL 1.2). Technical report, IBM, 2003.

[AHKS02] Paul Ashley, Satoshi Hada, Günter Karjoth, and Matthias Schunter. E-P3P privacy policies and privacy authorizations. In *Proceeding of the ACM workshop on Privacy in the Electronic Society (WPES-02)*, pages 103–109, New York, November 2002. ACM Press.

[AK05] Ali E. Abdallah and Etienne J. Khayat. A formal model for parameterized role-based access control. In *Formal Aspects in Security and Trust*, IFIP International Federation for Information Processing, pages 233–246. Springer, 2005.

[And00] Ross J. Anderson. Privacy technology lessons from healthcare. In *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, pages 78–79. IEEE Computer Society, 2000.

[And04] Ross J. Anderson. Personal information, privacy and ubicomp. In *2nd UK-UbiNet Workshop: Security, trust, privacy and theory for ubiquitous computing*, University of Cambridge, May 2004.

[APS02] Paul Ashley, Calvin Powers, and Matthias Schunter. From privacy promises to privacy management: A new approach for enforcing privacy throughout an enterprise. In *Proceedings of the New Security Paradigms Workshop (NSPW)*, pages 43–50. ACM Press, 2002.

[AS00] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, volume 29 of *SIGMOD Record (ACM Special Interest Group on Management of Data)*, pages 439–450. ACM Press, 2000.

[Bal90] Robert W. Baldwin. Naming and grouping privileges to simplify security management in large databases. In *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy*, pages 116–132, 1990.

[Bar97] Barry Barber. Security and confidentiality issues from a national perspective. In *Patient privacy, confidentiality and data security. Papers from the British Computer Society Nursing Specialist Group Annual Conference 1995*, volume 3 of *INFOrmed Touch*, 1997.

[BB89] Joachim Biskup and Hans H. Bruggemann. The personal model of data towards a privacy oriented information system. In *Proceedings of 5th IEEE International Conference on Data Engineering*, pages 348–355, February 1989.

[BBB09] BBBOnline, 2009. Retrieved on 16/10/2010, from http://www.bbb.org/online/.

[BBC07] BBC News. Uk's families put on fraud alert, 2007. Retrieved on 16/10/2010, from http://news.bbc.co.uk/1/hi/uk_politics/7103566.stm.

[BBC08] BBC News. Hsbc loses customers' data disc, 2008. Retrieved on 16/10/2010, from http://news.bbc.co.uk/1/hi/business/7334249.stm.

[BBC10] BBC News. Data on 9,000 school children stolen in burglary, April 2010. Retrieved on 16/10/2010, from http://news.bbc.co.uk/1/hi/england/london/8601769.stm.

[BBL05] Ji-Won Byun, Elisa Bertino, and Ninghui Li. Purpose based access control of complex data for privacy protection. In *Proceedings of the 10th ACM Symposium on Access Control Models and Technologies (SACMAT 05)*, pages 102–110. ACM Press, 2005.

[BCKM05] Simon Byers, Lorrie Cranor, David Kormann, and Patrick McDaniel. Searching for privacy: Design and implementation of a P3P-enabled search engine. In *2004 International Workshop on Privacy Enhancing Technologies (PET2004)*, volume 3424 of *Lecture Notes in Computer Science*, pages 314–328. Springer Berlin, 2005.

[BDMN06] Adam Barth, Anupam Datta, John C. Mitchell, and Helen Nissenbaum. Privacy and contextual integrity: Framework and applications. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 184–198. IEEE Computer Society Press, May 2006.

[BDVS01] Piero A. Bonatti, Ernesto Damiani, Sabrina D. Vimercati, and Pierangela Samarati. An access control model for data archives. In *Proceedings of the 16th international conference on Information security: Trusted information: the new decade challenge*, volume 193 of *International Federation For Information Conference Proceedings*, pages 261–276. Kluwer, 2001.

[Ben99] Paola Benassi. TRUSTe: An online privacy seal program. *Communications of the ACM*, 42(2):56–59, 1999.

[Ber03] Yolanta Beresnevichiene. A role and context based security model. Technical Report UCAM-CL-TR-558, University of Cambridge, January 2003.

[BFL+94] Juan Bicarregui, John Fitzgerald, Peter Lindsay, Richard Moore, and Brian Ritchie. *Proof in VDM: A Practitioner's Guide*. Springer-Verlag, 1994.

