DOCTOR OF SCIENCES

Managing and Enforcing Privacy-aware Policies in IT Systems

Rath, Thavy Mony Annanda

Award date:
2015

Awarding institution:
University of Namur

Managing and Enforcing Privacy-Aware Policies in IT Systems

Annanda Thavymony RATH

Faculty of Computer Science

University of Namur, Belgium

1st December 2015

Jury

1. Prof. Jean-Marie Jacquet, Faculty of Computer Science, University of Namur, Belgium (President).
2. Prof. Jean-Noël Colin, Faculty of Computer Science, University of Namur, Belgium (Supervisor).
3. Prof. Benoit Frenay, Faculty of Computer Science, University of Namur, Belgium (Internal Reviewer).
4. Prof. Rüdiger Grimm, Institute of Economic and Administrative Computer Science, University of Koblenz, Germany (External Reviewer).
5. Prof. Michaël Petit, Faculty of Computer Science, University of Namur, Belgium (Internal Reviewer).
6. Prof. Denis Zampunieris, Faculty of Communication and Technology, University of Luxembourg, Luxembourg (External Reviewer).

The PhD defense is held on the 1st of December 2015 at Computer Science Faculty in the University of Namur.

I would like to dedicate this thesis to my sisters, brother and parents ...

Acknowledgements

The whole adventure started on the impulse of Prof. Jean-Noël Colin. His vast knowledge and razor-sharp rigour helped me improve my knowledge on the field that I have never worked on. Over the past five years, Prof. Jean-Noël not only taught me how to do research, but also how to write a good research paper. This is really important for the development of my future career as the academic personnel and researcher.

As a part of research activities, Prof. Jean-Noël Colin has helped me find funding for research conference participation. Because of his generosity, I have participated in the remarkable numbers of conferences around the world. This motivated me and provided a chance to improve and update regularly my knowledge in the field.

In 2013, Prof. Jean-Noël Colin enrolled me in the European Research Project (P5: Privacy Preserving Perimeter Protection Project). This, again, provided the great chance to improve my knowledge and to gain experience in real working environment. Furthermore, as researcher in P5, I had the chance to work with many professional people across number of fields. This opened a new window to my future career.

I also thank my thesis committee members: Prof. Michaël Petit and Prof. Rüdiger Grimm for their valuable time in helping me improve my knowledge as well as my thesis report. I also have a special thank for all friends who helped and encouraged me during my hardship.

Finally and foremost, I am immensely grateful to my parents who kept me going through the straits of this thesis and dedicate this entirely to them.

Abstract

Private data is generally governed by privacy policy, which often places restrictions on the *purposes* for which a governed entity may use such data (e.g. EU directive 95/46/EC requires data processor to use private data only for the *purposes* it intends for). Laws require data processor to ensure the correct usage of data; this leads to the need of privacy policy enforcement. To enforce privacy policy using formal or automated methods requires a semantics of *purpose* restrictions to determine whether an action is for a *purpose* and that *purpose* could be achieved or not once access permission is granted. We model ***purpose*** as a ***workflow*** and we argue that an action is for a *purpose* if and only if that action is part of a plan for the satisfaction of that *purpose*. Based on that formalisation, we propose an approach to enforce *purpose*. In our approach, the access authorisation is based not only on the control of workflow process, but also on the estimation of the level of certainty of *purpose* achievement, which is determined by ***purpose*** **achievement prediction** (a probabilistic system estimating how likely user can reach his claimed *purpose* after access permission is granted). The prediction module is built using **Association Rule Learning** method where user's access history and contextual information are used as the input data for rule analysis. The semantics of *purpose* with our enforcement approach enable us to create and implement an algorithm for enforcing the privacy policies, and to describe formally and compare rigorously with previous enforcement methods. To validate our semantics, we provide an example application, build a prototype and validate it against the existing enforcement methods with the specific validation criteria.

Contents

List of Figures

List of Acronyms

AA Attribute Authority

ACM Association for Computing Machinery

C Contribution

CC Creative Commons

CCTV Closed-circuit television

CONF Confidence

CVV Contextual Variables Validation

DAC Discretionary Access Control

DPRL Digital Property Rights Language

DRM Digital Rights Management

DS Data Sequence

EHR Election Health Record

EPAL Enterprise Privacy Authorisation Language

GBAC Group-based Access Control

HBAC History-based Access Control

HDCP High Bandwidth Content Protection

HL7 Health Level 7

LMF Local Media File

LWDRM Light Weight Digital Rights Management

MAC Mandatory Access Control

MB Mega Bytes

MPEG-21 Moving Picture Expert Group

OASIS Organisation for the Advancement of Structured Information Standards

ODF Obligation Definition Function

ODRL Open Digital Rights Language

OMA Open Mobile Alliance

OrBAC Organisation-based Access Control

P-RBAC Privacy-aware Role-based Access Control

P5 Privacy Preserving Perimeter Protection Project

PA Purpose Achievement

PACM Privacy-aware Access Control Module

PAV Past access Analysis Variable

PDP Policy Decision Point

PEP Policy Enforcement Point

PF Privacy-aware Filter

PG Purpose Graph

PPM Purpose Prediction Module

RBAC Role-based Access Control

RDF Resource Description Framework

REL Rights Expression Language

Rel-BAC Relationship-based Access Control

RQ Research Question

SMF Signed Media File

SMP Session Management Point

TG Task Graph

TTP Trusted Third Party

UCDP Usage Control and Decision Point

UCON Usage Control

WMA Window Media Audio

WMDRM Window Media Digital Right Management

WRM Workflow Reference Monitor

XACML eXtensible Access Control Markup Language

XML eXtensible Markup Language

XrML eXtensible Rights Markup Language

Chapter 1

Introduction

1.1 Protecting Private Data

Private Data is a category of information associated with an individual person and it can be used to uniquely identify, contact or locate that person. For example, health records are considered as private data. Unlike public data that can be accessible without any restriction, accessing private data needs an authorisation from data owner as required by laws [28].

EU Directive 95/46/EC, Article 2 (a) [28] defines private data as any information related to an identifiable natural person. An identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity.

With widespread use of the Internet and the advance of multimedia technology like compression techniques, digital content can be distributed instantaneously across the Internet to end-users around the globe. However, without proper protection, digital content (in particular, private data of individual) can be copied, altered or transferred, which results in privacy violation. For example, the use of social media networks such as Facebook or Twitter to circulate unauthorised personal videos or photos is a form of privacy violation and by laws in some countries like US [59] or European countries [28], such activity is considered as illegal. Although there are laws applied to the processing of private data of individual in a digital world, the enforcement of such laws is still the major concern. This is because to effectively enforce such laws in vast network (e.g. Facebook) requires the powerful data usage control technology that is able to manage properly and effectively the circulation of such data.

1.2 Legislations: Privacy Policies and Purpose of Use

Purpose of use is a key concept in privacy policies. In 1995, EU proposed the Directive 95/46/EC [28] on the protection of individuals with regard to the processing of private data and on the free movement of such data. It aims at ensuring that private data is used in compliance with its intended PURPOSE.

"EU Directive 95/46/EC, Article 6 states that : Member States shall provide that personal data must be [...] collected for specified, explicit and legitimate purposes and not further processed in a way incompatible with those purposes...

Article 8 (3): the person or institution processing personal data must take a full responsibility and make sure that such data must be safely processed... "

The United States also has laws putting purpose restrictions on information in some domains such as the Health Insurance Portability and Accountability Act (HIPAA) for medical information and the Gramm-Leach-Bliley Act [59] for financial records. Other example of privacy laws is the special agreement, such as the Safe Harbor Frameworks[1], which is implemented between the European Union, the United States and Switzerland. These laws and best practices motivate organisations to discuss in their privacy policies the purposes for which they will use information. Some privacy policies inform users that the policy provider may use certain information for certain purposes. For example, the privacy policy of Wallonie Healthcare Network in Belgium [54] states, "We may disclose your [protected health information] for public health activities and researches [. . .] with anonymity". Other examples include the privacy policy of Yahoo Email [2], which states that "Yahoo's practice is not to use the content of messages stored in your Yahoo Mail account for marketing purposes". Some policies even limit the use of certain information to an explicit list of purposes. Privacy policy of Facebook [3] states that "Facebook provides advertisers with information of user for marketing purpose. However, personally identifying information is removed, or combined with other information so that it is no longer a personally identifying information".

These examples show that each organisation has its own privacy policies and to verify that an organisation obeys its privacy policies requires semantics of purpose restrictions and policy enforcement. In particular, for policy enforcement, it requires the

[1] http://www.export.gov/safeharbor/, retrieved Nov. 23th, 2014.
[2] https://info.yahoo.com/privacy/us/yahoo/ latest access: 19th March 2015.
[3] https://www.facebook.com/legal/terms latest access: 19th March 2015.

ability to determine that the organisation under scrutiny obeys at least two classes of purpose restrictions. As shown in the example rule from Yahoo, the first requirement is that the organisation does not use certain sensitive information for a given purpose. The second, as the example rule from Facebook, is that the organisation uses certain sensitive information only for a given list of purposes. In summary, two legal responsibilities of organisation when handling private data are:

1. When processing private data, organisation needs to ensure that it is properly protected and it is used in compliance with its original purpose, EU Directive 95/46/EC, Article 6.

2. When private data is shared across boundary, organisation is bound to a legal responsibility to make sure that such data must be safely processed, EU Directive 95/46/EC, Article 8(3).

1.3 Research Methodology

The research methodology we have followed to address our research questions is inspired by the guidelines of Peffers et al [60], design science research methodology. However, our method deviates slightly from Peffers in that we separate our research activities into only 4 instead of 6 activities proposed by Peffers. We combine the "demonstration " and "evaluation" activities into one single activity. In this section, we begin with the motivation where the problems and challenges are identified. Then, we detail the research objectives and a brief description of the solutions. The following section focuses on the design and development activity where we point out our proposed system architecture and techniques to be used to support our proposed solutions. Finally, the demonstration and evaluation section where we discuss system prototype implementation and its evaluation against a set of defined criteria .

1.3.1 Motivation

Private data protection in distributed healthcare is not a new research topic, but although many solutions have been proposed [50][35][38][52], only a few simple ones have been implemented. For example, the HL7 framework [37], which intends to make healthcare systems to be able to work in interoperable way by providing the standard messaging protocol for sharing healthcare records between different healthcare information systems. However, HL7 is more about message protocol rather than a tool to control and protect the usage of health records in distributed environment in a secure manner. Similarly, the technologies like Digital Rights Management (DRM), some of them [83][71] can also be used, but they cannot provide the security we need as required by law [28] for processing of private data. This is because the existing

DRM technologies are not specifically built for private data. They are built to protect commercial contents (e.g. multimedia contents); they are content-specific[66][71]. This rules out the possibility of using existing DRM technologies, without complement or extra support functionalities, to control the processing of private data.

So far we are not aware of a complete solution designed for managing and enforcing privacy-aware usage control policies in distributed healthcare. Consequently, it would be best to design a dedicated system for distributed healthcare, in a way that addresses the requirements [10] in such system.

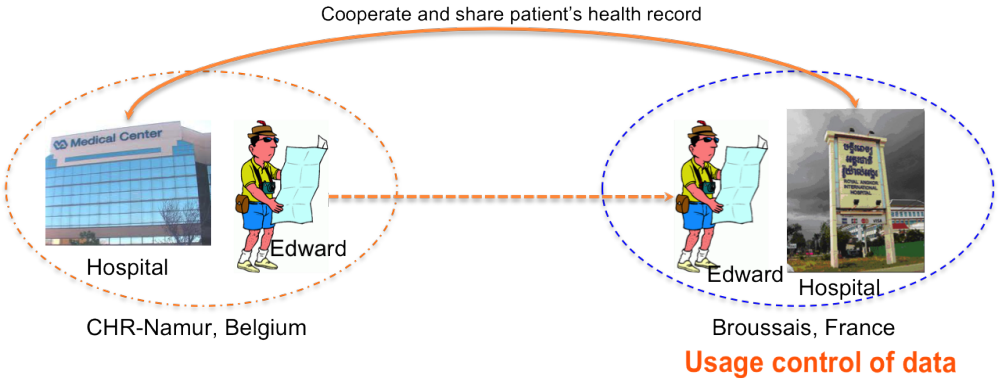

Figure 1.1: Example of two hospitals share Edward's health records. Edward's health records are transferred to Broussais for Edward's heart treatment purpose.

1.3.2 Thesis Objectives

We start with an example in Figure 1.1, an informal private data processing scenario in distributed healthcare system. We suppose that two hospitals, one in Paris France (Broussais) another in Namur Belgium (CHR-Namur), have signed a cooperation agreement on sharing their patients' medical records. Under the agreement both hospitals can share their patient's medical records when needed. The processing of patient's medical record must be strictly controlled and must comply with the policies defined by the data owner. For example, if Broussais processes the medical records belonging to CHR-Namur, Broussais must fully respect the data usage policy defined by CHR-Namur (representing the data owner).

A Belgium citizen, Edward, has registered in CHR-Namur hospital for heart treatment. All the medical records concerning his heart are managed by CHR-Namur under the heart treatment purpose. This means, CHR-Namur can share his medical records for such purpose and only for his treatment. Some time later, when Edward visits Paris, he faces a heart disorder and needs an emergency treatment (surgery). Edward

is hosted at Broussais hospital in Paris. Before performing heart surgery, the cardiologist needs his past heart medical records for pre-surgery examination. The cardiologist acquires those medical records from CHR-Namur under CHR-Namur-Broussais agreement. The medical records are transferred to Broussais and they stay there for a limited period of time, to be precise, during the treatment of Edward. During those periods, cardiologist can examine Edward's medical records given that the usage policies defined by CHR-Namur are respected.

The legislations, Directive 95/46/EC, require CHR-Namur to protect the health records of Edward when processing them locally and CHR-Namur takes also a full responsibility when sharing Edward's medical records to Broussais. Broussais, on the other hand, bonds to the CHR-Namur-Broussais agreement. In short, both parties need to ensure that Edward's medical records are correctly processed as their intended purpose. Given this scenario, we can see that managing and enforcing privacy usage control policy is really important because it is the only way to ensure that both parties respect the privacy policies they defined when the private data of patient stay at their local system. **We believe that it is possible to enforce privacy-aware policies if we have a usage control system integrated with an effective and efficient purpose enforcement technique that is able to prevent the usage of data that does not correspond to the data owner's intended purpose**. Thus, the objectives of thesis are:

"to design, implement and evaluate a privacy-aware usage control system supporting purpose enforcement for the processing of private data."

In order to achieve our research goal, we need to cover the following issues: design of (1) privacy-aware usage control model, (2) privacy-aware usage control policy and policy expression language, (3) privacy-aware usage control policy enforcement technique and (4) privacy-aware usage control system architecture supporting purpose enforcement. The research is broken down into five main research questions that address different issues we listed above.

RQ1.1: what are the requirements for the protection of private data in distributed system? We focus on distributed healthcare. Therefore, we expect to investigate the existing healthcare information systems. Since private data is protected by laws, what are the legal and technical requirements for the processing of such data? We will need to study thoroughly and analyse deeply the legal documents for the protection of such data. European laws will be the target of our study. However, we will also take a look at USA and Canada laws.

RQ1.2: since the purpose of use plays an important role in privacy policies, we need to define the meaning of purpose and its model. **How to model the purpose of use**

in such a way so that it can be easily managed and effectively enforced in distributed environment?

RQ1.3: what usage control model should be used to effectively control the usage of private data in distributed environment? Are the existing usage control models good enough to be used in our context? We need to survey different existing usage control models, and based on the requirements identified in RQ1.1, we determine the appropriate usage control model. Once the usage control model is defined, **which policy language should be used?**

RQ1.4: what are the efficient and effective ways to enforce the purpose of use for privacy-aware policy in distributed environment?

RQ1.5: what are the existing usage control technologies that can be used to control access and usage of private data in distributed environment? We will study the existing technologies and examine how different techniques may have varying degree of effectiveness when used in our system context. If the existing technologies are not appropriate to be used in our context, we consider the extension of the existing technologies or a completely new one.

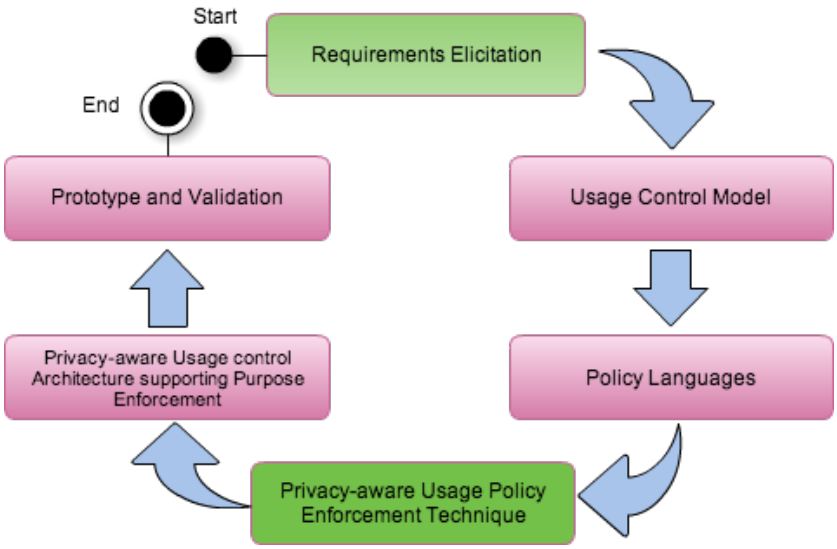

Figure 1.2: An illustration of step-by-step processes to address the research questions.

1.3.3 Design and Development

In design and development activity, we identify desired functionalities and its architecture and then create the actual usage control system that can be used to support the processing of private data. In order to create a usage control system, we propose a step-by-step processes as presented in Figure 1.2. We start from requirements elicitation. The requirements are collected based on the study and analysis of the existing system [54] and also study of the EU directive 95/46/EC [28]. Based on those requirements, in step 2, we work on usage control model. The proposed usage control model takes into account all the requirements we identified in step 1. The usage control policies derived from the proposed usage control model need to be expressed in the machine readable policy language. Thus, our third step is to identify the appropriate policy language that can be used to express such policy. In order to make sure that client respects the usage control policy defined by data owner, we need to create the policy enforcement technique. The fourth step dedicates to this work. The fifth step is the development of usage control system architecture that takes into account all the desired functionalities and requirements for controlling the usage of private data. The final step in design and development is the creation of the prototype of the usage control system.

1.3.4 Demonstration: Application Domains

We mainly focus on the protection of private data. At the early state of our research we considered two application domains: the distributed healthcare information systems and social networks (e.g. Facebook). However, after a series of researches, we find that although social networks deal largely with private data, privacy protection is less important when comparing with distributed healthcare since healthcare systems deal mostly with sensitive private data. Therefore, we focus our research on distributed healthcare as our main domain application for implementation and validation. Other domain that we also consider in this thesis is the perimeter protection system [9] where privacy protection is also the important issue.

Distributed Healthcare: health record history is important in the course of a treatment process for the proper continuing care of patients. Over last decade, with the increase of the electronic materials in healthcare and the improvement of network and system, Electronic Health Records (EHR) has become increasingly common and widespread to replace the traditional paper-based record. However, making the information available electronically poses new security concerns, especially when exchanging it between different healthcare institutions, as it is more vulnerable to attacks compared with paper-based.