[BFTs04] Ron Berman, Amos Fiat, and Amnon Ta-shma. Provable unlinkability against traffic analysis. In *Proceedings of the 8th International Conference on Financial Cryptography*, Lecture Notes in Computer Science, pages 266–280. Spring-Verlag, 2004.

[BGA06] Jaijit Bhattacharya, S. K. Gupta, and Bhurvi Agrawal. Protecting privacy of health information through privacy broker. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, volume 5. IEEE Computer Society, 2006.

[BHAS04] Davide Bolchini, Qingfeng He, Annie I. Antón, and William H. Stufflebeam. 'I need it now': Improving website usability by contextualizing privacy policies. In *Proceedings of the 4th International Conference on Web Engineering*, volume 3140 of *Lecture Notes in Computer Science*, pages 31–44. Springer, 2004.

[BKP04] Claus Boyens, Ramayya Krishnan, and Rema Padman. On privacy-preserving access to distributed heterogeneous healthcare information. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04)*, volume 6, page 60135.1. IEEE Computer Society, 2004.

[BL08] Ji-Won Byun and Ninghui Li. Purpose based access control for privacy protection in relational database systems. *The VLDB Journal*, 17(4):603–619, 2008.

[BLM01] Jean Bacon, Michael Lloyd, and Ken Moody. Translating role-based access control policy within context. In *Proceedings of the International*

Workshop on Policies for Distributed Systems and Networks, volume 1995 of *Lecture Notes in Computer Science*, pages 107–119. Springer, 2001.

[BLPT04] Michael Butler, Michael Leuschel, Stephane Lo Presti, and Phillip Turner. The use of formal methods in the analysis of trust (position paper). In *iTrust*, Lecture Notes in Computer Science, pages 333–339. Springer, 2004.

[BM05] Adam Barth and John C. Mitchell. Enterprise privacy promises and enforcement. In *Proceedings of the 2005 Workshop on Issues in the Theory of Security*, pages 58–66. ACM Press, 2005.

[BN89] David F. Brewer and Michael J. Nash. The chinese wall security policy. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 206–214. IEEE Computer Society Press, 1989.

[BP76] David E. Bell and Leonard J. La Padula. Secure computer systems: Unified exposition and multics interpretation. Technical Report MTR-2997, MITRE Corporation, 1976.

[CAG02] Lorrie Cranor, Manjula Arjula, and Praveen Guduru. Use of a P3P user agent by early adopters. In *Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society*, pages 1–10. ACM, 2002.

[Cal03] Michael Caloyannides. Privacy vs. information technology. *IEEE Security and Privacy*, 1(1):100–103, 2003.

[CAMN+02] Roy H. Campbell, Jalal Al-Muhtadi, Prasad Naldurg, Geetanjali Sampemane, and M. Dennis Mickunas. Towards security and privacy for pervasive computing. In Mitsuhiro Okada, Benjamin C. Pierce, Andre Scedrov, Hideyuki Tokuda, and Akinori Yonezawa, editors, *Software Security ł Theories and Systems*, volume 2609 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2002.

[Cha81] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[Cha92] David Chaum. Achieving electronic privacy. *Scientific American*, 267(2):96–101, 1992.

[Cha01] Ramaswamy Chandramouli. A framework for multiple authorization types in a healthcare application system. In *17th Annual Computer Security Applications Conference (ACSAC'01)*, pages 137–148. IEEE Computer Society, 2001.

[CLM02a] Lorrie Cranor, Marc Langheinrich, and Massimo Marchiori. A P3P preference exchange language 1.0 (APPEL1.0). World Wide Web Consortium, Working Draft WD-P3P-preferences-20020415, 2002. Retrieved on 16/10/2010, from http://www.w3.org/TR/2002/WD-P3P-preferences-20020415.

[CLM+02b] Lorrie Cranor, Marc Langheinrich, Massimo Marchiori, Martin Presler-Marshall, and Joseph Reagle. The platform for privacy preferences 1.0 (P3P1.0) specification. World Wide Web Consortium Recommendation, April 2002. Retrieved on 16/10/2010, from http://www.w3.org/TR/2002/REC-P3P-20020416/.

[CM03] Frédéric Cuppens and Alexandre Miège. Modelling contexts in the orBAC model. In *19th Annual Computer Security Applications Conference (ACSAC 03)*, pages 416–427. IEEE Computer Society, 2003.

[CW+96] Edmund M. Clarke, Jeannette M. Wing, et al. Formal methods: State of the art and future directions. *ACM Computing Surveys*, 28(4):626–643, 1996.