Given the fact that healthcare information systems deal largely with sensitive private data, failure to secure such data can lead to huge fines, lawsuits or long-term loss of patients' trust. Yet to provide adequate security, in a manner that is not burdensome

to patient can be a major challenge. Our research focuses on this type of system; particularly, a distributed system where two healthcare institutions share patients' health records for a particular purpose. Our main research goal is to produce a system that is able to effectively control the usage of data when it is being processed at client side application.

Since 2013 we have been participating in the European project, the Privacy Preserving Perimeter Protection. Our role in the project is to design the privacy enhancing technology that could be used to protect personal data of the individual affected from the surveillance in the protected facility. This provides us a real application domain where we could implement our finding.

P5 Project: Privacy Preserving Perimeter Protection Project (P5) is European FP7 (http://www.foi.se/p5) project for the protection of critical infrastructures to benefit the sustainability of society and future well-being of the European Citizens. The goal of the P5 project is an intelligent perimeter proactive surveillance system that works robustly under a wide range of weather and lighting conditions and that has strong privacy preserving features. The system will monitor the region outside the security area of critical buildings and infrastructures, and give early warning if terrestrial or airborne threats are detected. A multispectral sensor suite comprising both passive and active sensor is used (e.g. a system based on radar, visual and thermal sensors). The sensor suite will be complemented with advanced algorithms for information fusion, object detection and classification, privacy preservation and high level modelling of intent and behaviour analysis.

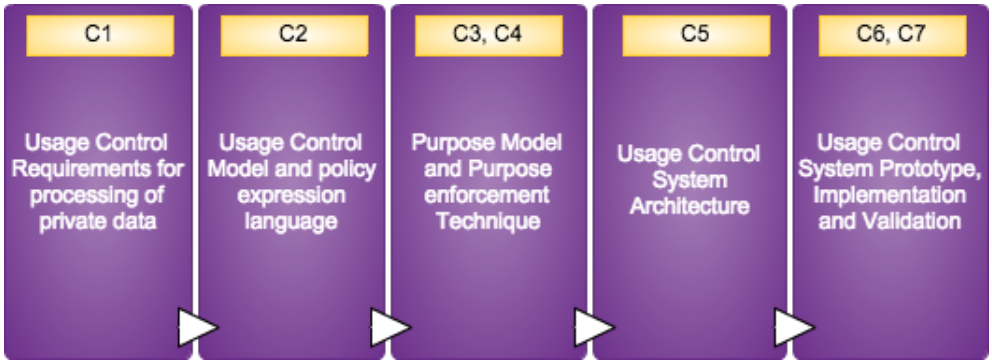

Figure 1.3: A graphical summary of the contributions.

1.4 Claimed Contributions

The main contribution of this thesis is a sound foundation for purpose management and enforcement for privacy policies. Specifically, it consists of (see Figure 1.3):

1. C1: a systematic investigation and understanding of security requirements for the protection of private data. We target healthcare system for our study. We start with a systematic analysis of various access and usage control requirements of the Réseau Santé Wallon (RSW) [54], a network of healthcare system in French speaking region in Belgium. At the same time, to fully understand the legal requirements for the processing of such data, we study thoroughly the Directive 95/46/EC concerning the protection of private data of individual. The motivation behind the study of 95/46/EC directive is to align the technical and legal requirements. From the conclusion drawn from the survey, we propose generic usage control requirements for distributed healthcare system. The work on access and usage control requirements was published in the 7th International Conference on Health Informatics, Barcelona, Spain, 2013 [10].

2. C2: a systematic investigation of access and usage control models. The motivation behind our investigation is to find out which models can be used to address the access and usage control to private data. We start with a survey of various access and usage control models. A survey shows that although there are numbers of works focusing on access control model for private data, there is a lack of focus on usage control model for private data in distributed environment. From the conclusion drawn in survey and the usage control requirements in C1, we propose a usage control model where "purpose of use" is incorporated. The result of our work on privacy-aware usage control model was published in the Fourth International Conference on eHealth, Telemedicine and Social Medicine (eTELEMED 2012) [11]. More details about the proposed usage control model, one can find in Chapter 4 (Section 4.6).

3. C3: an understanding of "Purpose" and its role in privacy policies. This includes the understanding of the meaning of "Purpose" by public and the meaning of "Purpose" defined in legislation, in particular, Directive 95/46/EC. From this study, we draw a formal definition of "Purpose". Based on the formal definition of purpose, we propose a purpose model. Purpose is modelled as workflow. Based on this formulation we propose a purpose enforcement technique in C4. The work on purpose modelling was published in the Fourth International Conference on eHealth, Telemedicine and Social Medicine (eTELEMED 2012) [11].

4. C4: we propose a purpose enforcement technique, which is based on the prediction of the purpose achievement. The prediction model is built based on Association Rule Learning technique [4] where the information, such as user's role, contextual information and user's past access history are used as input data for rule analysis. The proposed technique is able to tell if the purpose can be achieved successfully or not once access permission is granted. It is worth noting that before arriving at the conclusion of using association rule learning method for analysing the

access log of user, we have studied different prediction and forecasting methods, such as Markov Decision Process [65], Naive Bayes [34], Logistic Regression [47], k-nearest neighbor algorithm [21] and Decision Tree [67]. Among them, only four can be used in our context: Markov Decision Process, Decision Tree, Association Rule Learning and Naive Bayes, but with different degree of effectiveness. However, Association Rule Learning is the best among them (see Section 6.1.3 for more details). The result of this research, "Towards enforcement of purpose for privacy policy in distributed healthcare", was published in 3rd IEEE International Workshop on Consumer eHealth Platforms, Services and Applications, Las Vegas, USA, CCNC 2013 [12].

5. C5: privacy-aware usage control architecture supporting purpose enforcement. To support the proposed purpose enforcement technique for privacy policies, we need to have a dedicated usage control system being able to enforce the privacy policies using our proposed purpose enforcement technique. With that reason, we propose a complete and comprehensive privacy-aware usage control architecture for distributed system. This system will act as a secure platform at remote client system. It is responsible for ensuring that the privacy policy and the purpose of use are properly enforced. The work related to the design of usage control system supporting purpose enforcement was published in International Journal of Security and Networks, August 2013 [13].

6. C6: a complete implementation and validation of all the definitions, properties and analyses in a toolset. A privacy-aware usage control has been implemented in Java. The Enterprise-Java-XACML[1] policy decision engine has been used in our implementation. However, in order to support our proposed enforcement technique and system architecture, we need to extend the core engine of Enterprise-Java-XACML [87]. This is because Enterprise-Java-XACML is originally an attribute-based access control engine. Beside of the use of Enterprise-Java-XACML, we also use XACML policy language to express both the privacy-aware usage policies and purpose enforcement policies.

7. C7: a complete implementation of our privacy protection method in P5 project [32] for preserving privacy of individual affected from the surveillance, the P5 system is our application domain used for demonstration and validation of our proposed solution for the protection of private data. These include the implementation of privacy-aware access control model and system and the implementation of privacy policy enforcement technique based on user access log observation. This contribution was published in the 29th Annual International Federation

[1]Enterprise-Java-XACML is intent to fully implement OASIS XACML 2.0, and provide a high performance and good usability that can be used in enterprise environment.

for Information Processing (IFIP), WG 11.3 Working Conference on Data and Applications Security and Privacy, Fairfax, VA, USA , 2015 [9].

1.5 Literature Survey Method: Survey Protocol and Materials

The research method we have followed to collect and review papers is inspired by the guidelines of Kitchenham et al [45]. However, our method deviates slightly from Kichenham in that we leave out the detailed quantitative analysis to favour an in-depth qualitative analysis. In this section, we begin with the presentation of the survey protocol. It then details the survey (analysis) materials used to harvest data systematically.

The survey protocol is divided into five main steps that go from the selection of papers to their analysis. This process is depicted in Figure 1.4. Starting from the top of the diagram the survey materials is composed of the whole set of papers, technical reports and previous thesis that are collected and recorded in our databases. The study of previous thesis is important because it provides us insight of what have been done, which problems have and have not been addressed. The materials we collect are based on the 21 materials provided by our advisor; and all the references in each provided paper are examined and the papers that are relevant to our research are kept. In addition to that some papers are searched from Internet based on our defined key words (see phase 2, Figure 1.4).

The second step is filtering process during which papers are kept for a complete review. The filtering is based on the search for the keywords in the papers and preliminary review of the abstract and introduction of those papers. In essence, the papers that are not related to our addressing issues are discarded.

The third step is the classification. Since protecting private data in distributed environment requires us to address different issues from requirements, access and usage control to purpose management and enforcement, we need to classify the papers according to their fields. We define five different categories. The first category is the access and usage control requirements. The second category is the access and usage control model and policy expression languages. The third category is the purpose management and enforcement technique. The fourth category is usage control technology. The fifth category is the distributed usage control architecture supporting purpose enforcement.

The fourth step is the complete review of each class of papers. The paper review is splitted between the old and recent papers in order to be able to follow the progress of the research in the area. Our final step is to analyse those papers to answer our research questions.

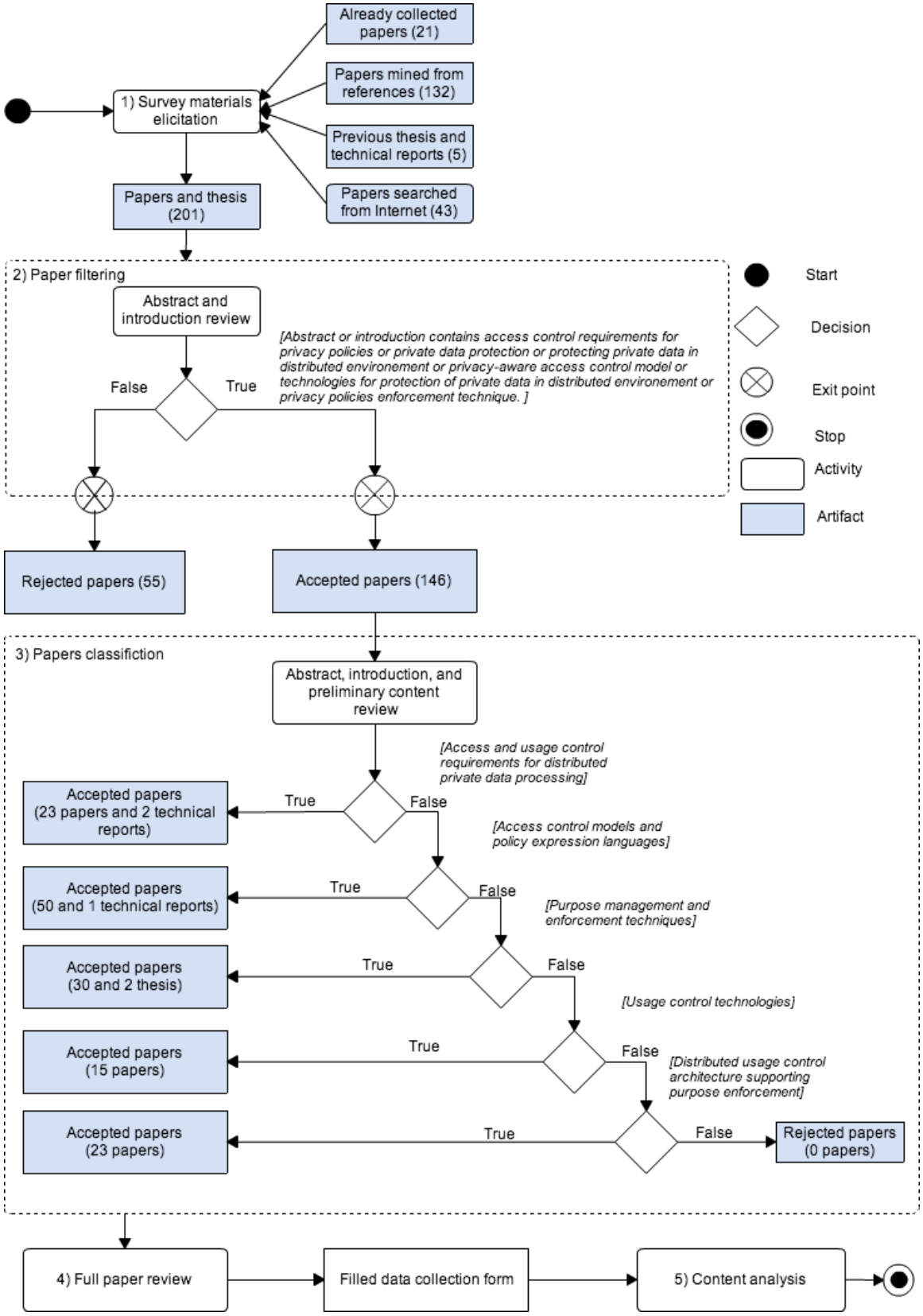

Figure 1.4: Survey protocol: dataflow

Our survey materials covers (1) research papers published in peer-reviewed workshops, conferences and journals, (2) thesis, books and other manuscripts, such as technical reports published by commercial or public institutions.

We consider three input sources for our survey. Firstly, we focus on the initial set of papers proposed by our advisor and numbers of thesis that are relevant to our addressing issue. Then, scanning the references section in each paper, a list of relevant papers (papers' titles) is extracted from the references section of those papers. After having collected list of relevant papers, we start collecting those papers, by directly contacting the authors of those papers or in some cases; we need to buy them from the publishers. Secondly, IEEE, ACM, and Springer databases are also the sources of our survey. We search through their databases by using our key words (see Figure 1.4, phase 2) and select the most relevant papers to our problem. Third source is Internet, with the help of Internet search engine, we are able to find number of interesting research papers.

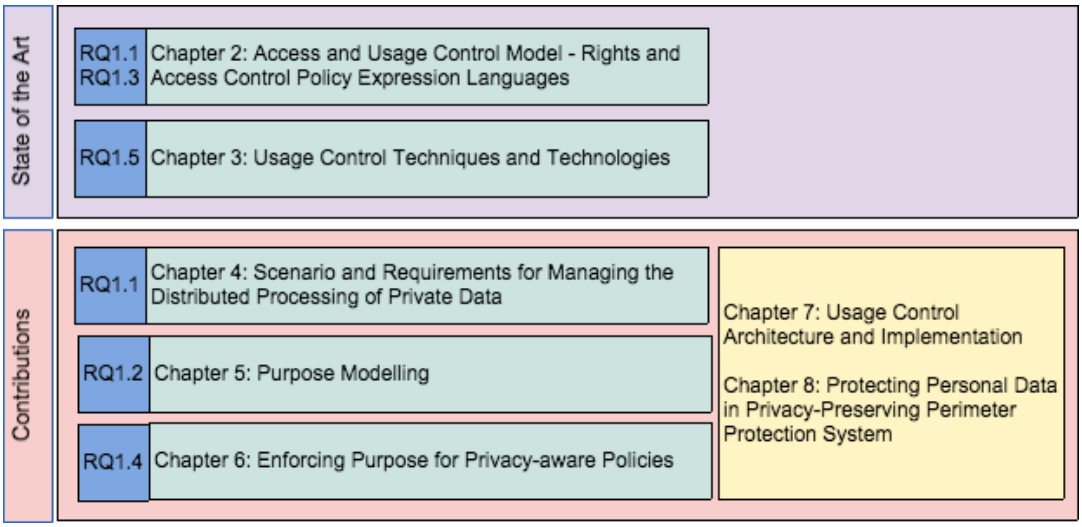

Figure 1.5: Thesis Map

1.6 Thesis's Structure

Chapters are organised in layers, as shown in Figure 1.5. We group the chapters into two main parts. The first two chapters are about the state of the art and the work related to access control model, usage control technique and rights and access control authorisation language. The second part, that covers the rest of the chapters, is the contribution. **Chapter 2** revisits the related work on access and usage control model

where different access control models are presented; the advantages and disadvantages are also listed (see the contribution **C1** in Section 1.4). Then, we point out which access and usage control model should be used in our context. From the access and usage control model we chose, we start investigating the rights and access control expression language (**C2**). Our main goal is to identify which rights and access authorisation language is appropriate for expressing the access as well as usage control policy of the model we identified. The thorough study of different commercial as well as open standard rights and access authorisation language is conducted in this chapter.

In **Chapter 3**, we present the result of a systematic analysis on the existing usage control techniques and technologies (**C2**). We identified different usage control techniques and technologies used for controlling the usage of data for both centralised and distributed environment. Our findings show that although there are numbers of usage control technologies (e.g. Digital Rights Management System (DRMs)), they are designed to be used to protect mostly the commercial content, and none of them are designed to address the protection of private data. This rules out the possibility of using the existing technology to solve our addressing issues.

The contribution part covers the chapters 4-8. We start our contribution from **Chapter 4** where we discuss the needs for controlling the use of private data in distributed environment. This chapter focuses mostly on the private data protection requirements (**C1**). These include the legal and technical requirements; for legal requirement, we mainly focus on the EU Directive 95/46/EC. In addition to that, we present the private data processing scenarios for distributed healthcare information system. The requirements and scenarios for the processing of private data are also presented in this chapter. **Chapter 5** focuses on purpose modelling. In this chapter, we provide the definition of purpose, purpose model and formal access control model for system that uses workflows (**C3**).

From the purpose model presented in **Chapter 5** and usage control requirements presented in **Chapter 4**, we propose a method to enforce purpose of use/access for privacy-aware policies for the system that uses workflows in **Chapter 6** (**C4**). In our approach, the access authorisation is based on the estimation of the level of certainty of purpose achievement, which is determined by purpose achievement prediction module. The prediction module is built using association rule learning method where user's access history and contextual information are used as the input data for rule analysis. We argue that by using the combination of contextual information and purpose achievement prediction, we can get a reliable purpose enforcement technique. In **Chapter 6**, we also discuss in detail the purpose achievement prediction algorithms and the access control model for controlling the resources assigned for each task of the workflow.

To support our purpose enforcement technique, we need a usage control platform. **Chapter 7** looks into the design of usage control system architecture supporting purpose enforcement (**C5**). The proposed architecture is designed to address the usage control of private data for a system that uses workflows. Furthermore, to validate

our purpose achievement prediction algorithms proposed in **Chapter 6**, we implement the usage control system in Java and we use datasets and challenges as the validation and assessment method (**C6**). In addition that, we also validate our solution in Privacy Preserving Perimeter Protection System (P5). P5 is the European research project and its main objectives is to protect personal data of individual generated by different surveillance tools (e.g. CCTV). We have contributed to the project ranging from the design of global system architecture to the design of access control model and their implementation. The solutions we invented are also used to address private data protection in P5. The works concerning P5 are presented in **Chapter 8**.

Finally, **Chapter 9** concludes our research. We also discuss in this chapter the remaining challenges, future work and our vision.

1.7 Publications Related to This Thesis

The research presented in this thesis reuses and extends publications of the author. We list below all the papers the author published so far.

Journal

1. Annanda Thavymony RATH and Jean-Noël Colin. Towards purpose enforcement model for privacy-aware usage control policy in distributed healthcare. Int. J. Secur. Netw. 8, 2 (Aug. 2013), 94-105. Volume 8, Issue 2, DOI: 10.1504/IJSN.2013.055943.