[Dam02] Nicodemos C. Damianou. *A Policy Framework for Management of Distributed Systems*. PhD thesis, University of London, 2002.

[Dep09] Department of Justice of Canada. Using the access to information act and privacy act, 2009. Retrieved on 16/10/2010, from http://www.justice.gc.ca/eng/pi/atip-aiprp/use-uti.html.

[DFvL91] Anne Dardenne, Stephen Fickas, and Axel van Lamsweerde. Goal-directed concept acquisition in requirements elicitation. In *Proceedings of the 6th International Workshop on Software Specification and Design*, pages 14–21. IEEE Computer Society Press, 1991.

[DHTK93] Steven A. Demurjian, M.-Y. Hu, T. C. Ting, and D. Kleinman. Towards an authorization mechanism for user-role based security in an object-oriented design model. In *Proceedings of the 12th Annual International*

Phoenix Conference on Computers and Communications, pages 195–202. IEEE Computer Society Press, 1993.

[DS99] Ian Denley and Simon W. Smith. Privacy in clinical information systems in secondary care. In *British Medical Journal*, volume 318, pages 1328–1331, 1999.

[eBa10] eBay Inc. Privacy policy of ebay, 2010. Retrieved on 16/10/2010, from http://pages.ebay.com/help/policies/privacy-policy.html.

[EKS02] Matthias Enzmann, Thomas Kunz, and Markus Schneider. Privacy protection through unlinkability of customer activities in business processes using mobile agents. In *Proceedings of the 3rd International Conference on E-Commerce and Web Technologies*, volume 2455 of *Lecture Notes in Computer Science*, pages 314–323. Springer, 2002.

[Ele08] Electronic Privacy Information Centre. Public opinion on privacy, September 2008. Retrieved on 16/10/2010, from http://epic.org/privacy/survey/.

[FBD99] David Ferraiolo, John Barkley, and D. Richard Kuhn. A role-based access control model and reference implementation within a corporate intranet. *ACM Transactions on Information and System Security*, 2:34–64, 1999.

[FCD95] David Ferraiolo, Janet Cugini, and D. Richard Kuhn. Role based access control (RBAC): Features and motivations. In *Proceedings of the Annual Computer Security Applications Conference*. IEEE Computer Society Press, 1995.

[Fed98a] Federal Trade Commission (FTC). Children's online privacy protection act, 1998. Retrieved on 16/10/2010, from http://www.ftc.gov/ogc/coppa1.htm.

[Fed98b] Federal Trade Commission (FTC). The gramm-leach bliley act, 1998. Retrieved on 16/10/2010, from http://www.ftc.gov/privacy/privacyinitiatives/glbact.html.

[Fed98c] Federal Trade Commission (FTC). Privacy online: A report to congress, June 1998. Retrieved on 16/10/2010, from http://www.ftc.gov/reports/privacy3/index.htm.

[FH01] Simone Fischer-Hübner. *IT-Security and Privacy - Design and Use of Privacy-Enhancing Security Mechanisms*, volume 1958 of *Lecture Notes in Computer Science*. Springer Berlin, 2001.

[FHO98] Simone Fischer-Hübner and Amon Ott. From a formal privacy model to its implementation. In *Proceedings of the 21st NIST-NISSC National Information Systems Security Conference*, pages 512–524, 1998.

[FHS96] Simone Fischer-Hübner and Kathrin Schier. Risks on the way to the global information society. In *Information Systems Security: Facing the information society of the 21st century*, pages 487–488. Chapman & Hall, 1996.

[FK92] David Ferraiolo and Rick Kuhn. Role-based access controls. In *Proceedings of the 15th NIST-NCSC National Computer Security Conference*, pages 554–563, 1992.

[Fre09] Alexandra Frean. Security flaws halt work on contact-point child database, 2009. Retrieved on 16/10/2010, from http://women.timesonline.co.uk/tol/life_and_style/women/families/article5962974.ece.

[FWBBP06] Elke Franz, Hagen Wahrig, Alexander Boettcher, and Katrin Borcea-Pfitzmann. Access control in a privacy-aware elearning environment. In *Proceedings of the First International Conference on Availability, Reliability and Security*, pages 879–886. IEEE Computer Society, 2006.

[GGK+99] Eran Gabber, Phillip B. Gibbons, David M. Kristol, Yossi Matias, and Alain Mayer. On secure and psuedonymous client-relationships with multiple servers. *ACM Transactions on Information and System Security*, 2(4):390–415, 1999.