Conference

1. Tith Dara and Annanda Thavymony RATH. Prediction methods: Predicting access goal based on user's access history. The third international conference on inclusive innovation and innovative management, ICIIIM 2015, 25- 26 Nov 2015, Pathumthani, Thailand.

2. Annanda Thavymony RATH and Jean-Noël Colin. "Protecting Personal Data in Privacy Preserving Perimeter Protection System: From Legal to Technical Requirements and Implementation". Second Industrial Surveillance day, 12th IEEE International Conference on Advanced Video-and Signal-based Surveillance, August 25-28 2015, Karlsruhe Institute of Technology Fraunhofer IOSB, Karlsruhe, Germany. (Poster)

3. Annanda Thavymony RATH and Jean-Noël Colin. "Protecting Personal Data: Access Control for Privacy Preserving Perimeter Protection System". The 29th Annual International Federation for Information Processing (IFIP), WG 11.3 Working Conference on Data and Applications Security and Privacy, Fairfax,

VA, USA , July 13-15, 2015. This paper is also a part of a book (Chapter 16) published in Springer 2015. ISBN: 978-3-319-20809-1, Book ID: 340025_1_En. Acceptance rate: not given.

4. Annanda Thavymony RATH and Jean-Noël Colin. "Modelling and Expressing Purpose Validation Policy for Privacy-aware Usage Control in Distributed Environment". The 8th ACM IMCOM, International Conference on Ubiquitous Information Management and Communication, January 9-11 2014, Siem Reap, Cambodia. ISBN: 978-1-4503-2644-5, DOI:10.1145/2557977.2557991. Acceptance rate: 29%.

5. Annanda Thavymony RATH and Jean-Noël Colin. "Towards enforcement of purpose for privacy policy in distributed healthcare". The third IEEE International Conference on Consumer eHealth Platforms, Services and Applications (CeHPSA), IEEE CCNC 2013. Las Vegas, NV, USA. DOI: 10.1109/CCNC.2013.6488578, Print ISBN:978-1-4673-3131-9, publisher: IEEE. Acceptance rate: 30%

6. Annanda Thavymony RATH and Jean-Noël Colin. "Access and Usage Control requirements for Patient Controlled Record type of Healthcare Information System", International Conference on Health Informatics (HEALTHINF) 2013, Barcelona, Spain, pp. 331-336. Acceptance rate: 26%.

7. Annanda Thavymony RATH and Jean-Noël Colin. "A purpose model and policy enforcement engine for usage control in distributed healthcare information system", International Conference on Health Informatics (HEALTHINF) 2013, Barcelona, Spain, pp. 174-180. Acceptance rate: 26%

8. Annanda Thavymony RATH and Jean-Noël Colin. "Analogue attacks in e-health: Issues and Solutions". The second IEEE International Conference on Consumer eHealth Platforms, Services and Applications (CeHPSA), 2012. Las Vegas, NV, USA. Acceptance rate: 40% (not published).

9. Annanda Thavymony RATH and Jean-Noël Colin. "Patient Privacy Preservation: P-RBAC vs OrBAC in Patient Controlled Records Type of Centralised Healthcare Information System. Case study of Walloon Healthcare Network, Belgium". The Fourth International Conference on eHealth, Telemedicine, and Social Medicine, eTELEMED 2012. Valencia, Spain, pp. 111-118. ISBN: 978-1-61208-179-3. Acceptance rate: 30%

Workshop

1. Annanda Thavymony RATH and Jean-Noël Colin. "Modelling and Expressing Purpose Validation Policy for Privacy-aware Usage Control in Distributed Environment". 4e Atelier sur la Protection de la Vie Prive (APVP) (4th Workshop on the Protection of Privacy). Les Loges en Josas, 17-19 June 2013, France.

Doctoral Symposium

1. Annanda Thavymony RATH and Jean-Noël Colin. "Purpose Management and Enforcement for Sensitive Private Data in Open Environments", 10th International Workshop for Technical, Economic and Legal Aspects of Business Models for Virtual Goods incorporation with the 8th International ODRL (Open Digital Rights Language Initiative) Workshop and Working Group Meeting. 24-25 September 2012, Namur, Belgium.

2. Annanda Thavymony RATH and Jean-Noël Colin. "Applying DRM scheme for the protection of private data in privacy sensitive environment.", GRASCOMP's day, 3rd Nov 2011, Universit Libre de Bruxelles, Brussels, Belgium.

Chapter 2

Access and Usage Control Model - Rights and Access Control Policy Expression Languages

In this chapter, we present the access and usage control models and access control policy languages. The rest of this chapter is organised as follows. Section 2.1 talks about the access and usage control models. We also provide the definitions of policy, model and mechanism in this section. Section 2.2 mainly focuses on the standard rights and access control policy languages; we introduce both the general languages used to express different kind of policies for different types of digital content and dedicated languages that are designed for specific digital content. Section 2.3 is the summary.

2.1 Access and Usage Control

Two controlling steps are required to ensure that protected data goes to the right people and it is used in the right way: access and usage control. The main goal of access control is to selectively determine who can access resources and what access is provided exactly. Access control prevents unauthorised access to the resources of system and it is implemented as a result of certain access control requirements, which are generally in line with the institution's policies.

Access Control is about defining and enforcing the rules to ensure that only authorised users get access to resources in a system. Access control is concerned with determining the allowed activities of legitimate users, mediating every attempt by a user to access a resource in the system, *Vincent C. Hu et al [82].*

In some systems, access is granted after successful authentication of the user, but most systems require complex control. In addition to the authentication mechanism (e.g. username and password), access control is concerned with how authorisations are organised. In some cases, authorisation may reflex the structure of the organisation, while in others it may be based on the sensitivity level of documents.

In general, access control is a fairly good technique for centralised system where the data are processed within the boundary of a system. However, in the scope of distributed environment where the data are shared between different entities in different systems, the access control alone is not enough. Since to properly protect the data, we need to know, not only who can access data, but also what will happen to data once the access permission is granted. Therefore, usage control is required.

While access control concerns about who should or should not be allowed to access, usage control concerns about what should and should not happen to data item once the access permission is granted.

Usage control is a generalisation of access control that also addresses how data is used after it is released, *Alexander Pretschner [7].*

Usage control generalises access control by controlling not only who may access which data, but also how data may be used or distributed afterwards. In distributed settings, usage control is generally a controlling process at client or consumer side where data resides after access is granted. As presented in Figure 2.1, when data consumers request data from data provider, they have to commit themselves to an access and usage control. In general, access control happens at a time when data consumers initiate request at server side and usage control happens when they start processing data at client-side control domain. The dedicated usage control mechanism can provide data provider a sufficient amount of control over what data consumer can do when data is out of the controlling environment of server-side control domain.

2.1.1 Policies, Models and Mechanisms

When designing an access or usage control system, we should be consider three abstractions : policies, models and mechanisms [82]. Access control policies are high-level requirements that express how access is managed and who may access what information in which situation. While access control policies can be application-specific, policies are just as likely to relate to user actions within the context of an organisational unit or across organisational boundaries. The access control policies within a hospital may relate to privacy and skill (e.g., only cardiologist may prescribe medication for heart treatment). Even within a specific business domain, policy will vary from institution to institution. Moreover, access control policies are dynamic in nature, in that they normally change over time in response to ever-evolving business factors, government

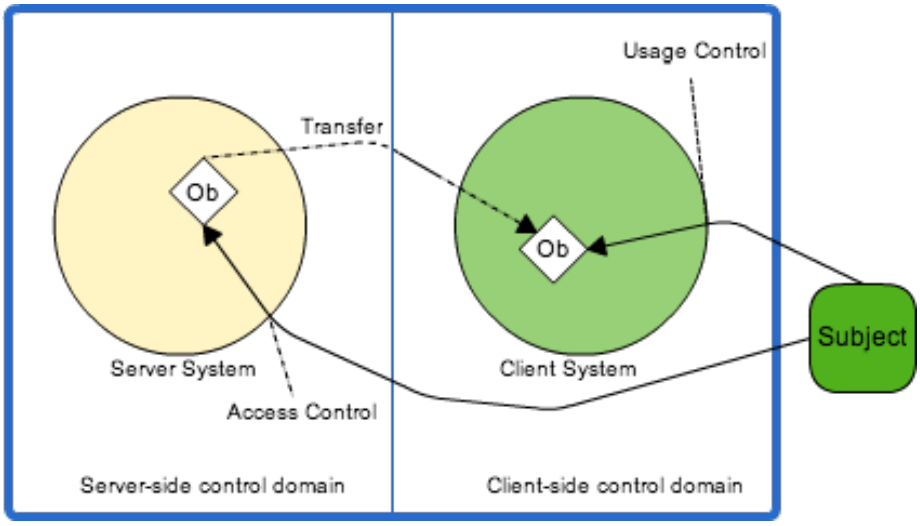

Figure 2.1: High level overview of access and usage control. (Ob: Object).

regulations or laws.

At a high level, access control policies are enforced through a mechanism that interprets a user's access request. There are different mechanisms; for example, a table lookup can be used to perform a grant or deny access. Although no standard yet exists for determining their policy support, some access control mechanisms are the results of the direct implementations of formal access control policy concepts [82].

A model is a formal presentation of the security policy enforced by the system. It bridges a gap in abstraction between policy and mechanism. Access control mechanisms can be designed to adhere to the properties of the model. Users see an access control model as a detailed expression of access control requirements. System developers see access control models as design and implementation requirements.

2.1.2 Access Control Models

In this section, we discuss the existing access control models. However, we detail only a few models that have been widely implemented in the systems, such operating systems or database management systems.

2.1.2.1 Discretionary Access Control (DAC)

Discretionary access control (DAC) [24], the restriction of access to objects is done based on the identity of subjects. In other words, DAC leaves a certain amount of access control to the discretion of the object's owner or anyone else who is authorised

to control the object. In DAC, it is the owner of the file who controls other users' accesses to the file. Only users specified by the owner may have some rights of read, write, execute, and other permissions to the file. One implementation of DAC is the access control list (ACL) [24], which has been used widely in operating-, networking- and database management system.

Although DAC policy is widely used, it is known to be relatively weak for two reasons. Firstly, granting read access is transitive; for example, when David grants Edward read access to a file, nothing stops Edward from copying the contents of David's file to an object that Edward controls. Edward may now grant any other user access to the copy of David's file without David's knowledge. Secondly, DAC policy is vulnerable to Trojan horse attacks; because programs inherit the identity of the invoking user. Thus, generally, the drawbacks of DAC are as follows:

1. Information can be copied from one object to another; hence, the assurance on the flow of information in a system is not possible.

2. No restrictions apply to the usage of information when the user has received it. Thus, it can't prevent information redistribution.

3. The rights for accessing objects are decided by the owner of the object, rather than through a policy that reflects the organisation's security requirements.

DAC is commonly discussed in contrast to mandatory access control (MAC). A system as a whole is said to have "discretionary" access control as a way of saying that the system lacks mandatory access control. On the other hand, systems can be said to implement both MAC and DAC simultaneously, where DAC refers to one category of access controls that subjects can transfer among each other, and MAC refers to a second category of access controls that enforces constraints upon the first.

2.1.2.2 Mandatory Access Control (MAC)

Mandatory access control (MAC) [24] means that access control decisions are made by a central authority, not by the individual owner of an object like DAC, and the owner cannot change access rights. In MAC model, system constrains the ability of a subject to access or perform some sort of actions to an object. Subjects and objects each carry a set of security attributes and when a subject makes an attempt to access an object, an authorisation rule controlled by the system examines these security attributes and then the decision can be made based on the defined authorisation rule. To determine if the operation on the object by a subject is allowed or not, those parameters will be tested and validated against the set of the authorisation rules made by the policy maker. MAC provides the central control of the security. Subject does not have rights to assign or override the access policy unlike DAC, which allows the subject to make decision or override the access policy. MAC supports more control level as both subject

and object carry the secured attributes that need to be checked or tested by the system for every access attempt.

The good example models that can be used to express MAC policy are Bell LaPadula Model (BLP) [90] and Biba [46]. The BLP, also called the multi-level model, was proposed by Bell and LaPadula for enforcing access control in government and military applications. In such applications, subjects and objects are generally partitioned into different security levels. A subject can only access objects at certain levels determined by his security level. The Biba integrity model was published in 1977 [46] at the Mitre Corporation. The Biba is created to address the integrity issue because BLP is able to address only the confidentiality but not data integrity. Data integrity relates to the accuracy and consistency of data over its entire life-cycle of data usage and it is an important aspect to the design, implementation and usage of any system which processes data.

The disadvantage of MAC exists in the complexity of the configuration, since for each resource (e.g. application or data) and subjects must be determined, which access authorisations are necessary. This tends to be very difficult for the system that works with the large number of users and resources. In addition to that, MAC can unnecessarily over classify data through the high-water mark principle and hurt productivity by limiting the ability to transfer information between systems and restricting user control over data.

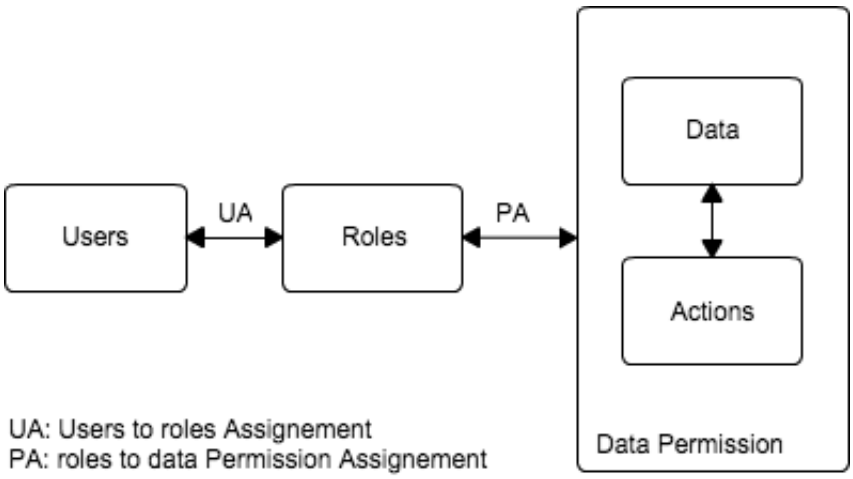

Figure 2.2: Basic RBAC model

2.1.2.3 Role-Based Access Control (RBAC)

In this section we discuss a family of RBAC model, one of the well-known access control models, implemented in wide range of information management system [24][20].

Basic RBAC model. In RBAC, access decisions are based on the roles that users have as part of an organisation. Users carry on assigned roles (e.g. doctor, cardiologist, etc.). Access rights are grouped by role name, and the use of resources is restricted to individuals authorised to the associated role. For example, within a hospital system, the role of cardiologist can include operations to perform a heart diagnosis, prescribe medication for heart treatment and heart surgery. The use of roles to control access can be an effective means for developing and enforcing complex enterprise-specific security policies and for facilitating the security management process.

Under RBAC, users are granted membership into roles based on their skills or profession and responsibilities in the organisation. The operations that a user is permitted to perform are based on the user's role. User membership into roles can be revoked and new memberships established as job assignments happen. Role associations can be established when new operations are created, and old operations can be withdrawn as organisational functions change and evolve by time. This facilitates the administration and management of rights; roles can be updated without updating the rights for every user on an individual basis. When a user is assigned to a role, the user can have no more rights than is necessary to perform the job.

Core RBAC consist of five administrative entities (see Figure 2.2): users, roles, permissions, operations and objects, where permissions consist of operations applied to objects. A role is a semantic construct around which access policy is formulated. Permissions are associated with roles, and users are made members of roles; hence, acquiring the roles' permissions. A single user can be assigned to one or more roles, and a single role can have one or more user members. This arrangement provides great flexibility and granularity of role to user as well as role to permission assignment.

In RBAC, there are three primary rules:

- Role assignment: a subject can exercise permission only if the subject has been assigned a role.
- Role authorisation: a subject's active role must be authorised for the subject. This rule ensures that users can have only roles for which they are authorised.
- Permission authorisation: a subject can exercise permission only if the permission is authorised for the subject's active role. This rule ensures that users can exercise only permissions for which they are authorised.

In RBAC, role engineering can be a complex task. The challenge of RBAC is the contention between strong security and easy administration. For strong security, it is

good for each role to be more granular, thus having many roles per user. For easy administration, it is better to have less number of roles to manage.

Given its weakness, the extensions of RBAC models have been introduced, such as Privacy-aware RBAC, Conditional RBAC, Constraint-based and Hierarchical RBAC.

RBAC's Extensions. There are number of extensions of basic RBAC model intending to solve different access control requirements for different systems environment. Below are some of the well-known RBAC extensions.

Privacy-aware Role-Based Access Control: P-RBAC [20] is an extension of the model RBAC. It provides complete support for expressing highly complex privacy policies. Its focus is to protect personally identifiable information and as such privacy-sensitive, taking into account characteristics such as purposes, conditions and obligations. P-RBAC extends the classical RBAC by adding three more privacy relevance entities such as obligations, conditions and purposes. In P-RBAC, data permissions are assigned to roles for a specific purpose. Conditions are the mechanisms to precisely define the authority over data to a specific role; using condition, we can express different access rights for user in the same role. For example, user in role "cardiologist" can access patient's heart record if and only if he has patient's consent. Patient's consent, in this case, is considered as the condition. Obligations are the necessary actions to be made before the actions on content can be exercised, for example, notifying data owner every access to data.

Hierarchical RBAC [24]: under RBAC, roles can have overlapping responsibilities and rights; that is, users belonging to different roles may need to perform common operations. Some general operations may be performed by all users. In this situation, it would be inefficient and administratively complex to specify repeatedly these general operations for each role that gets created. Role hierarchies can be used to represent the natural structure of an organisation. A role hierarchy defines roles that have unique attributes and that may contain other roles; that is, one role may implicitly include the permissions that are associated with another role. Role hierarchies are a natural way of organising roles to reflect authority, responsibility, and skill of a group of users.

Constraint-based RBAC [24]: in Constraint-based RBAC, organisations can put constraints on access either on subject, role or object. For example, constraints such as patient's consent or working-hour, can be placed on physician access so that only the related records that are associated patient are accessible for the physician while he is on duty. Another example concerning the access constraints put on data object, a healthcare provider may decide that the role of Cardiologist must be constrained to post only the results of certain tests concerning heart. In this example, type of data is used as constraint to limit the access to data for posting for user in role cardiologist.

2.1.2.4 Other Access Control Models

There are number of access control models that have been introduced in the research literature. However, they were designed for specific system requirements and they have not been implemented in any well-known information systems, unlike DAC, MAC or RBAC.

OrBAC [2][33] allows expressing a variety of security policies based on the concept of organisation. The main goal of OrBAC is to allow the policy designer to define a security policy independently from the deployment. The solution to fulfil this goal is the introduction of an abstract level in the model. OrBAC model is based on three principles: organisation, concrete and abstract level and context. Like other models, concrete authorisation in OrBAC relies on three entities, which are subject, action and object. Subject is an interactive entity, user or application that requests access on the organisation's object. Action is an operation on the object. Object is a resource requested by the subject. In OrBAC, a concrete authorisation is derived from abstract permission, which consists of three entities such as role, activity and view. Role represents a function or job title within the organisation. Activity groups actions into an abstract set and view is a set of objects. Typically, a subject in concrete level is mapped to a role in abstract level where an action is mapped to an activity and an object is mapped to a view. OrBAC has many advantages, in addition to its ability to express the permission; it can also express a mixed policy with permissions, prohibitions and obligations. With OrBAC, security policies could take into account delegation, hierarchy and context.

Attribute-based Access Control (ABAC) [88] is an access control model where the access right is decided based on a set of attributes associated with data or subject. Each attribute is a distinct discrete and possibly unrelated field. The access authorisation is based on the comparison of the attributes value presented by user to the predefined values in the system. Actually, the attributes that are used as the access control parameters may come from different sources, such as temporal attributes, spatial attributes, attributes related to data or subject. The good example of ABAC implementation is the eXtensible Access Control Markup Language (XACML) [8].

Group-Based Access Control (GBAC) [16], the access permission is granted based on a concept of group and all users under the assigned group can exercise the same level of right. Conceptually, GBAC is similar to RBAC. The only difference is that in RBAC, role is a semantic construct around which access policy is formulated while GBAC uses group as a semantic construct around which access policy is formulated. In GBAC, users in the same group do not necessary have the same role.

History-Based Access Control (HBAC) [24] is another example of access control model that access permission is based on past access of user. However, this access control model has the drawback that since new users do not have past access for background check, it needs a special control every first access attempt of new user. HBAC

is generally used with other access control model to achieve a better security for highly sensitive data.

Relation-Based Access Control (RelBAC) [24], the access permission in RelBAC is based on the relationship between data owner and data subject. In RelBAC, relationship between data owner and data requester is a semantic construct around which the access permission is formulated. In other words, any users who have relationship with data owner can access data. A good example of such access control model is the Facebook's "friends" and "friends of friends" concept. Anyone who is in relationship as "friends" to the poster can have a right to see the post and perform some allowed operations.

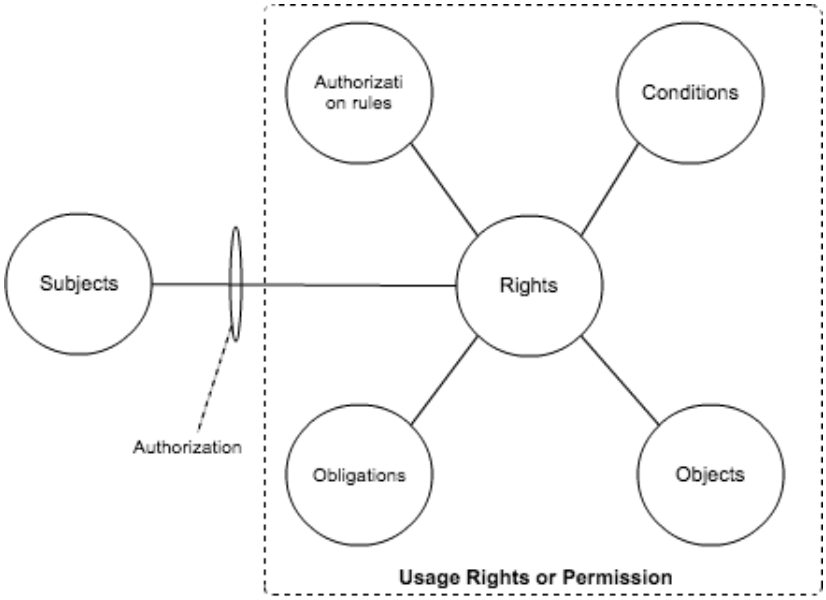

Figure 2.3: Basic UCON model

2.1.3 Usage Control Models

Usage Control (UCON), proposed by Jaehong Park and Ravi Sandhu [41], is a model that encompasses traditional access control, trust management and digital rights management and goes beyond them in its definition and scope. UCON enables fine-grained control over usage of digital objects than that of traditional access control model where it considers only a priori control of data. UCON model consists of six components (see Figure 2.3): subjects, rights, objects, conditions, authorisation rules and obligations. Subjects are those who request access to data. Rights are the authorised action for subjects to perform on objects. Objects refer to the digital objects that subjects want to

access. Obligations are requirements (e.g. payment before listening a song) that have to be fulfilled by subjects. Conditions are subject and object environmental constraints (e.g. using data between 8AM to 5PM). Authorisation rules are the rules applied on the rights of subject. In today highly dynamic, distributed environment, obligation and conditions are also crucial decision factors for richer and finer controls on usage of digital resources. UCON is designed to support the complex and fine-grain usage control on digital resources. But UCON is a general usage control model and it is not designed to specifically deal with private sensitive data. Thus, to make it suitable to express privacy policy requires a "purpose expression". We propose an extension. The details of it is presented in Chapter 4 (Section 4.6).

2.2 Rights and Access Control Policy Expression Languages

In previous section, we discussed the access and usage control models. In this section, we mainly focus on rights and access control policies languages, the existing standard languages, which can be used to formally express the access and usage control policies derived from the models we presented in Section 2.1.

2.2.1 What is a Rights Expression Language?

A Rights Expression Language (REL) [69] is a machine-readable language used in Digital Rights Management (DRM) [71] systems that supports different aspects of the digital access environment (e.g. licensing, payment, access and usage control). Most RELs are expressible in XML format and embedded in form of metadata with digital contents (e.g. video, song or eBook). However, some other formats are also used, such RDF (Resource Description Framework) [69], which is embedded into HTML file for web services. With the growth of the DRM technologies, many RELs are developed ranging from a relatively simple expression of rights holders' preferences such as Creative Commons (CC) [69] to a highly complex expression for the secured and trusted system environment, such as Open Digital Right Language (ODRL) [56] or a more complex access control and authorisation language like XACML.

2.2.2 Goal of Rights Expression Language

In general, REL can be used for the following purposes: 1) statement of legal copyright, 2) expression of contractual language and 3) implementation of control.

1. Statement of Copyright

The copyright law is a statement about ownership of intellectual works and the rights of various parties, in particular the creators or the owners of the works. In general, the copyright is attached with the agreement or contract stating the limited actions that can be performed on the property. Creative Commons (CC) is an example of REL that falls into this category.

2. Expression of Contract

In addition to copyright, the rights holder can extend copy and distribution rights through the mechanism of contracts and licenses. These agreements can give more rights to users of the copyrighted material than would be covered by copyright law. Contracts are regarded as the agreements between an agent and specific individuals. They can contain conditions and requirements that the parties agree on. ODRL is a good example of the type of REL that falls into this category.

3. Control on digital content

Both copyright law and contracts cannot provide any actual control over the behaviour of users on contents after giving access. Instead, they rely on the parties to act within the stated agreement mentioned in contract. Because digital contents must be mediated through software and hardware for use, it is possible to exercise a priori control over access to and use of the contents through that technology. The nature of the control may or may not also be expressed in a human-readable user license. There are two key points in controlling the digital contents. The first one is to control the access on the contents. It refers to the permission of the access and it is nothing to do after access is granted. For example a permission to download the file, in this case after the file is downloaded, we are no longer able to control it, user can copy or share file with other users. The second one is the usage control that refers to the phenomenon after the access is granted, in this case, the contents are periodically controlled during the allowed usage session.

2.2.3 Standard Data Elements in Rights Expression Languages

REL is generally made up of resources, agents, rights, constraints and conditions. Although these elements are considered nearly universal, the use of these standard elements may vary from one to another rights language since different rights languages may have different degrees of development based on their immediate and intended uses.

Agent represents the party or parties to the contract or license that the REL expresses. Most languages use a fairly general agent data element that can represent any number of different roles in the environment of the REL.

Resources is the targeted object for the parties. Resource in REL is described in form of metadata and the link indicated where the resource is located. The resource can be a digital or non-digital format.

Rights expresses the allowed actions on resources (e.g. read, write, copy or transfer).

Constraints are the key elements that are used to put more restriction on the permitted actions. Constraints can be anything that can logically be applied to the action, but tend to be quantitative elements in actionable RELs (e.g. time, geographical region, number of usage, etc.).

Conditions. In addition to constraints, which generally limit the rights assigned to user, there may be specific conditions that must be fulfilled before user can exercise their rights, the most common of which is payment.

2.2.4 Rights Expression Languages

In this section we introduce in brief several languages that are widely used in commercial digital content protection technologies. Details of the state of the art of rights expression language can be found in our technical report [69].

2.2.4.1 Open Digital Rights Language (ODRL)

The Open Digital Right Language (ODRL) [56][19], created in 2000 by Renato Ianella of IPR Labs in Australia, is a standardised W3C language to express the rights information over digital content. The ODRL aims at providing the flexible and interoperable mechanism to support the use of the digital resources in publishing and distributing of the electronic publication, digital images, learning object, computer software and other digital forms. Figure 2.4 shows the core model of ODRL 2.0 consisting of central entity (policy) interconnecting with other entities such as permission, prohibition, duty, party constraint and action.

Policy is a set of rules indicating the permissions or prohibitions for a user to perform actions on assets. ODRL 2.0 provides a number of possibilities for policy expressions and can be used in the following usage scenarios: “Agreement” is a form of contract stipulating all terms of usage and the parties involved. “Offer and request” express the terms of usage. “Privacy” expresses the terms of usage over the personal data. “Ticket” expresses the terms of usage by any party who holds the ticket in their possession.

Permission expresses the allowed rights on assets (e.g. permission to play a movie).

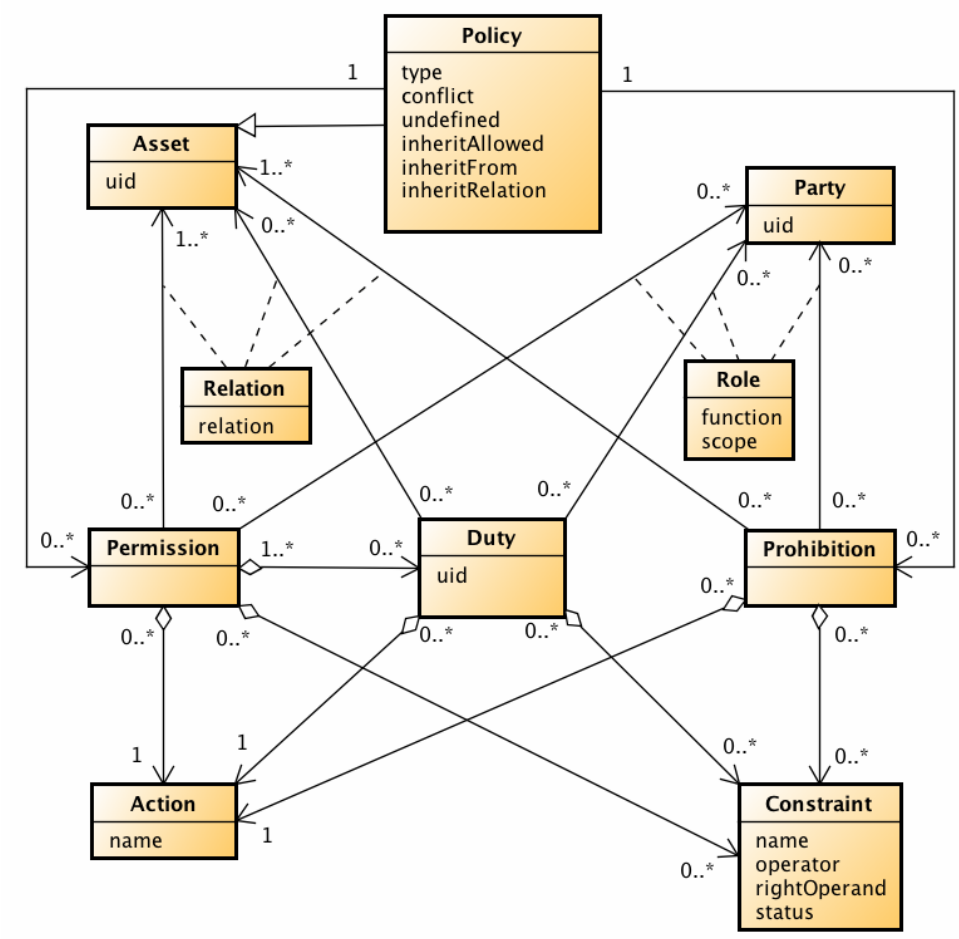

Figure 2.4: ODRL 2.0 Core Model, this figure is brought from [56].

Prohibition expresses the prohibited actions on asset (e.g. prohibiting user from transferring data to third party).

Action expresses the allowed execution rights on asset (e.g. action "read", "write" and "print").

Duty expresses action that user (assignee) needs to fulfil before the permission or rights can be granted or exercised (e.g. payment duty).

Constraint allows policy makers to create a fine grain control for complex policy. Constraint puts more restriction on action performed on asset. For example, user can access the assets when he is in a specific location or uses a specific device to access (spatial constraint).

Relation is an associated class that is used to link the asset to permission, duty and prohibition. This entity consists of an attribute called relation that describes the relationship between the asset and the linked entities.

Role is an associated class that is used to link a party to permission, duty and prohibition. It indicates which role the party takes in respect to those entities, for instance, a role as consumer, or owner of content. In ODRL, a requester has a role as "assignee" and the content owner has a role as "assigner", these are defined in the Common Vocabulary Specification of ODRL 2.0 [56].

2.2.4.2 Other Rights Expression Language

eXtensible Rights Markup Language (XrML) [69] is the XML-based rights language for specifying rights and conditions to control the access to digital contents and services. XrML has its roots in Xerox Palo Alto Research Centre and it was first introduced in 1999. XrML was not actually built from the ground; it was derived from Digital Property Rights Language (DPRL) introduced in 1996, which became XrML when the meta-language was changed from a lisp-style meta-language [69] to XML.

Using XrML, content owner can determine (1) the parties allowed to use (2) those resources, (3) the rights available to those parties and (4) the terms and conditions under which those rights may be exercised. These four elements are the core of the language. Since its creation, XrML has evolved through industry feedback, critical review and product implementation. In late 2003, XrML was adopted by MPEG community to be used in MPEG-21 REL. In early 2001, Creative Commons [69] had settled on the approach of creating machine-readable licenses based on the World Wide Web Consortiums, which is part of the W3C Semantic Web Activity. In 2002 the first machine-readable licensed was unveiled and Creative Commons recommended that publishers use the RDF syntax to express license properties in HTML.

2.2.5 Access Control Authorisation Languages

Access Control Authorisation Languages is the standard language designed to express security policies and access control to data in the system. We introduce two languages: XACML and EPAL, since both languages are well-known and being used in many applications[1]. Moreover, we have studied XACML and EPAL in great detail and some of our works published in [14] relate to them. For more details of the state of the art of access control authorisation languages, one can find it in our technical report [69].

2.2.5.1 eXtensible Access Control Markup Language (XACML)

XACML [87], XML-based language, is OASIS[2] standard describing policy languages as well as access control decision and response languages. The policy language is used to describe the general access control requirements while the access control decision request/response languages are used to form a query to ask whether or not a given action should be granted or denied.

XACML Policy Model

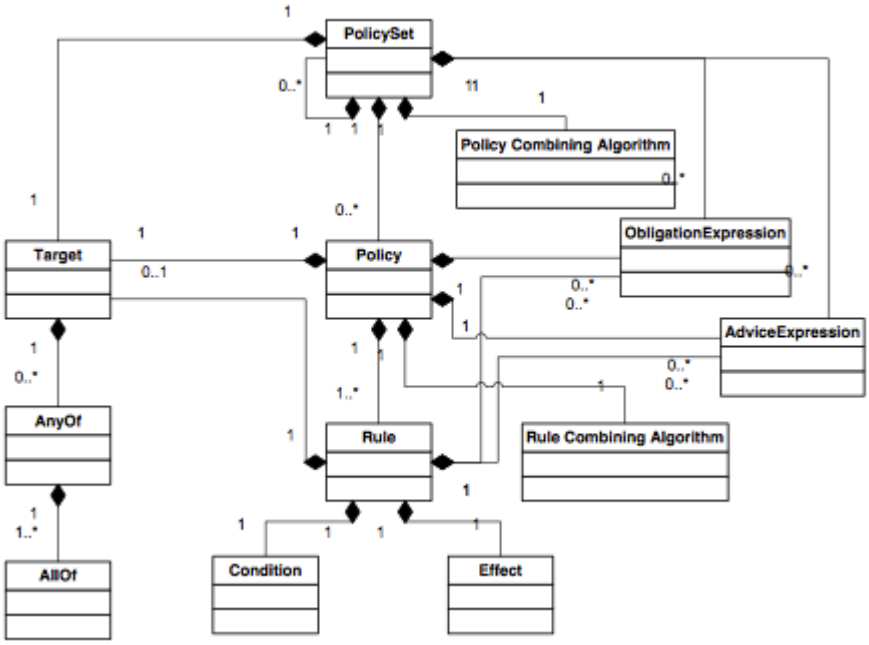

Figure 2.5: XACML Policy Model, this figure is brought from [87].

[1] http://xacmlinfo.org/2013/12/06/xacml-based-access-control-for-web-application/
[2] https://www.oasis-open.org

The XACML policy model consists of three mains components: policy set, policy and rule. Figure 2.5 shows the structure of the XACML policy.

Rule is the most elementary unit of policy. It defines the action that the party can perform with the specific condition and the obligation that need to be fulfilled. In XACML, rule is encapsulated in a policy. Rule consists of the following elements.

- Rule target defines a set of requests to which the rule is intended to apply. The condition element may further refine the applicability established by the target. For example, in role-based model, user to role assignment information can be defined in target. This means that the rule applies to certain roles with certain users. The target needs to be checked before going to the detail policy validation, if the elements in the target are not valid, further validation is ignored.
- "Effect" indicates the rule-maker's intention of a "True" evaluation for the rule. There are two possible values for effect: "Permit" and "Deny". "Permit" means access permission is granted while "Deny" means otherwise.
- "Subject" represents a person or a group of people authorised to access resource.
- "Resource" represents the digital object that subject can access.
- "Action" is an allowed operation on resource.
- "Condition" represents a boolean expression that refines the applicability of the rule beyond the predicates (required information) implied by its target. However, it may be absent.
- "Obligation" indicates the duty that user needs to fulfil after or before access permission is granted.
- "Advice" is similar to obligation. However, advice may be safely ignored while obligation is all time compulsory.

Policy combines the rules to form a set of rules. Policy consists of four main elements: policy target, rule-combining algorithm, rules and obligation and advice.

- Policy target, similar to the rule target, specifies the set of requests to which it applies.
- Rule-combining algorithm specifies the procedure for combining all the results of the rules. For example, "deny-override" is a function that returns "Deny response" if one of the rules in the policy returns negative response.
- Obligation or advice in policy has similar function to that of rule.

Policy Set groups policies together. It consists of four elements: target, policy-combining algorithm, policies and obligation and advice. Policy set's target has the same functionality to that of rule and policy target.

- The policy-combining algorithm specifies the procedure for combining all the results of the policies.
- Obligation and advice in policy has similar function to that of rule and policy. The difference is that they apply to a set of policies.

It is worth noting for the details of XACML policy expression and example, one can find them in Chapter 4, Section 4.7.