[GI96] Luigi Giuri and Pietro Iglio. A formal model for role based access control with constraints. In *Proceedings of the 9th IEEE workshop on Computer Security Foundations*, pages 136–145. IEEE Computer Society, 1996.

[GL04] Marco Gruteser and Xuan Liu. Protecting privacy in continuous location-tracking applications. *IEEE Security and Privacy*, 2(2):28–34, 2004.

[Goo09] Google Inc. Google privacy policy, March 2009. Retrieved on 16/10/2010, from http://www.google.com/intl/en/privacypolicy.html.

[HAF05] Bernardo A. Huberman, Eytan Adar, and Leslie R. Fine. Valuating privacy. *IEEE Security and Privacy*, 3(5):22–25, 2005.

[HB98] Ronald Hes and John Borking. Privacy enhancing technologies: the path to anonymity (revised edition). The Dutch Data Protection Authority, September 1998.

[He03] Qingfeng He. Privacy enforcement with an extended role-based access control model. Technical Report TR-2003-09, Department of Computer Science, North Carolina State University, 2003.

[Hop08] Christopher Hope. Ministry of defence says up to two million records of potential recruits are missing, October 2008. Retrieved on 16/10/2010, from http://www.telegraph.co.uk/news/newstopics/politics/defence/3190124/Ministry-of-Defence-says-up-to-two-million-records-of-potential-recruits-are-missing.html.

[HS04] Dominic Hughes and Vitaly Shmatikov. Information hiding, anonymity and privacy: a modular approach. *Journal of Computer Security*, 12(1):3–36, 2004.

[Hun05] Patrick C. Hung. Towards a privacy access control model for e-healthcare services. In *Proceedings of the 3rd Annual Conference on Privacy, Security and Trust*, 2005.

[Inf08] Information Commissioner's Office. Personal information survey, February 2008. Retrieved on 16/10/2010, from http://www.ico.gov.uk/upload/documents/library/data_protection/detailed_specialist_guides/icm_research_into_personal_information_feb08.pdf.

[Jac02] Daniel Jackson. Alloy: a lightweight object modelling notation. *ACM Transactions on Software Engineering and Methodology*, 11(2):256–290, 2002.

[Jon87] Cliff B. Jones. VDM proof obligations and their justification. In *Proceedings of the VDM-Europe Symposium on VDM - A Formal Method at Work*, volume 252 of *Lecture Notes in Computer Science*, pages 260–286. Springer, 1987.

[Jon90] Cliff B. Jones. *Systematic Software Development using VDM (second edition)*. Prentice Hall, 1990.

[JSS97] Sushil Jajodia, Pierangela Samarati, and V. S. Subrahmanian. A logical language for expressing authorizations. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 31–42. IEEE Computer Society Press, 1997.

[JSSS01] Sushil Jajodia, Pierangela Samarati, Maria L. Sapino, and V. S. Subrahmanian. Flexible support for multiple access control policies. *ACM Transactions on Database Systems*, 26(2):214–260, 2001.

[Kes92] Volker Kessler. On the chinese wall model. In *Proceedings of the 2nd European Symposium on Research in Computer Security (ESORICS 92)*, volume 648 of *Lecture Notes in Computer Science*, pages 41–54. Springer, 1992.

[Klo06] Tomaz Klobucar. Privacy and data protection in technology-enhanced professional learning. In *Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT-ICIW'06)*. IEEE Computer Society, 2006.

[Kor02] Larry Korba. Privacy in distributed electronic commerce. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)*. IEEE Computer Society, 2002.

[KS02] Günter Karjoth and Matthias Schunter. A privacy policy model for enterprises. In *Proceedings of the 15th IEEE Computer Security Foundations Workshop (CSFW'02)*, pages 271–281. IEEE Computer Society, 2002.

[KS03] Alfred Kobsa and Jörg Schreck. Privacy through pseudonymity in user-adaptive systems. *ACM Transactions on Internet Technology*, 3(2):149–183, 2003.

[KSH03] Günter Karjoth, Matthias Schunter, and Els Van Herreweghen. Translating privacy practices into privacy promises -how to promise what you can keep. In *Proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2003)*, pages 135–146. IEEE Computer Society, 2003.

[KSW02] Günter Karjoth, Matthias Schunter, and Michael Waidner. Privacy-enabled services for enterprises. In *Proceedings of the 13th International Workshop on Database and Expert Systems Applications*, pages 483–487. IEEE Computer Society, 2002.

[KSW03] Günter Karjoth, Matthias Schunter, and Michael Waidner. Platform for enterprise privacy practices: Privacy-enabled management of customer data. In *Privacy Enhancing Technologies*, volume 2482 of *Lecture Notes in Computer Science*, pages 194–198. Springer, 2003.

[KW09] Md. Enamul Kabir and Hua Wang. Conditional purpose based access control model for privacy protection. In Athman Bouguettaya and Xuemin Lin, editors, *ADC*, volume 92 of *CRPIT*, pages 137–144. Australian Computer Society, 2009.

[KWB10] Md. Enamul Kabir, Hua Wang, and Elisa Bertino. A conditional purpose-based access control model with dynamic roles. *Expert Systems with Applications*, In Press, Corrected Proof:–, 2010.

[Lib03] Liberty architecture framework for supporting Privacy Preference Expression Languages (PPELs), 2003. Retrieved on 16/10/2010, from http://www.projectliberty.org/liberty/files/whitepapers.

[Lib09] Liberty Alliance Project, 2009. Retrieved on 16/10/2010, from http://www.projectliberty.org.

[LLT00] Jim J. Longstaff, Mike A. Lockyer, and Michael G. Thick. A model of accountability, confidentiality and override for healthcare and other applications. In *Proceedings of the fifth ACM workshop on Role-based access control*, pages 71–76. ACM, 2000.

[LO02] Frans A. Lategan and Martin S. Oliver. A chinese wall approach to privacy policies for the web. In *Proceedings of the 26th Annual International Computer Software and Applications Conference*. IEEE Computer Society, 2002.

[LZY06] Zude Li, Guoqiang Zhan, and Xiaojun Ye. Role-based peer-to-peer model: Capture global pseudonymity for privacy protection. In *Advances in Web-Age Information Management*, volume 4016 of *Lecture Notes in Computer Science*, pages 193–204. Springer Berlin, 2006.

[Mal08] Bradley Malin. k-unlinkability: A privacy protection model for distributed data. *Data and Knowledge Engineering*, 64(1):294–311, 2008.

[MCW08] Robin McKenzie, Malcolm Crompton, and Colin Wallis. Use cases for identity management in E-government. *IEEE Security and Privacy*, 6(2):51–57, 2008.

[Mic09] Microsoft Corporation. Microsoft online privacy statement, October 2009. Retrieved on 16/10/2010, from http://privacy.microsoft.com/en-us/fullnotice.mspx.

[MPKB99] Ioannis Mavridis, George Pangalos, Marie Khair, and L. Bozios. Defining access control mechanisms for privacy protection in distributed medical databases. In *International Federation for Information Processing (IFIP) Working Conference on User Identification and Privacy Protection*, 1999.

[MR07] Dragana Martinovic and Victor Ralevich. Privacy issues in educational systems. *International Journal of Internet Technology and Secured Transactions*, 1:132–150, August 2007.

[NLBL08] Qun Ni, Dan Lin, Elisa Bertino, and Jorge Lobo. Conditional privacy-aware role based access control. In *Proceedings of the 12th European Symposium On Research In Computer Security (ESORICS 2007)*, volume 4734 of *Lecture Notes in Computer Science*, pages 72–89. Springer, 2008.

[NO93] Matunda Nyanchama and Sylvia Osborn. Role-based security: Pros, cons & some research directions. *ACM SIGSAC Review*, 11(2):11–17, 1993.

[NO94] Matunda Nyanchama and Sylvia Osborn. Access rights administration in role-based security systems. In *Proceedings of the IFIP WG11.3 Working Conference on Database Security VIII*, pages 37–56. North-Holland Publishing Co., 1994.

[NO99] Matunda Nyanchama and Sylvia Osborn. The role graph model and conflict of interest. *ACM Transactions on Information and System Security*, 2(1):3–33, February 1999.

[Not96] LouAnna Notargiacomo. Role-based access control in ORACLE7 and trusted ORACLE7. In *Proceedings of the first ACM Workshop on Role-based access control*. ACM, 1996.

[NTBL07] Qun Ni, Alberto Trombetta, Elisa Bertino, and Jorge Lobo. Privacy-aware role based access controls. In *Proceedings of the 12th ACM symposium on Access control models and technologies*, pages 41–50. ACM, 2007.

[Obe01] Hendrik J. Oberholzer. A privacy protection model to support personal privacy in relational databases. Master Thesis. Rand Afrikanns University, 2001.