2.2.5.2 Enterprise Privacy Authorisation Language (EPAL)

Since EPAL is designed for expressing access authorisation for private data and in the early stage of our research we put significant effort on it, we introduce it briefly in this section. EPAL [27], developed by IBM research group, is a language for exchanging privacy policy in a structured format between applications or enterprises. It is a formal language for writing enterprise privacy policies to control data handling practices in IT systems according to fine-grained positive and negative authorisation rights. It focus on the core privacy authorisation while abstracting data models and user-authentication from all deployment details such as data model or user-authentication.

EPAL Privacy Policy Model, EPAL policy (Figure 2.6) defines lists of hierarchies of data-categories, user-categories, purposes, actions, obligations and conditions. User-categories are the entities that use collected data. Data-categories define different categories of collected data that are handled differently from a privacy perspective (e.g. medical-records vs. contact-address). Purposes model the intended service for which data are used. Actions model how the data are used (e.g. transfer vs. read). Obligations define actions that must be taken by the environment of EPAL (e.g. delete after 30 days). Conditions are boolean expressions that evaluate the context around user and action.

The above-mentioned elements are then used to formulate privacy authorisation rules that allow or deny actions on data-categories by user-categories for certain purposes under certain conditions while enforcing certain obligations. It is worth noting that EPAL only defines how each data-category is handled. It does not handle the enforcement of policy. The enforcement of policy (e.g. implementation of obligation) is the role of the application maker to develop according to their requirements. The detailed EPAL profile [1] and policy expression for purpose validation policy, that we

[1]EPAL profile is the outline or description of EPAL's functionalities and components in its core model.

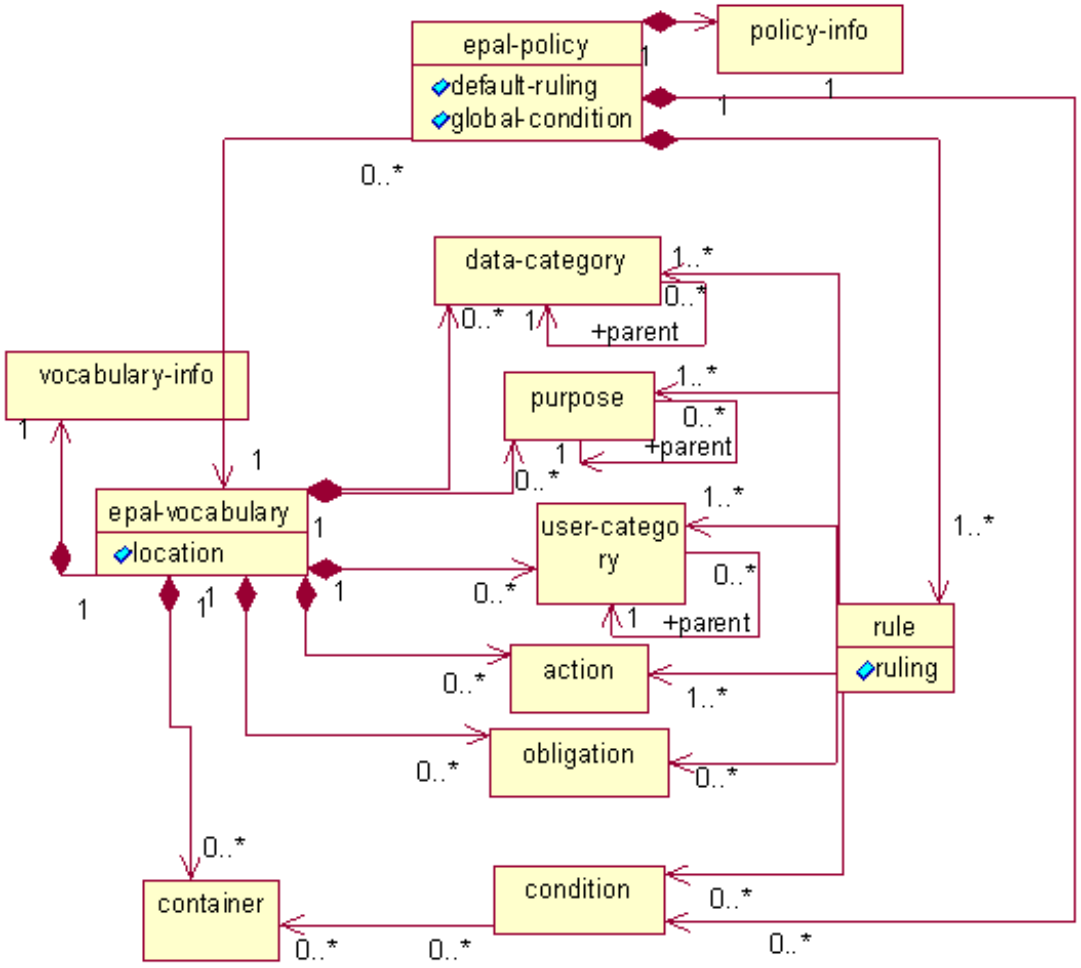

Figure 2.6: High-level UML Overview of an EPAL policy [27].

published in the 8th ACM International Conference on Ubiquitous Information Management and Communication, can be found in [14].

2.3 Summary

In this chapter, we discussed different access and usage control models, such as DAC, MAC, RBAC and other models. The advantages and disadvantages of them were also presented. Furthermore, we also discussed different rights and access control languages. We presented some well-known languages such as ODRL, EPAL and XACML. However, we focused largely on two languages: XACML and EPAL. According to our study [69], we find that XACML is very suitable to be used for expressing complex privacy-aware policies (see Chapter 4 for more details). XACML goes across many reviews, implementations and regular updates. XACML research group has developed Java-enterprise-XACML[1], a Java-based engine that is able to validate XACML policies. Java-enterprise-XACML is extensible and can be adjusted to our requirements (see Chapter 4). EPAL is another good candidate and it is also designed to address the access authorisation for private data. However, EPAL provides only the language for expressing the privacy-aware access authorisation policies. Unlike XACML, EPAL does not have policy validation engine, and anyone wants to use this language needs to build their own engine. This is one of EPAL limitations. Moreover, EPAL does not provide the solution for policy enforcement; it is the role of the developer to create their own policy enforcement engine. Taking into accounts all the factors, we decided to use XACML for our implementation. We have done a survey on different rights expression languages. The detailed technical report can be found in [69].

It is worth noting that the information, concerning access and usage control models and policy expression languages, provided in this chapter will be used as the knowledges for access and usage control model selection, which will be presented in Chapter 4. We compare each model to the requirements for processing private data we defined in Chapter 4 and find out which model meets the defined requirements.

[1] https://code.google.com/p/enterprise-java-xacml/

Chapter 3

Usage Control Techniques and Technologies

In this chapter we present the state of the art of the existing techniques in usage control. We also provide the comparison of different usage control techniques used in different system environments and point out which technique is appropriate to which environment. We also discuss the existing usage control technologies such as Digital Rights Management. The rest of this chapter is organised as follows. Section 3.1 provides the definition of the usage control and enforcement techniques. Section 3.2 describes the general usage control and enforcement techniques. Section 3.3 presents the existing Digital Rights Management technologies both for commercial and open source systems. Section 4.3 is the summary of this chapter.

3.1 Usage Control Techniques and Technologies

A usage control technique [70] is a method or procedure used to enforce a usage control policy. The usage control techniques are classified into two different types: less restrictive and highly restrictive. Less restrictive technique refers to any technique allowing to observe the misuse of digital content, but not be able to prevent the misuse. This technique, of course, discourages user from performing illegal act, but not being able to prevent user from redistributing or misusing digital content. For example, the use of watermark on digital content provides a means for content provider to trace the original source of the illegal distribution, but watermark cannot be used to prevent user from sharing or redistributing content. The second class, the highly restrictive technique, refers to any technique allowing content owner to protect and control content usage at any point in time during the usage session. In this technique, only an authorised user can perform the authorised action on content. The use of sticky policy and logging in combination with trusted client-side application [73], to support the usage control of

content, is a good example of restrictive usage control enforcement technique.

Usage control technology [70], the application of usage control, is the collection of techniques used in controlling the usage of digital content. Usage control technology can be embedded in application softwares, machines, computers or devices, which can be operated by individuals without detailed knowledge of the workings of such things. Digital Rights Management (DRM) technology is one of the types of usage control technology. For example, Windows Media DRM is embedded into Window Media Player for controlling the usage rights on windows media contents. Window Media DRM uses encryption and license as the techniques to control and enforce the usage of video or audio content played on Window Media Player.

3.2 Usage Control and Enforcement Techniques

An enforcement Technique is a method used to enforce the usage restriction on content either directly or indirectly. It provides a means to protect, secure, trace or detect fraud. We present below some techniques used in the existing DRM technologies.

3.2.1 Watermarking and Steganography

Digital contents and documents are flying through cyberspace to consumers. Unfortunately, along the way, individuals may choose to intervene and take those contents for themselves. Digital watermarking and steganography technology can discourage this practice by limiting or eliminating the ability of third party to redistribute the content that he has taken.

Watermarking [22] is the process of embedding information into digital contents (e.g. audio, pictures, video or text) in a way such that it is difficult to remove. Watermark can be used to trace the original source of content leakage. For example, when consumer requests to view digital content, it can be watermarked with the identity of consumer, in case, consumer shares content to third party or beyond, the embedded information is a source used to trace the illegal distribution.

Steganography [22] is the practice of concealing a file, message, image, or video within another file, message, image, or video. Generally, the hidden messages will appear to be (or be part of) something else: images, articles, shopping lists, or some other cover text. For example, the hidden message may be in invisible ink between the visible lines of a private letter.

The advantage of steganography over cryptography alone is that the intended secret message does not attract attention to itself as an object of scrutiny. Visible encrypted messages arouse interest, and may in themselves be incriminating in countries where encryption is banned. Thus, whereas cryptography provides the means to protect the

contents of a message, steganography is about concealing the fact that a secret message is being sent.

3.2.2 Encryption

Encryption [63] is used to protect digital content when it is shared or moved out of the system coverage. With encrypted content, user can only access content if he has valid decryption key. This ensures that content goes to the right person. However, this technique has weaknesses when key is compromised, content can be shared without any control. Encryption is generally used in conjunction with other techniques to make tighter security in controlling the usage of content. For example, the combination of encryption and watermark, encryption protects content from unauthorised user while watermark allows tracing illegal content usage.

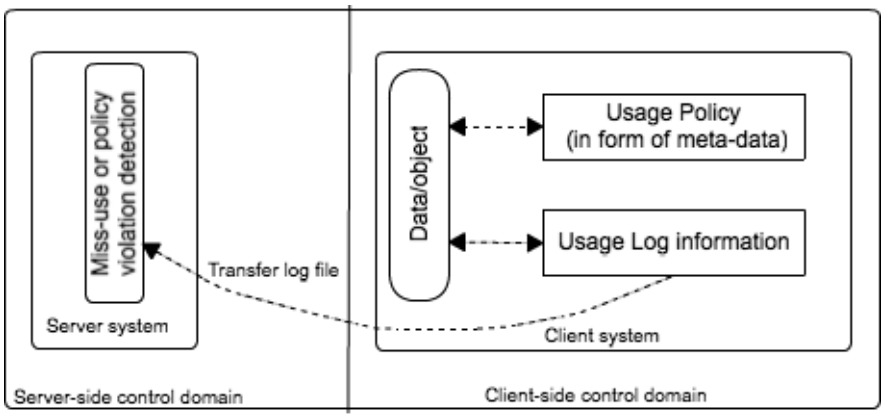

Figure 3.1: High level overview of sticky policy and usage log for usage control enforcement.

3.2.3 Policy-based with the Support of Secured Client-side Application

This technique requires content to be attached with a usage policy. When user requests to use content, an application at client side triggers usage policy validation and decides if usage can be granted or not. This technique requires a secure and trusted client application that ensures the data integrity and accountability. Policy-based technique [70] provides a fine-grain control on content usage.

3.2.4 Fingerprint or Digital Signature

Fingerprint [55] refers to the characterisation of the content based on its representation (signals or features). Fingerprinting is used to protect content against the alteration by the third party. Fingerprinting is different from watermarking in that Fingerprinting is a persistent output associated with content rather than an embedded entity. It also does not alter content. A hashing function is used in the fingerprinting process. For example, before being sent to consumer, a content is hashed and the hashing result is attached with content sent to consumer. Upon receiving content and hashing result, the application at consumer side hashes the received-content and then compares the hashing result received from content provider and the hashing result at consumer side, if the results are matched, it shows that content has not been altered. In general, fingerprint is used in conjunction with other techniques to provide a tighter security in protecting and controlling the use of digital content. This technique alone is primarily used to solve the tamper resistant problem (or content alteration problem).

3.2.5 Usage Logging and Notification

Usage logging [70] is a method used to capture the usage information when consumer gets access to content. It records all the necessary information before, during and after the use of content. Then they are used to validate the content usage. Usage log can be stored at client application or attached with content and goes along with it depending on system specification. Attaching usage log with content provides an opportunity to trace the circulation of content in the network.

Notification refers to the act of informing system every access to digital content. The notification message consists of necessary information used to analyse the current state of content usage (e.g. location, time of access or executed actions). The notification can occur before, while using or after the content usage. Notification provides necessary information for the server and based on that information it can make a future decision whether to keep granting or to revoke the access to the content.

3.3 Digital Rights Management Technologies

DRM [71] is used to protect high-value digital content. DRM technologies are developed to serve different business models [70] with different level of security requirements. Some are device and platform dependent and some are designed to be interoperable among devices and platforms. Digital Rights Management is generally complex and extensible; it supports a diversity of devices, users, platforms, media types, content types and a variety of system requirements. This section provides an overview of DRM system and technologies.

3.3.1 System Overview

DRM aims at supporting legal distribution of digital content while protecting appropriate property rights. DRM has two important aspects: digital rights management and digital rights enforcement [48].

- **Rights Management:** the rights holders have to be able to manage and specify the terms and conditions of content usage [36].
- **Rights Enforcement:** to ensure that content is only used as stipulated in the terms and conditions associated with it [36].

These two aspects form basic security for digital content distribution. The core concept of DRM is the use of digital license. Through digital licensing, content provider can have more control over what consumer can do over content. The digital content and its license can be attached or stored separately. Storing separately can provide a flexible way for content providers to freely distribute the protected content among users and license requests can take place later.

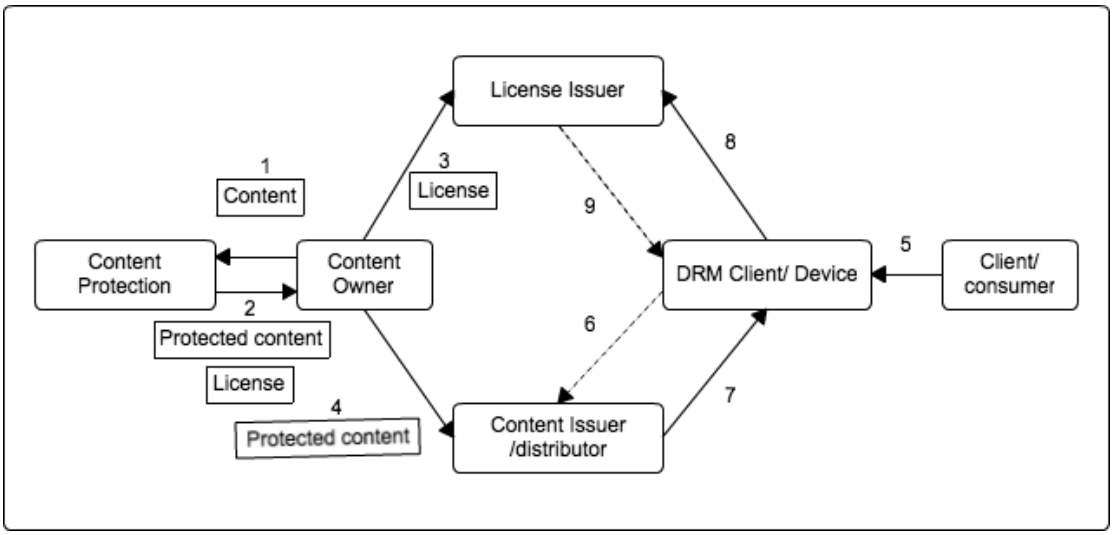

Figure 3.2: A typical DRM model, showing the principal components forming DRM system and the interaction between client, content owner, and DRM components. This figure is drawn by author based on the figure in [36]. Numbers in the figure represent the processing order.

3.3.2 Typical DRM Model

A typical DRM model (see Figure 3.2) provides a complete view of DRM system and also the interaction processes between consumer, DRM system components, and content owner. The system consists of the following components.

- **Content Protection** can be implemented in any way such that it provides content owner a way to upload and configure the usage rights of content in the DRM system. In this phase, content is protected using data protection techniques, such as encryption, watermark or fingerprint. The security protection during this phase is important as the leakage of content related information can destroy the whole process.
- **Distributor** is the place where submitted content resides. Once content is provided, it is stored in a secure environment in content repository. Content can be stored in a plain or encrypted format depending on DRM specification. Note that content issuer issues content through a particular preference media, it is not limited to the Internet. It can be other kind of distribution channels (e.g. wire-network, physical media (CD, DVD, USB), Mobile or PC).
- **License Issuer** is a place where license is securely stored. It issues licence whenever there is request from DRM client.
- **DRM Client** refers to the trusted hardware or software, which is a proxy to user (consumer). It is trusted in the sense that it would not allow the unauthorised user to access content. It also reinforces the terms and conditions of usage. DRM Client is responsible for initiating license as well as content request when there is request from consumer.
- **Content Owner** is an entity providing digital contents.
- **Client** is a physical person consuming digital content by retrieving downloadable or streaming content through distribution channel.

Figure 3.2: the interactive process between DRM's components.

Content owner sends content to "content protection module" (1), then "content protection module" shields content with an appropriate security mechanism and returns protected content with its associated license (2). After getting the protected content and its associated license, "content owner" sends license to "license issuer" (3) and protected content to "content issuer" (4). Protected content and its associated license will be sent out when there is request from DRM client. Client requests content through DRM client (5). DRM client initiates request to "content issuer" (6) and "content

issuer" returns the requested content (7). DRM client initiates request to "license issuer" (8) and "license issuer" returns the license (9). After getting the two objects (e.g. content and its associated license) DRM client performs access authorisation process.

3.3.3 DRM Technologies

This section presents the existing DRM technologies that have been designed for different purposes and used in different systems environment and business models.

3.3.3.1 Windows Media DRM (WMDRM)

WMDRM [86], developed by Microsoft, provides the protection of audio and video content. Its solution is based on Windows Media Player and server. WMDRM's architecture is not fixed; it offers a set of software development tool kits that allow using the core DRM services and combining them into various configurations. WMDRM supports some media formats, such as Advanced Streaming Format (ASF), Windows Media Audio (WMA) format and Windows Media Video (WMV) format. WMDRM uses MPEG-21 REL [40] to express license on content. The main advantage of WMDRM is that Windows media format is widely used in the Internet and the Windows media player has incorporated DRM support. This makes WMDRM popular and adopted in many digital content distribution markets. For example, PressPlay [64], a large online music service, adopts WMDRM technology to offer the digital music from Sony and Universal studio. Although WMDRM has many advantages, it also has some drawback concerning the control on usage, it does not have the ability to trace the illegal distribution because of the absence of the watermark and other usage enforcement techniques.