[Odl03] Andrew M. Odlyzko. The unsolvable privacy problem and its implications for security technologies. In *Proceedings of the 8th Australasian Conference on Information Security and Privacy (ACISP 2003)*, volume 2727 of *Lecture Notes in Computer Science*. Springer Berlin, 2003.

[Off00] Office of the Privacy Commissioner of Canada. Personal information protection and electronic documents act, 2000. Retrieved on 16/10/2010, from http://www.priv.gc.ca/legislation/02_06_01_e.cfm.

[Oli95] Claude Oliver. Privacy, anonymity and accountability. *Computers & Security*, 14(6):489–490, 1995.

[Org] Organization for the Advancement of Structured Information Standards (OASIS). OASIS extensible access control markup language (xacml). Retrieved on 16/10/2010, from http://www.oasis-open.org/specs/#xacmlv2.0.

[Org80] Organisation of Economic Co-operation and Development (OECD). OECD guidelines on the protection of privacy and transborder flows of personal data, 1980. Retrieved on 16/10/2010, from http://www.oecd.org/document/18/0,3343,en_2649_34255_1815186_1_1_1_1,00.html.

[OSM00] Sylvia Osborn, Ravi Sandhu, and Qamar Munawer. Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Transactions on Information and System Security*, 3(2):85–106, May 2000.

[OT02] Sven Overhage and Peter Thomas. WS-specification: Specifying web services using UDDI improvements. In *Web, Web-Services, and Database Systems*, volume 2593 of *Lecture Notes in Computer Science*, pages 100–119. Springer, 2002.

[PAS02] Calvin Powers, Paul Ashley, and Matthias Schunter. Privacy promises, access control, and privacy management: Enforcing privacy throughout an enterprise by extending access controls. In *Proceedings of 3rd International Symposium on Electronic Commerce (ISEC'02)*. IEEE Computer Society, 2002.

[Pay10] Paypal. Privacy policy for paypal services, July 2010. Retrieved on 16/10/2010, from http://www.paypal.com/cgi-bin/webscr?cmd=p/gen/privacy-outside.

[PBS05] Craig Pearce, Peter Bertók, and Ron Van Schyndel. Protecting consumer data in composite web services. In *Security and Privacy in the Age of Ubiquitous Computing*, volume 181 of *IFIP International Federation for Information Processing*, pages 19–34. Springer Boston, 2005.

[Poo99] Ralph S. Poore. Anonymity, privacy, and trust. *Information Security Journal: A Global Perspective*, 8(3):16–20, 1999.

[Pri01] Privacy Commissioner. Privacy survey 2001, 2001. Retrieved on 16/10/2010, from http://www.privacy.org.nz/privacy-survey-2001/.

[Pri06] Privacy Commissioner. Privacy survey 2006, 2006. Retrieved on 16/10/2010, from http://www.privacy.org.nz/privacy-survey-2006/.

[Pri08] Privacy Commissioner. Privacy survey 2008, 2008. Retrieved on 16/10/2010, from http://www.privacy.org.nz/privacy-survey-200/.

[PW87] Andreas Pfitzmann and Michael Waidner. Networks without user observability. *Computers and Security*, 6(2):158–166, 1987.

[RBE03] Abdelmounaam Rezgui, Athman Bouguettaya, and Mohamed Y. Eltoweissy. Privacy on the web: Facts, challenges, and solutions. *IEEE Security and Privacy*, 1(6):40–49, 2003.

[RCHS03] Jason Reid, Ian Cheong, Matthew Henricksen, and Jason Smith. A novel use of RBAC to protect privacy in distributed health care information systems. In *The 8th Australasian Conference on Information Security and Privacy (ACISP 2003)*, 2003.

[RLTY07] Yi Ren, Min Luo, Zukai Tang, and Lingqing Ye. A composite privacy protection model. In *Proceedings of the Second International Workshop on Security (IWSEC) 2007*, volume 4752 of *Lecture Notes in Computer Science*, pages 380–395. Springer Berlin, 2007.

[Ros04] Richard Rosenberg. *The Social Impact of Computers*. Elsevier Academic Press, 2004.

[RR97] Michael K. Reiter and Aviel D. Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1:66–92, 1997.

[RS98] Chandramouli Ramaswamy and Ravi Sandhu. Role-based access control features in commercial database management systems. In *Proceedings of the 21st NIST-NCSC National Information Systems Security Conference*, pages 503–511, 1998.

[RWB02] Abdelmounaam Rezgui, Zhaoyang Wen, and Athman Bouguettata. Enforcing privacy in interoperable E-government applications. In *Proceedings of the 2002 annual national conference on Digital government research*, pages 1–10. Digital Government Society of North America, 2002.

[San96] Ravi Sandhu. Role hierarchies and constraints for lattice-based access controls. In *Proceedings of the 4th European Symposium on Research in*

Computer Security: Computer Security, volume 1146 of *Lecture Notes In Computer Science*, pages 65–79. Springer-Verlag, 1996.

[SCFY94] Ravi Sandhu, Edward Coyne, Hal Feinstein, and Charles Youman. Role-based access control: A multi-dimensional view. In *Proceedings of the Tenth Computer Security Applications Conference*, pages 54–62, 1994.

[SCFY96] Ravi Sandhu, Edward Coyne, Hal Feinstein, and Charles Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.

[SdV01] Pierangela Samarati and Sabrina C. de Vimercati. Access control: Policies, models, and mechanisms. In *Foundations of Security Analysis and Design*, volume 2171 of *Lecture Notes in Computer Science*, pages 137–196. Springer Berlin, 2001.

[SFK00] Ravi Sandhu, David Ferraiolo, and Richard Kuhn. The NIST model for role-based access control: Towards a unified standard. In *Proceedings of the 5th ACM Workshop on Role-Based Access Control (RBAC-00)*, pages 47–63. ACM Press, 2000.

[SH05] Haibo Shen and Fan Hong. A context-aware role-based access control model for web services. In *IEEE International Conference on e-Business Engineering (ICEBE 2005)*, pages 220–223. IEEE Computer Society, 2005.

[Shi09] Shibboleth, 2009. Retrieved on 16/10/2010, from http://shibboleth.internet2.edu.

[SHV99] Gregory Saunders, Michael Hitchens, and Vijay Varadharajan. An analysis of access control models. In *Proceedings of the 4th Australasian Conference on Information Security and Privacy*, volume 1587 of *Lecture Notes in Computer Science*, pages 281–293. Springer, 1999.

[SHW02] Matthias Schunter, Els Van Herreweghen, and Michael Waidner. Expressive privacy promises–how to improve the platform for privacy preferences (P3P). Position paper for W3C Workshop on the Future of P3P, 2002. Retrieved on 16/10/2010, from http://www.w3.org/2002/p3p-ws/pp/ibm-zuerich.pdf.

[Sim08] Aislinn Simpson. NHS: Personal details of 18,000 staff 'lost in the post', September 2008. Retrieved on 16/10/2010, from http://www.telegraph.co.uk/health/2965231/NHS-Personal-details-of-18000-staff-lost-in-the-post.html.

[SK03] Sandra Steinbrecher and Stefan Kopsell. Modelling unlinkability. In *International Workshop on Privacy Enhancing Technologies (PET)*, volume 2760 of *Lecture Notes in Computer Science*, pages 32–47. Springer Berlin, 2003.

[Sol06] Daniel J. Solove. A taxonomy of privacy. *University of Pennsylvania Law Review*, 154(3):477–560, January 2006.

[Spi87] Michael Spivey. *The Z Notation: A Reference Manual*. International Series in Computer Science. Prentice Hall, 1987.

[SS94] Ravi Sandhu and Pierrangela Samarati. Access control: Principles and practice. *IEEE Communications Magazine*, 32(9):40–48, 1994.

[SS06] Phiniki Stouppa and Thomas Studer. A formal model of data privacy. In *Perspectives of Systems Informatics*, volume 4378 of *Lecture Notes in Computer Science*, pages 400–408. Springer Berlin, 2006.

[ST94] Ravi Sandhu and Roshan Thomas. Conceptual foundations for a model of task-based authorizations. In *Proceedings of the 7th Computer Security Foundations Workshop (CSFW '94)*, pages 66–79. IEEE Computer Society Press, 1994.

[Ste97] Mark Stefik. Trusted systems. *Scientific American*, 276(3):78–81, 1997.

[Sto02] Jennifer Stoddart. Respecting privacy in E-government. In *Certification and Security in E-Services*, volume 255 of *IFIP Conference Proceedings*, pages 201–210. Kluwer, 2002.

[Swe02] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[SWI09] SWITCHaai, 2009. Retrieved on 16/10/2010, from http://www.switch.ch/aai/.

[SZ97] Richard Simon and Mary E. Zurko. Separation of duty in role-based environments. In *Proceedings of the 10th Computer Security Foundations Workshop (CSFW '97)*. IEEE Computer Society Press, 1997.