3.3.3.2 Open Mobile Alliance (OMA)

OMA [15] focuses on DRM for mobile devices for digital contents such ringtones, songs, screensavers and backgrounds. In June 2004, OMA released a DRM Enabler, which includes three types of functionality: Forward lock, combined delivery and separate delivery. **Forward lock:** content is packaged in a special container format called DRM message. Content is not encrypted and stays in the receiving device after reception; content can be consumed without limitation. However, content is strictly attached with device and it cannot be shared. This type of enabler can be used with low value content such as daily newspaper in mobile service. **Combined Delivery:** this enabler is an extension of the forward lock. In combined delivery, DRM message is divided into two parts: the rights information and content. When the DRM message reaches the client side, the rights object is extracted from the DRM message and used to

monitor the usage of content. **Separate Delivery:** this enabler is derived from the combined delivery. While in combined delivery content and rights object are packaged in DRM message; in separated-delivery, content and rights objects are packaged and sent separately to client's device. With separate delivery, the content can be reused at client's device.

The security in OMA depends on one part of the security measure in rights issuer and content distributor and other part on the DRM agent. In OMA, content is packaged in DRM content format container. It is encrypted using a symmetric content encryption key (CEK) and signed by content issuer. The security problem in data protection of OMA is the encryption key. The key is content specific that can pose the risk if the device that uses content is compromised.

3.3.3.3 MPEG-21

MPEG-21 [40] is an open standardised multimedia framework used in wide range of networks and devices. The main concept of the MPEG-21 is the digital item that system needs to distribute. Each item has it own digital identification, known as DII (Digital Item Identification), and the declaration information, known as DID (Digital Item Declaration). DII specifies a term and concept describing the relation between items and DII specifies the unique identification of the item. In MPEG-21, digital items are grouped and put in a single package, known as container. MPEG-21 can support and work in many multimedia file types and platforms. It uses MPEG-21 REL to express license on digital item. The security in MPEG-21 depends on the implementation, one part of the user component and other part on the core system of MPEG-21. The core system will ensure the secure delivery of the content while user part ensures the safe use of content. MPEG-21 does not specify the DRM client component architecture, it considers DRM client as the independent component that user can develop separately.

3.3.3.4 Adobe PDF Merchant/ Web Buy

Adobe PDF Merchant [3] is a server-based DRM technology that provides protection to PDF content. It consists of two components: Adobe Merchant DRM that acts as DRM server and Web Buy, that is integrated in Acrobat Reader, acts as client DRM. Adobe PDF security depends solely on the encryption and authentication techniques that allow protecting content when transferring it from server to client DRM and when content resides at client's device. Adobe PDF Merchant has also integrated watermarking technique into their system that makes content to be traceable. This DRM technology constraints with both, platform and file types, it is available only for Windows and Macintosh platform and only PDF file is supported.

3.3.3.5 Fairplay

FairPlay [39], developed by Apple, is used in QuickTime and the well-known iTunes music store. FairPlay is considered a less restrictive DRM technology as it allows songs to be copied to any number of the iPod devices and played on up to five authorised devices. FairPlay's digital item packed with MPEG-21 container. The encryption uses AES asymmetric key with MD5 hashing function. FairPlay's architecture and their security protection techniques are not available in public, making it difficult to discuss in details, however, the basic functionality can be described as following. FairPlay allows user to copy the digital content from its store and listen in 5 different devices. User needs to register their device and the number of permitted device reduces whenever user registers a new device. User may use the license attached with content and view the content offline if the license is valid, otherwise DRM client needs to contact server to acquire new license. FairPlay supports traceability and it is also a file type and platform dependent system.

3.3.3.6 Other DRMs

Light Weight DRM (LWDRM) [25] provides less restriction on content usage. It allows consumer to use content freely except the mass distribution (super distribution). This feature provides LWDRM to be an application of fair-use principal [1]. LWDRM uses two format files: Local Media File (LMF) and Signed Media File (SMF). An LMF file is bound to a single device by a hardware driven key but can be converted into SMF-format, which can be played in different devices that support LWDRM. In LWDRM, the identification of the owner is embedded in the content and if such file is found in the mass distributed, it is allowed to find the source and the prosecution of the illegal distribution can be followed.

High Bandwidth Content Protection (HDCP) DRM [26], developed by Intel Subsidiary Digital Content Protection, is a DRM technology standard that provides a protection method for streaming encrypted audiovisual content between devices. With HDCP, content provider can set a policy on content preventing content from being stored, distributed or displayed on non-HDCP compliance devices.

AXMEDIS [77] is an open source interoperable DRM technology for the automatic production and distribution of cross-media content over number of different distribution channels such as networked PC, PDA, Mobile phone or I-TV. AXMEDIS is designed to interact with web services. In AXMEDIS, the interoperability is achieved by introducing a translation module into DRM component architecture. The module functions as the right expression translator, which is responsible for the translation of one DRM license to another.

[1] http://en.wikipedia.org/wiki/Fair_use

DRMs	Content types	File types	RELS	Fraud Detection	Distribution models	Utilization	Platforms	Interoperability across different DRMs
WMDRM	Audio &Video	AFS/WMA	MPEG-21	No	B2C, SupD, B2B	yes	Windows	No
OMA	All	ATRAC3	ODRL	N/A	B2C, SupD, B2B	yes	All	No
FairPlay	All	m4p	N/A	N/A	B2C	yes	Mac &Windows	No
Abode PDF Merchant	Doc	pdf	N/A	Yes	SupD, B2B, B2C	yes	Mac &Windows	No
MPEG-21	Audio &Video	mp21/ m21	MPEG-21 Rel	N/A	B2C, B2B	yes	All	No
LWDRM	Audio &Video	LMF/SMF	N/A	Yes	B2C	yes	Windows, Mac &Linux	No
CORAL	ALL	*	#	#	#	no	All	Yes
EMMS	ALL	All	N/A	No	B2B, B2C	yes	Windows, Mac &Linux	No
OpenSDRM	*	*	ODRL or MPEG-21	#	#	yes	Window, Mac&Linux	Yes
Chillout	*	*	MPEG-21	#	#	yes	Window, Mac&Linux	Yes
AXMEDIS	*	*	MPEG-21	#	#	yes	Window, Mac&Linux	Yes

*: Depend on original DRM content format
#: Depend on original DRM specification
Distribution Models:
SupD: Super-distribution
B2B: Business to business
B2C: Business to Client

Figure 3.3: Functionalities comparison between different DRM technologies. For more details, refer to [71].

Chillout [78] is an interoperable and software-based DRM technology. Most of its components are based on MPEG-21. However, it also has its own distinctive feature, which is the introduction of Domain Management Device (DMD) component. This component is responsible for controlling all the devices used in acquiring the digital content. Chillout is developed in Java platform, it can support different content formats and run on different platforms (e.g. Windows, Mac and Linux).

OpenSDRM [79] is developed to address the possibility of adaptation to several business models and different types of digital content, aiming at enabling business involving multimedia content to function, by enforcing licensing agreements for content usage and offering business opportunities to the content owners and content providers. OpenSDRM deals with the rights management and not directly with the copy protection. OpenSDRM's architecture is developed based on some existing standard specification such as MPEG-4.

InterTrust [80] offers a solution for content packaging, distribution and rights management based on a packager program and rights server technology. This technology supports, rentals, pay-per-use and try-before-buy business model. InterTrust provides DRM client for varieties of devices such as PCs, mobile phone, and music players. InterTrust also provides a development tool kit that allows content provider to integrate InterTrust DRM into their service and products.

VideoGuard [49] provides end-to-end protection of an operator's service, leveraging the unique security capabilities of each individual platform for home TV networking services. VideoGuard is designed to help TV operators to extend their pay-TV services to connected media devices. It enables the secure ingestion, delivery and consumption of premium content over networks while maintaining subscription privileges across devices. VideoGuard solution is also based on the encryption and the management of the authentication key. For more details on DRM technologies including the functionalities comparison, one can look at our technical report of DRM technologies at [71].

3.4 Summary

In this chapter we addressed two important points: usage control enforcement techniques and DRM technologies. We presented the techniques used to enforce usage control on data. Concerning DRM, we presented a system overview and a standard model of DRM systems. Then, a list of existing DRM technologies and their functionalities comparison was provided (see Figure 3.3). Based on our study on the state of

the art of DRM technology [71], we concluded that the existing Digital Rights Management (DRM) technologies cannot provide the security we need as required by laws [28] for processing of private data. This is because the existing DRM technologies are not specifically built for private data. They are built to protect commercial contents (e.g. multimedia contents); they are content-specific and lack of generalness [66][71]. This rules out the possibility of using DRMs, without complement or extra support functionalities to support the processing of private data in distributed environment.

Concerning the survey of DRM technologies [71], we focused on the functionalities and security option provided by those technologies, then we compared what could be provided by those technologies with the legal and technical requirements of private data processing (see Chapter 4). So far we are not aware of a complete solution (DRM technology) that designs specifically for managing and enforcing privacy-aware usage control policies in distributed healthcare, especially, the systems that use workflows. Consequently, it would be best to design a dedicated system for distributed healthcare, in such a way that addresses the requirements [10] in such system. Although the existing DRM technologies could not be used to support the processing environment of private data, the DRM scheme can be used as a model for controlling and enforcing the private data in distributed environment. However, in order to support the security requirements for processing of private data, we need to design a dedicated usage control enforcement technique that is able to address those security requirements [10] (see also Chapter 4). To build the dedicated usage control system, we need to have an effective usage control enforcement technique [71], we conclude that among those techniques, policy-based with support of trusted client application (see Section 3.2.2) is the most appropriate that can be used to control private data in distributed environment since it provides more levels of control and granular policy expression; policy-based is flexible and can be used to control complex usage control requirements of private data (see Chapter 4).

Chapter 4

Scenario and Requirements for Managing the Processing of Private Data

In this chapter, we discuss the scenario and requirements for managing the processing of private data. We start with the introduction of the scenario in healthcare, a model for the distributed processing of data. Then, we derive the requirements for managing such data, including legal, contractual and technical requirements. Furthermore, we compare the existing access and usage control models (presented in Chapter 2) against those requirements. Finally, we propose a usage control model taking into account the privacy aspect and the requirements for processing private data. The rest of this chapter is organised as follows. We introduce a usage control of private data scenario in Section 4.1 and discuss the involved parties and the details of the processes with processing rules in Section 4.1.1 and 4.1.2, respectively. A model for distributed processing of data is presented in Section 4.2. The legal and contractual aspects of private data processing are discussed in Section 4.3. In Section 4.4, we analyse usage control requirements for processing of private data (e.g. distributed healthcare), and based on those requirements we define usage control model. In this section, we also present the comparison between different access and usage control models based on our defined requirements. Section 4.5 talks about the access and usage control model selection. We introduce privacy-aware UCON in Section 4.6 and REL profile for access and usage control model in Section 4.7. Finally, Section 4.8 is the summary.

4.1 Scenario - Distributed Processes in Health Care

Managing the processing of private data in distributed environment is required as depicted by the following scenario. It is worth noting that our scenario is inspired by

the Wallonie Healthcare Information System [54][85] while the hospital management process is inspired by the work of Jean Herveg and Anne Rousseau [44] on management process in hospitals in Belgium.

The scenario presents an informal distributed healthcare system where two hospitals have cooperation and agree on sharing patient's health records when needed. Each hospital has its own usage policy and they agree that each hospital needs to respect its defined usage policy of the sharing information when using them.

The scenario depicts the need to control the use of patient's health records when they are moved from one to another hospital. The scenario shows that it is necessary to restrict the processing of the data by means of policies (rules) expressing permissions and obligations, and each time before releasing the data, system needs to ensure that user requesting to use data is the one who really has the rights to do so and for the purpose he claims. Thus, predicting the purpose of using data declared by user before releasing data is really important because it is a way to prevent mal-intended user from unnecessarily accessing protected data with a fault-claim.

4.1.1 Involved Parties

We restate the informal private data processing scenario in distributed healthcare system, presented in Chapter 1. We suppose that two hospitals, one in Paris, France (Broussais) another in Namur, Belgium (CHR-Namur), have signed a cooperation agreement on sharing their patients' medical records. Under the agreement both hospitals can share the medical records when needed. The processing of patient's medical records must be strictly controlled and must comply with the policies defined by data owner. For example, if Broussais processes the medical records belonging to CHR-Namur, Broussais must fully respect the data usage policy defined by CHR-Namur.

A Belgium citizen, named Edward, has registered in CHR-Namur hospital for the heart treatment. All the medical records concerning his heart are managed by CHR-Namur under the heart treatment purpose. This means, CHR-Namur can share his medical records for such purpose and only for his treatment. Some time later, when Edward visits Paris, he faces a heart disorder and he needs an emergency treatment (surgery). Edward is hosted at Broussais hospital in Paris. Before performing heart surgery, the cardiologist needs his past heart medical records for pre-surgery examination. The cardiologist acquires those medical records from CHR-Namur under CHR-Namur-Broussais agreement. The medical records are transferred to Broussais and they stay there for a limit period of time, to be precise, during the treatment of Edward. During that period, cardiologist can examine Edward's medical records given that the usage policies defined by CHR-Namur are respected. With this informal example, we have three parties involved in the process. The CHR-Namur, Broussais and Edward.

- **CHR-Namur** is the name of a hospital in Namur, Belgium. It is also a part

of Wallonie Healthcare Network Information System. We use it in our informal example as one involved party in our proposed healthcare information system.

- **Broussais** is another involved party in our informal example.
- **Edward** is a name of a patient who has registered himself in CHR-Namur.

4.1.2 Process

For our considerations we need a detailed overview of the processing of health records as well as the usage policies issued by the hospital. In our informal health records processing scenario, defined in Figure 4.1, the processing of health records is based on the work of Jean Herveg and Anne Rousseau [44] and policies are based on the policies defined in RSW information system [85]. Figure 4.1 depicts the processes that need to be executed for a treatment of patient in emergency situation, the action-by-action run-through of the processing and the data that are required for each process. The steps depicted in the figure are:

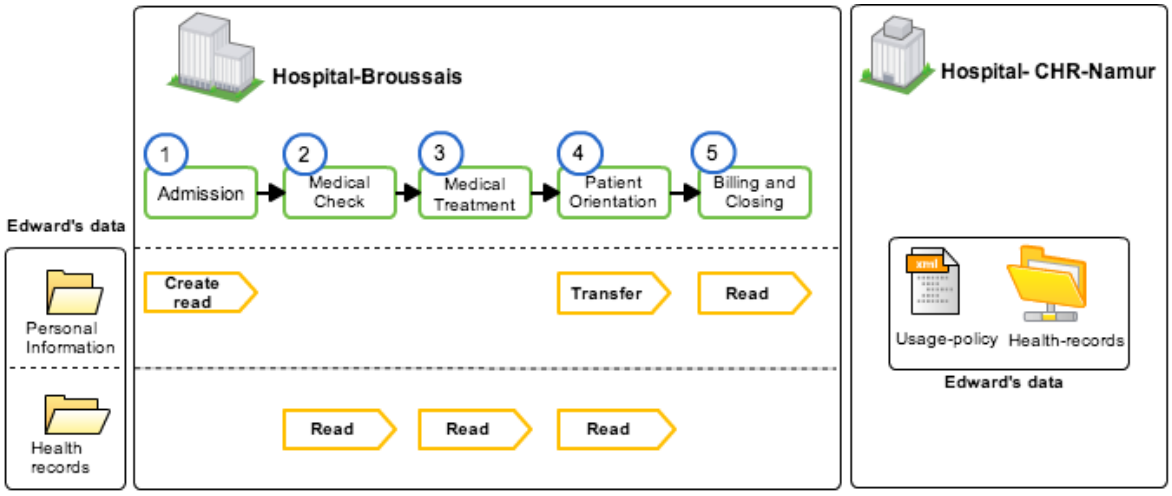

Figure 4.1: Process Overview of patient's medical treatment scenario. In the scenario, Broussais is taking care a patient (Edward) who has previously registered at CHR-Namur. Broussais needs some health records from CHR-Namur for Edward's heart treatment purpose.

1. **Admission**: with the admission of Edward as a patient of Broussais Hospital a new health record is created by the hospital. However, in order to create new record for Edward, hospital staff may need to have some personal information

of Edward, such as home address, contact information, insurance information, etc. Admission is the first processing step of the workflow. At the beginning the health record is empty. The create action is logged by the hospital. As part of the admission the personal information of Edward is collected and stored in the record. The record is stored on a file server of the hospital and is only accessible by the hospital.

2. **Medical Check**: after admitting to the hospital, the next step of the process is to perform the medical check for Edward. To do medical check, hospital staff may need to access (read) some required health records of Edward, and as the medical records of Edward are stored at CHR-Namur, Broussais's staff has to send a request to CHR-Namur for permission. Then, if the permission is granted by CHR-Namur, the requested data and its usage policy are transferred to Broussais and stored temporarily in Edward newly created record at Broussais.

3. **Medical Treatment**: after performing medical check, the next step is to diagnose the patient. Patient diagnostic consists of number of sub processes where medical treatment is one of them. Other processes, such as additional tests (e.g. blood test, radio,...), expert advices, ..., are the supplementary processes that need to be done if required. The detailed overview of the processing of patient diagnostic is depicted in Figure 4.2. The data required for medical treatment vary according to illness and it requires sub-processes (workflow) to handle it. Figure 4.1 is the treatment process for heart surgery operation.

4. **Patient Orientation**: after medical treatment, patient orientation is needed. There are two cases. Firstly, patient can be temporarily hospitalised. Secondly, if there is something serious and the hospital is not capable to handle it, the patient needs to be transferred to other hospital. In case of transfer, the patient's personal data may also need to be transferred and health records may also need to be reused or re-accessed.

5. **Billing and Closing**: it is the final process that patient needs to clear when the medical treatment is completed. The information required for this step is the personal information of patient, such as contact address, insurance information and other information used for payment if needed.

In Figure 4.1, we define other components as follows.

- Personal Information: we refer to the information related to individual patient. That information is something like name, address, contact information, insurance information, etc.

- Health Records of Patient: they are the data related to health history of patient. Something like the records of illness (e.g. heart-attack, diabetes, blood's type,...)

- Usage Policy: it is the policy defined to control the usage of health records when they are at remote client.