[The95] The European Parliament and The Council of The European Union. Directive 95/46/ec of the european parliament and of the council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data, November 1995. Retrieved on 16/10/2010, from http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:EN:HTML.

[The06] The European Parliament and The Council of The European Union. EU directive concerning the processing of personal data and the protection of privacy in the telecommunications sector, June 2006. Retrieved on 16/10/2010, from http://www.ictregulationtoolkit.org/en/Publication.1492.html.

[TM01] Herman T. Tavani and James H. Moor. Privacy protection, control of information, and privacy-enhancing technologies. *ACM SIGCAS Computers and Society*, 31(1):6–11, 2001.

[TW09] Michael Carl Tschantz and Jeannette M. Wing. Formal methods for privacy. In Ana Cavalcanti and Dennis Dams, editors, *FM 2009: Proceedings of The Second World Congress on Formal Methods*, volume 5850 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2009.

[Uni96] United States Department of Health. The health insurance portability and accountability act of 1996, 1996. Retrieved on 16/10/2010, from http://www.hhs.gov/ocr/hipaa.

[WB90] Samuel D. Warren and Louis D. Brandeis. The right to privacy. *Harvard Law Review*, 4(5):193–220, 1890.

[WC08] Janice Warner and Soon Ae Chun. A citizen privacy protection model for e-government mashup services. In *Proceedings of the 2008 International Conference on Digital Government Research*, pages 188–196. Digital Government Society of North America, 2008.

[Web] WebTrust. Retrieved on 16/10/2010, from http://www.webtrust.org/.

[Wes70] Alan F. Westin. *Privacy and Freedom*. The Bodley Head Ltd, 1970.

[Wik] Wikipedia. Privacy. Retrieved on 16/10/2010, from http://en.wikipedia.org/wiki/Privacy.

[WLF08] Chundong Wang, Ting Li, and Lichun Feng. Context-aware environment-role-based access control model for web services. In *2008 International Conference on Multimedia and Ubiquitous Engineering (MUE 2008)*, pages 288–293. IEEE Computer Society, 2008.

[Wor06] World Wide Web Consortium (W3C). The platform for privacy preferences 1.1 (P3P1.1) specification, 2006. Retrieved on 16/10/2010, from http://www.w3.org/TR/2006/NOTE-P3P11-20061113/.

[Wor07] World Wide Web Consortium (W3C). Platform for Privacy Preference (P3P) Project, 2007. Retrieved on 16/10/2010, from http://www.w3.org/P3P/.

[XCHW09] Fei Xu, K. P. Chow, Jingsha He, and Xu Wu. Privacy reference monitor - a computer model for law compliant privacy protection. In *ICPADS '09: Proceedings of the 2009 15th International Conference on Parallel and Distributed Systems*, pages 572–577, Washington, DC, USA, 2009. IEEE Computer Society.

[XHWX09] Fei Xu, Jingsha He, Xu Wu, and Jing Xu. A privacy-enhanced access control model. In *NSWCTC '09: Proceedings of the 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing*, pages 703–706, Washington, DC, USA, 2009. IEEE Computer Society.

[Yah06] Yahoo! Inc. Yahoo! privacy policy, November 2006. Retrieved on 16/10/2010, from http://info.yahoo.com/privacy/us/yahoo/details.html.

[YBZ08] Naikuo Yang, Howard Barringer, and Ning Zhang. A purpose-based access control model. *Journal of Information Assurance and Security (JIAS)*, 3(1):51–58, 2008.

[YTT97] Masashi Yasuda, Takayuki Tachikawa, and Makoto Takizawa. Information flow in a purpose-oriented access control model. In *1997 International Conference on Parallel and Distributed Systems (ICPADS'97)*. IEEE Computer Society, 1997.

[YTT98a] Masashi Yasuda, Takayuki Tachikawa, and Makoto Takizawa. A purpose-oriented access control model. In *The 13th International Conference on Information Networking (ICOIN'98)*. IEEE Computer Society, 1998.

[YTT98b] Masashi Yasuda, Takayuki Tachikawa, and Makoto Takizawa. A purpose-oriented access control model for information flow management. In *Proceedings of the IFIP TC11 14th International Conference on Information Security (SEC98)*, 1998.

[YTT98c] Masashi Yasuda, Takayuki Tachikawa, and Makoto Takizawa. A purpose-oriented access control model for object-based systems. In *Proceedings of the the 1st IEEE International Symposium on Object-Oriented Real-Time Distributed Computing*. IEEE Computer Society, 1998.