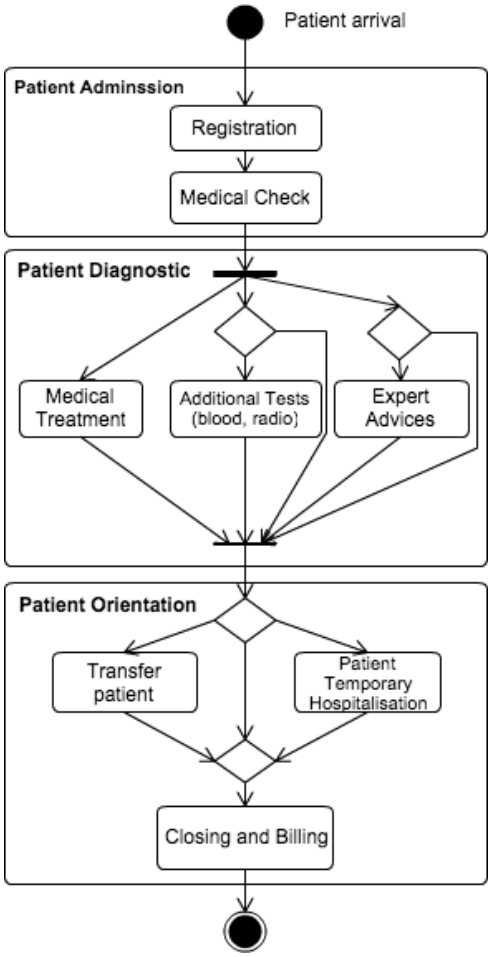

Figure 4.2: Flowchart, example of the general processing steps for patient treatment in case of emergency service, Jean Herveg and Anne Rousseau [44] .

4.2 A Model for Processing Data: Physical Execution

Physical execution [72] is the execution of the data by the physical entities in the workflow. It may involve, application, services, database management system, or Web application. For example, in a Web environment, data is processed by Web services as

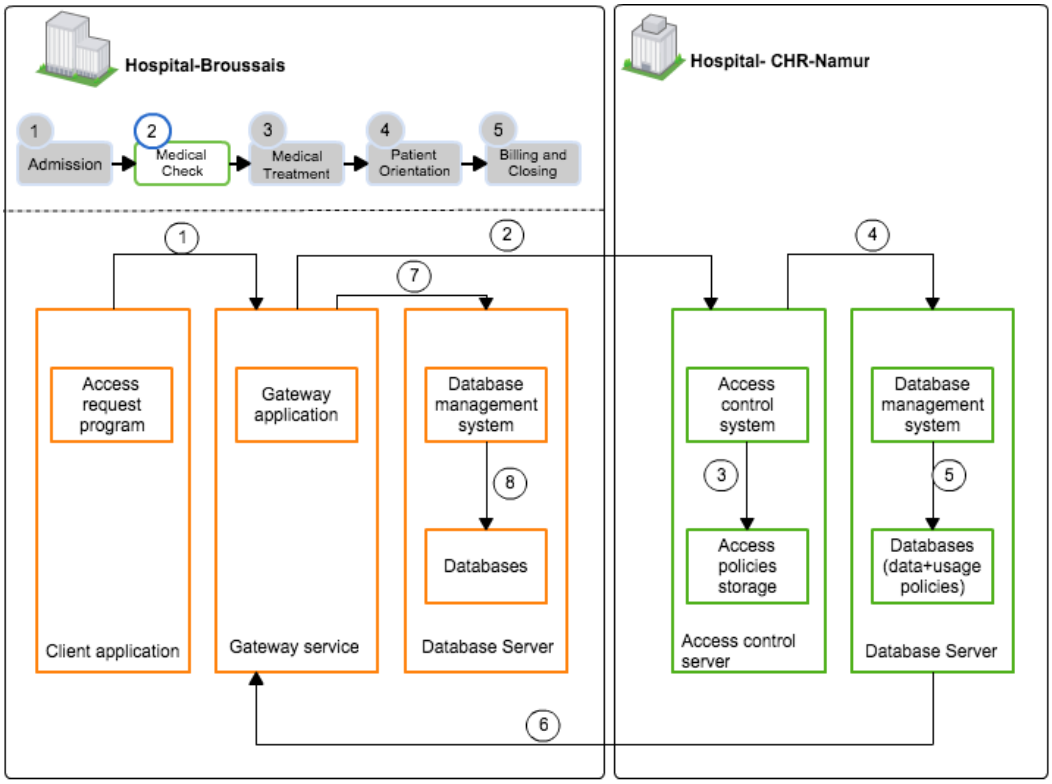

Figure 4.3.1: An example of physical execution of a request to transfer Edward's health records from CHR-Namur.

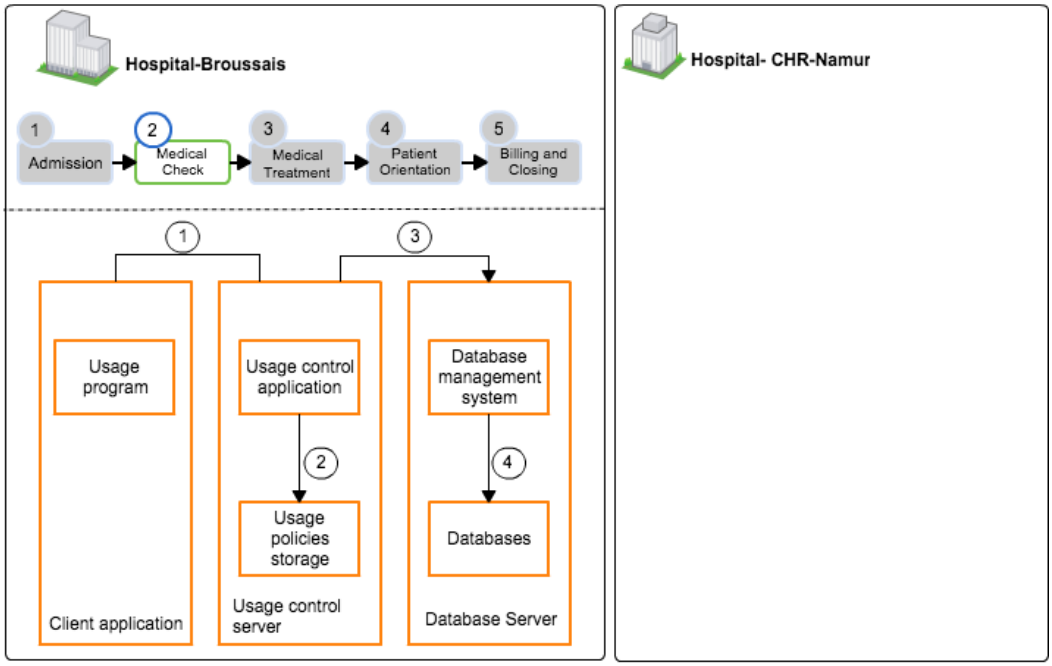

Figure 4.3.2: An example of physical execution of a request to use Edward's health records, which are recently stored at Broussais.

Figure 4.3: An example of physical execution of a request for transfer and a request to use Edward's health records, which are originally stored at CHR-Namur.

well as other Web applications (e.g. Web forms). As part of the workflow execution, entities (services, applications, etc.) communicate and exchange data with other entities. In this section, we present the physical execution of the data for two different access scenarios: request to transfer data from hospital in Namur to Broussais hospital in Paris and the request to use data by user at hospital in Broussais when the data is temporarily stored at Broussais's system. The physical execution model, shown in Figure 4.3, is based on the process overview of patient's medical treatment presented in Figure 4.1.

Description of Physical Execution in Figure 4.3

Figure 4.3 depicts the physical execution of health record by different components of the system. The figure shows the technical details of the snippet of the health record processing steps. Figure 4.3 has two figures: Figure 4.3.1 and Figure 4.3.2. Figure 4.3.1 depicts the processes of transferring health record and usage control policy from Namur hospital to Broussais hospital. Figure 4.3.1 depicts the processes of controlling the usage of health record when it is temporarily stored at Broussais's system.

Figure 4.3.1, the process starts during "medical check" processing step where Edward's past health records are required to complete the medical check procedure. With this reason, user (e.g. physician) who is responsible for this processing step needs to access to Edward's health records. To get the records which are currently stored at CHR-Namur, they need to create an access request using "client application". The client application forwards the request to gateway service (call (1)). The gateway service is responsible for providing secure communication between Broussais and CHR-Namur system. After receiving an access request from client application, gateway service forwards the request to "access control server" situated in CHR-Namur (call (2)). Then, "access control server" validates the request based on the applicable access control policies in server storage (call (3)). If the access permission is granted, access control server contacts database management system (call (4)) to retrieve the records as well as its corresponding usage control policy (call (5)). The records and usage policy are then sent to gateway service (call (6)). Finally, gateway service forwards the records and usage policy to database server (call (7)) and the records and usage policy are stored in database (call (8)).

Once Edward's health records are transferred from CHR-Namur and temporarily stored at Broussais, user at Broussais can use them. Figure 4.3.2 depicts the physical execution of a request to use the records at Broussais.

First, by using client application for controlling the usage of records, user can create a request to use the records. Then, client application sends the request to usage control server (call (1)); usage control application, which is a part of usage control server, validates the request based on the usage control policy available in storage (call (2)).

After validating the request and if the request is granted, usage control system contacts database management system (call (3)) to retrieve the related records (call (4)).

4.3 Requirements for the Use of Private Data

Since private data is protected by laws, managing the processing of such data requires the consideration of various legal, contractual, organisational and technical aspects. For example, in the member states of the European Union, there are privacy laws, implementing the EU Directive 95/46/EC [28], that limit the processing of sensitive private data. Other law, such as Safe Harbor Framework [59], intends to manage the processing of private data between the European Union, the United States and Switzerland. Laws may cover only the general issue; hence, further rules for the usage and protection of data are defined in contractual agreements concluded between the involved parties. In this section, we discuss mainly about the legal requirements for processing the private data in distributed environment. Mining requirements is important because it is the first step before we can define a usage control model to be used in such system.

4.3.1 Legal Requirements

Different countries have different privacy laws for regulating the processing of private data. In Europe, Directive 95/46/EC are designed to control the processing of private data between different countries in Europe. The laws specify rules for handling personal data in general and privacy sensitive data in particular. The directives define objectives for the legislation of the member states of the European Union and it is binding on the Member States as to the result to be achieved but leaves them the choice of the form and method they adopt to realise the community objectives within the framework of their internal legal order. All EU Member States need to implant the EU Directive into national legislation.

We have chosen the EU Directive 95/46/EC as starting point to derive legal requirements for the distributed processing of private data. EU directive defines the legal responsibility for data processor when processing private data and also the legal rights for data subject. It also defines the procedure and regulation for data processor when processing private data of individual. The Directive uses the terms "controller", "data subject" and "processing". Article 2 defines the terms in the context of the directive:

"Controller" shall mean the natural or legal person, public authority, agency of any other body which alone or jointly with others determines

the purposes and means of the processing of personal data, ...

"Data subject" is an identified or identifiable natural person, ...

"Processing" shall mean any operation or set of operations [...] whether or not by automatic means, such as collection, [...] storage, adaption or alteration, use, disclosure by transmission,....

Article 10 specifies the obligations of data controller when processing personal data. Article 10 lists the following information that data controller must provide to data subject.

- (a) the identity of the controller and of his representative, if any;
- (b) the purpose of the processing for which the data is intended;
- (c) any further information, such as the recipients or categories of recipients of the data, ...

Article 10 and Article 11 describe the need for an information service attending to the information rights of private persons. Such an information service can support the process of notifying the data subject. In summary, from those articles, there are three main requirements:

- Data controller needs to notify data subject when using data.
- Processing of private data is limited to the purpose for which data is intended; excessive use is not allowed.
- Consent from data subject is required when processing personal data.

4.3.2 Contractual Requirements

The distribution of data does occur between different organisations. Contracts are the agreements or rules for inter-organisational data processing and they are concluded between the involved organisations and they differ depending on various aspects, such as the involved organisations and the binding laws.

Contracts are not often publicly accessible. Only the general terms and conditions of open available services are accessible. The contracts vary from organisation to organisation, but through binding laws there is a common foundation. For example the agreement of inter-organisational private data processing is built upon the privacy law. Thus, we apply the legal aspects discussed before to the distributed processing of private data and define the contractual requirements in the following section.

4.3.3 Requirements for Processing Private Data

To manage the processing of private data, we need to derive the requirements for processing such data. For the legal aspects, we can derive requirements from the Articles 10 and 11 of the EU Directive. For contractual aspect we derive from the Wallonie Healthcare Network [54] and the work in [72]. It is worth noting that there are two types of requirements listed below. The first set is the data processing requirements. It describes the requirements that data processor need to do in order to comply with laws [28]. What type of information should data processor provide to data subject when processing private data. The security and confidentiality that data processor needs to provide to data subject. The second set is the requirements for usage control policy expression. It describe what should be in the policy and how the policy should be expressed. It is important to note that these requirements are general and it can be applied to any private data processing environment (EU only) since they are the products of our study of EU Directive.

1. **Requirements for data processor**. These are the duties that data processor should do when processing private data.

   "**Identifiability**": the information concerning the processing of private data provided to data subject must be sufficient to identify the entities processing the data as well as the recipients and sources of personal data. The difference between non-distributed and distributed environments is that not only one entity processes the data but many. The data processor must provide a complete list of all involved entities and their clear identification.

   "**Accessibility**": the data processor must enable the data subject to access information about the complete processing of his personal data at any time.

   "**Completeness**": the data processor must inform about which personal data item is processed, how it is processed and why it is processed. If requested, the given answer must be exhaustive. It must cover all processing steps performed on the data. The answer must contain details of each processing step and its ***intended purpose*** (e.g. research, marketing, etc.).

   "Confidentiality": the data processor must ensure that only the data controller and data subject have access to information.

2. **Policy expression and implementation requirements.** These are the requirements for policy expression and implementation.

"Controllability": the policy mechanism must apply the required policies to anybody accessing the associated data. The concerned entity must check whether a performed action complies with policies.

"Availability": the information required by policy conditions must be available even if confidential parts of the data and of the source of information stay hidden.

"Expressiveness": policies must allow for conditions and obligation that data accessor needs to fulfil. One must be able to depend permissions and restrictions on the history of the processing of data.

"Obligations": policies must allow for specifying obligations and to decide whether all active obligations can be met in the future. To comply with contracts, entities must be able to fulfil future obligations. For example, an obligation that data controller must notify data subject on every access or the obligation that the system needs to delete all the data temporarily stored in local device once the usage permission is expired.

"Level of Granularity for policy expression": the policy language and the model we choose must be able to express in detail the complex privacy policies designed for private data (e.g. the performed actions, their actors, their purposes and other contextual information).

The usage control requirements presented in the following section (see Section 4.4) derive from the private data processing requirements discussed in this section.

4.4 Usage Control Requirements

In Section 4.3, we present the general requirements for processing private sensitive data. In this section, we address particularly, the usage control requirements for distributed healthcare information system. We have published our finding in 7th International Conference on Health Informatics [10]. The proposed requirements are the result from our study of Wallonie Healthcare Network information system [54] and EU Directive 95/46/EC (based on the work in Section 4.3.1). Furthermore, based on the identified requirements, we propose a usage control model, which extends the traditional UCON model [41]. We name it privacy-aware UCON. It is worth noting that we propose the requirements in such a way so that they can be used to express the usage control policy for private data taking into account the legal requirements presented in Section

4.3.1. For example, in Figure 4.4, the data subject notification requirement is suggested in response to the legal requirement in EU Directive (Article 10) that requires data processor to notify data subject when data is used while the data subject's consent requirement is suggested in response to the article 11 of the EU Directive that requires data subject's consent before processing private data. The detail explanation of each requirement is presented below. It is worth noting that the data collected for requirements analysis are from the Wallonie Healthcare Network information system and the study of EU Directive. There are two main requirements: Usage restrictions

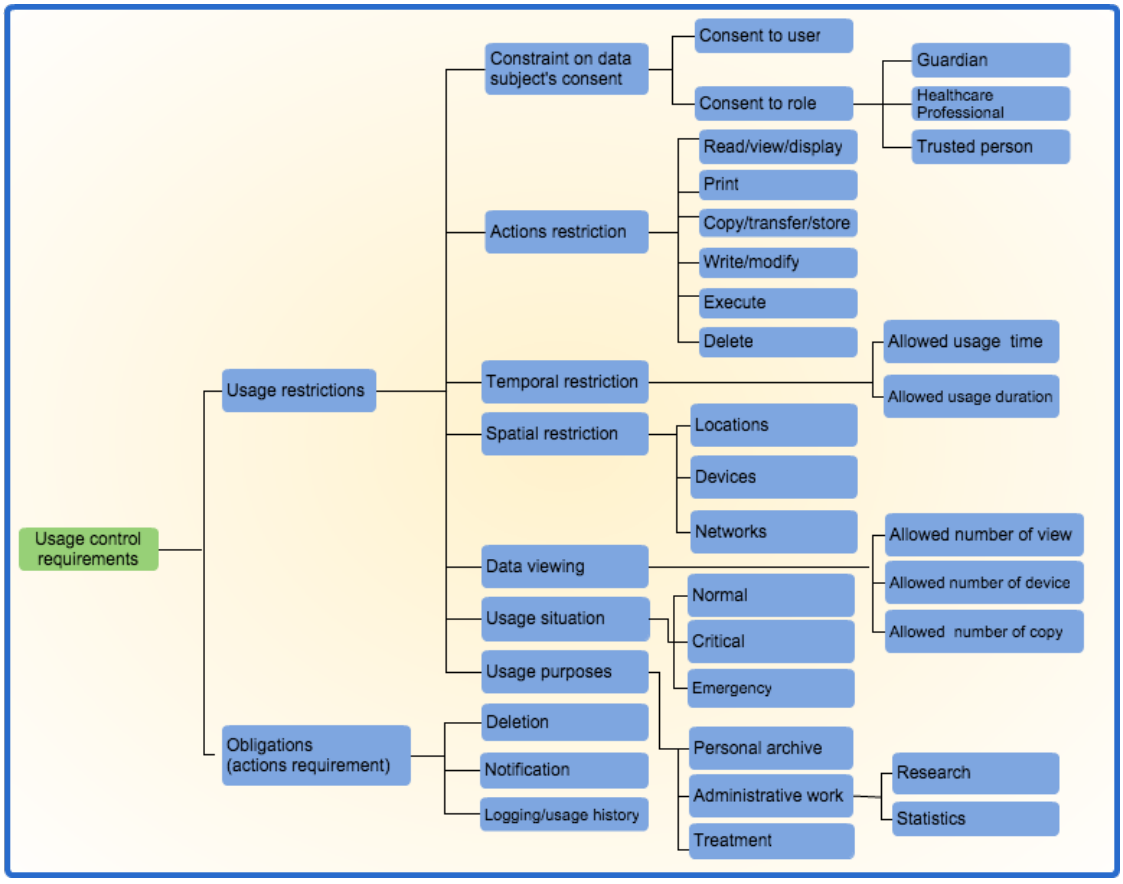

Figure 4.4: Classification of different requirements for usage control differentiates between usage restrictions and Obligations.

and Obligations, the details are illustrated in Figure 4.4.

4.4.1 Usage Restriction

Usage restriction defines the circumstances under which the content can be viewed and cannot be viewed. In this section, we present the usage restriction applied to distributed healthcare in general.

Constraint on data subject's consent refers to the permission granted by data subject to a person or a group of people. With consent of data subject, data processor can use data accordingly.

Action restriction refers to allowed or disallowed actions on patient's record. For example, "read", "write", "modify", "play" or "print".

Temporal usage restriction refers to time within which user is allowed to view patient's record. For example, user can view patient's record within working hours or during the treatment processes. It is important to note that, under context of healthcare information system, the amount of time permitted to use content must be sufficient, but not over necessary.

Spatial usage restriction refers to the place/location where data is allowed to be used, particularly, when data is moved out of its original location. It may be a healthcare institution, a specific computer/hardware or network. Spatial constraint allows policy designer to filter the unnecessary locations in the network where the data is not required to be used.

Data usage restriction refers to the number of times allowed to use data when it is stored at remote client. Restricting the number of times for data usage helps preventing the problem concerning the illegal data distribution and unnecessary data usage.

Usage situation restriction refers to user access situation. There are three main usage situations [54]. Firstly, it is normal situation. For example, the routinely check up of patient. Secondly, the critical situation, in this situation, hospital staff may bypass certain rules to ensure the proper treatment of patient. The last situation is the emergency where hospital's staff can declare bleak-glass situation. In this situation, all the rules are bypassed given the risk of patient's life.

Usage purpose restriction, one of the most important constraints, ensures patient's record being used is in the right direction. It is important to note that usage purpose should be continuously controlled during usage session.

4.4.2 Obligations

Obligation refers to any duty that needs to be executed by user or system during the usage session. For example, a "delete" action is required to be performed after the usage license is expired while "notify" needs to be performed before and after the use of data.

Deletion and Store "Delete" is an action that needs to be performed by user or system when the usage policies or rights are expired. Other action is "store". In distributed setting, data is shared among different organisational entities in the network. It is possible that data is temporarily stored at the destination repository.

Notification is very important in order to keep patient informed about the use of his record. Notification allows patient to know who uses his record at what time for what purpose. Notification can take place before or after the usage of content.

Logging (usage log) is a way to capture the usage information when data is used at remote client application. It records all necessary information before, during and after the use of content. Then, it is used to verify or check the validity and legality of the content usage. The information can be the actions executed on content, user identity, time at which user uses content, location and device.

4.5 Access and Usage Control Model Selection

In this section, we present the summary of the access and usage control requirements and then we point out which model is suitable to be used in our context. It is worth noting that the description of each model including its weakness and strength was presented in Chapter 2. This section is based largely on the information presented in Chapter 2. The usage control requirements listed are the general requirements for expressing privacy-aware usage control policies. The usage control model that we choose to be used in our context must be able to respond to all these requirements. For example, we state the usage control policy taking from Wallonie Healthcare Network [54], it says:

"***any user in role cardiologist can read patient's past heart record if and only if they have patient's consent, they are on-duty at the time of usage and they use past heart record with the support of hospital's devices. Moreover, they are obliged to notify patient on every access. They are also obliged to allow the system to log their activities during usage session. In addition to that the usage of that record is eligible for heart-treatment purpose only.***"

From that usage statement and the requirements in Figure 4.4, we can identify the entities required in the model as presented below. The model that will be used to express such policy must consists of these required entities.

1. **User** is an entity representing a person who uses the data. If we refer to policy statement above. It corresponds to the term **"any user"**.

2. **Role** is an entity presenting a profession or job responsibility of an individual or a group of user. If we refer to policy statement above. It corresponds to the term **"Cardiologist"**.

3. **Data** is an entity representing the data or resource to be used by user. If we refer to policy statement above. It corresponds to the term **"past heart record"**.

4. **Action** is an entity expressing the allowed action on data. If we refer to policy statement above. It corresponds to the term **"read"**.

5. **Condition** is an entity that can be used to express any constraints on data usage. For example, patient's consent is considered as constraint on data usage.

6. **Context** is an entity used to express contextual information that need to be valid during usage session. For example, time of use is considered as contextual information that needs to be validated when using data. If we refer to policy statement above. It corresponds to the term **"using the record when they are on-duty"** and **"using hospital's device to access the record"**.

7. **Purpose** is an entity used to express purpose of using data. It is an important entity for expressing privacy aware policy. If we refer to policy statement above. It corresponds to the term **"heart-treatment"**.

8. **Obligation** is an entity representing the duty that user needs to fulfil before or after using data. For example, notifying data subject or accepting system to log usage activities are the forms of obligation. If we refer to policy statement above. It corresponds to the term **"notify patient"** and **"log their activities"**.

We will present in detail below the access and usage control model selection. We look at each access and usage control model presented in Chapter 2 and compare their entities (entities in the core model) against the required entities we listed above. Then, we conclude our discussion and point out which access and usage control model should be used in our case. The comparison between the required entities and the entities in the core of the existing models is presented in Table 4.1.

	User	Role	Data	Action	Condition	Context	Purpose	Obligation
DAC	Yes	No	Yes	Yes	No	No	No	No
MAC	Yes	No	Yes	Yes	Yes	No	No	No
RBAC	Yes	Yes	Yes	Yes	No	No	No	No
P-RBAC	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
OrBAC	Yes	Yes	Yes	Yes	Yes	Yes	No	No
ABAC	Yes	No	Yes	Yes	No	No	No	No
HBAC	Yes	No	Yes	Yes	No	No	No	No
GBAC	Yes	No	Yes	Yes	No	No	No	No
RelBAC	Yes	No	yes	Yes	No	No	No	No
UCON	Yes	No	Yes	Yes	Yes	No	No	Yes

Table 4.1: The comparison between the required entities and the entities in the core of the existing models.

1. DAC is a simple access control model where the data subject has the rights to determine the access permission on data. DAC is the basic model where there are only three entities in the model: user, data and action (see Table 4.1). DAC is not able to support the complex privacy policy because of the absence of many entities in its core model, such as the ability to express user's role, purpose of use, obligation and context. Consequently, it can not be used in our context.

2. MAC provides better control on data compared with DAC. In MAC model, system constrains the ability of a subject to access or perform some sort of actions to an object. Subjects and objects each carry a set of security attributes and when a subject makes an attempt to access an object, an authorisation rule controlled by the system examines these security attributes. The security attributes, in MAC, are kind of conditions on user and data. Similar to DAC, MAC's core model does not provide the ability to express role, purpose and obligation (see Table 4.1) since the MAC's core model supports only the expression of user, action, data and condition on user and data. Thus, it can't be used in our context.

3. RBAC is a well-known model where the access authorisation is formulated around role of user. Users in the same role experience the same level of control. The traditional RBAC model consists of four entities: user, role, data and action. The use of roles to control access can be an effective means for developing and enforcing complex enterprise-specific security policies and for facilitating the security management process since it allows policy administrator to separate the user management domaine from policy management domaine. Although RBAC is better than DAC and MAC, it is still not sufficient to be used in our context because the absence of other entities allowing to express privacy policy such as,

purpose, condition, obligation and context (see Table 4.1).

4. P-RBAC is an extension of RBAC model. The extension aims at making the traditional RBAC to support the expression of privacy policies. Three entities, such as Condition, Obligation and Purpose, were added to the core model of RBAC. These entities allows for expressing complex and privacy-sensitive policies. P-RBAC supports nearly all our required entities (see Section 4.5), except, it does not have the concept of "contextual information".

5. OrBAC is a model where the access permission is formulated around an entity called "organisation". OrBAC has a concept of abstract and concrete level, which make it different from other models. OrBAC has similar concept to that of RBAC and provides also the ability to express complex policies. However, purpose and obligation are not directly expressed in the core model (see Table 4.1).

6. ABAC is a logical model that is distinguishable from other models because it controls access to objects by evaluating rules against the attributes of the entities (subject and object) actions and the environment relevant to a request. Attributes in ABAC are general concept and ABAC does not define them explicitly. The concept of role, purpose, obligation does not exist in ABAC (see Table 4.1) although those entities can be expressed informally as the attributes of user or data. ABAC is general model to be used to express general access control policy. It was not originally designed to address privacy policy. Hence, it lacks some features to support our defined requirements (see Section 4.5).

7. HBAC is a model where the access authorisation is based on the access history of user. The access permission is formulated around user, data, action and the access history of user. This model is not specifically designed for addressing access control to private data; hence, the entities such as purpose, obligation and condition are absent in its core model (see Table 4.1). This rules out the possibility of using this model in our context.

8. GBAC is a model where the access authorisation is formulated around group where the user belongs. This model is similar to RBAC, the only different is that RBAC formulates the policy based on role of user (represent the profession or responsibility of user).

9. RelBAC is a model where access authorisation is based on the relationship between user and data owner. There are no concept of role, purpose or obligation that are required to express privacy policy in RelBAC. Consequently, it can not be used in our context.

10. UCON is a usage control model proposed by [41]. It is designed to address the control of usage of general data. That's why there is no concept of purpose in its core model. UCON does provide some features that can be used to express privacy-aware usage control policies, such as condition and obligation. However, it does not explicitly provide a way to express "purpose" in its core model (see Table 4.1). Moreover, UCON permission assignment, based on "subject" not "role", which makes it difficult to be used in distribute environment.

Discussion. Based on our study the properties of each access and usage control model and the comparison of entities of each model as shown in Table 4.1, we conclude that for access control model, we can use P-RBAC, since this model was originally designed for controlling the access to private data. P-RBAC is able to express not only highly complex access control policy, but also the access control to private data. The present of the entities, such as "Purpose", "Obligation" and "Condition" in the core model are for that objective. While P-RBAC is our choice for access control, UCON fits with our requirements of usage control, but the extension is required since the original UCON model does not provide sufficient features to support the expression of privacy-aware usage control policy. For example, the absence of the entity "purpose" in the model. Consequently, we propose an extension of UCON model we term as "Privacy-Aware UCON", which will be presented in detail in Section 4.6.

4.6 Usage Control Model: Privacy-Aware UCON.

With the usage control requirements we listed in Section 4.5, we see the need to have a usage control model that is able to address those requirements, the most important of which are the obligation, purpose and condition. These three entities are the driving force, allowing for the expression of complex and fine-grain privacy-aware usage control policies. According to our study published in 7th International Conference on Health Informatic [11][10] (see also Section 4.5 and Table 4.1), we conclude that UCON, proposed by Jaehong Park and Ravi Sandhu [41], is the suitable one. However, some extensions are required for the traditional UCON to support the requirements we have identified. We introduce below the existing UCON and the extended one.

4.6.1 UCON Model

UCON encompasses traditional access control, trust management, and digital rights management and goes beyond them in its definition and scope. UCON enables fine-grained control over usage of digital objects than that of traditional access control policies and model. UCON model consists of six components (see Figure 4.6), such as subjects, rights, objects, conditions, authorisation rules and obligations.

1. **Subjects** are entities associated with attributes, and hold and exercise certain rights on object. The attributes are properties of subjects that can be used in authorisation process. For example, subject's identification.

2. **Objects** are entities that subjects hold rights on. Objects can be anything, such as multimedia contents or system resources. In general, objects are associated with attributes that can be used in the authorisation process like those of subjects.

3. **Rights** are privileges that subjects can hold on an object. Rights consist of a set of usage functions that enable a subject's access to object. Rights associates subjects and objects. In general, rights can be viewed as the usage actions allowed to perform on object, such as view, modify, copy or transfer. UCON rights can be divided into many functional categories as presented above. The two most fundamental rights categories might be a view and a modification. They are denoted as V and M respectively so we write R = {V, M}. Modification includes change to an existing digital object and creation of a new object that reuses an original digital object.

4. **Authorisation rules** are set of requirements that should be satisfied before allowing subjects access to objects or use of objects. There are two types of authorisation rules: Rights-related Authorisation Rules (RAR) and Obligation-related Authorisation Rules (OAR).

    - The RAR is used to check if a subject has valid privilege to exercise certain rights on a digital object. Examples include identities or roles verification, capabilities or properties checking, proof of payments, etc.

    - OAR is used to check if subject has fulfilled or agreed to fulfil their obligation, for instance, notify to patient or agreed on logging the usage activities.

5. **Conditions** are set of decision factors that the system should verify at authorisation process along with authorisation rules before allowing the use of data.

6. **Obligations** are mandatory requirements that subject has to perform before or after exercising rights on an object. There exist two possible obligations, namely, pre-obligation and post-obligation.

    - Pre-obligation refers to any actions that need to be performed before the usage session starts (e.g. in e-health, subject may have to notify patient before using patient's record).

    - Post-obligation refers to any actions that need to be performed after the use of content (e.g. delete patient's record after using it).

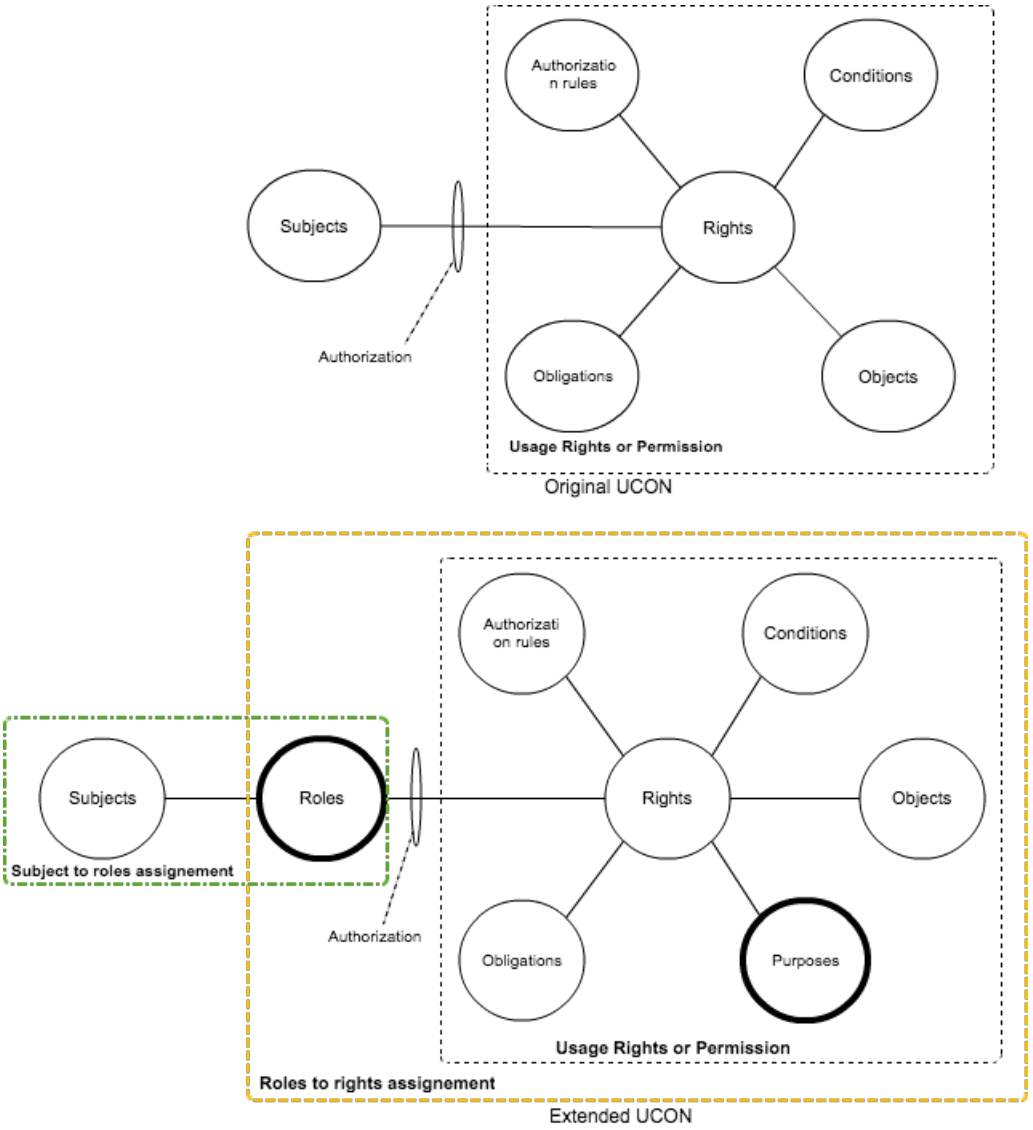

Figure 4.5: Traditional and extended UCON model components with purposes and roles extension [41].

4.6.2 Extended UCON: Privacy-aware UCON

UCON is a general usage control model that is not designed to particularly address the control of sensitive private data (see Section 4.5 and Table 4.1 for the requirements and comparison between different models). Thus, for our consideration, we propose to extend it by adding "**purposes**" (see Figure 4.6) component into the model to make it suitable to express the privacy-aware policies. Furthermore, to facilitate policy management, we propose a second adjustment. We add "roles" (see Figure 4.6) entity to the model. Instead of assigning the usage permission directly to the users (or subjects), we propose to assign usage permission to an abstract entity called "roles".

Roles, in an organisational structure, generally refer to profession, job title or the responsibility of an individual or a group of people. For example, administrator, marketing manager or receptionist.

With this modification, subjects are assigned to roles (see the requirements in Section 4.5 and Table 4.1) and everyone in the same role has the same level of usage rights. Assigning a usage permission to a group of users instead of individual user allows policy maker to minimise user management issue. When data with its usage policy are moved to another system away from the source system, in general, only users stipulated in usage policy could use data if a direct user to data permission assignment method is used. Using direct assignment between user and data permission in context of distributed environment can pose a burden for user as well as policy management since every system in the network needs to have a full list of users, not only the users in their system, but also the lists of users of other systems in the network. Any change in user management structure in a system, other systems in the network need also to update it; this causes the management overload. Seeing the difficulties more than the eases, we propose to use "role to data permission assignment" instead of "user to data permission assignment". Using role allows system engineer to separate the user management domain and the management can be done internally, only "role" needs to be managed globally.

To adjust to the change of the entity in the core UCON model, we introduce another type of authorisation rules over the two existing rules (RAR and OAR). We term it as "PAR: Purpose-related Authorisation Rules".

PAR is used to check if the purpose claimed by subject (or user) is valid. The validation of PAR depends on type of purpose. For example, if the purpose is to perform "heart surgery". What we need to validate is that user really uses the data for "heart surgery" and not for other purpose. In this example, a good source of information to validate the "heart surgery" purpose is the operation room since user can not claim that he uses data for heart surgery purpose if he has not reserved the operation room. Operation room reservation is a good justification or a proof for "heart surgery" purpose.

4.6.3 Privacy-aware UCON Model Expression

In the privacy-aware UCON in Figure 4.6, we have the following entities.

- Subjects (S); Role of subject (R); Objects (O); Rights on object (Ri);
- Conditions (C); Obligations (OB); Purposes (P);

In traditional UCON the authorisation is done based on the individual subject, there is no concept of role. However, the the privacy-aware UCON, the concept of role is introduced. Thus, the right on object is no longer assigned to individual subject, but to the role representing the profession of subjects. The role to subject assignment (RA) is the function that assigns a user to a particular role. The subject to role assignment is expressed as (S,R). Example, RA(David, Physician): David is a "subject" and "physician" is a role.

In privacy-aware UCON, authorisation process can be done through role based on the authorisation rules (AR) (RAR, OAR and PAR). Ri(R,O) means a set of authorised rights for role R on object O. The authorisation can be done in different ways.

- Ri(R, O)= AR(At(S), At(O)); authorisation is done by checking certain authorisation rules based on subjects' attributes (At(S)) and objects' attributes (At(O)).
- Ri(R, O)= AR(At(S), At(O), C); authorisation is done by checking certain authorisation rule based on subjects's attributes, objects' attributes and certain conditions.
- Ri(R, O)= AR(At(S), At(O), C, P); authorisation is done by checking certain authorisation rule based on subjects' attributes, objects' attributes, certain conditions and purpose of use.
- Ri(R, O)= AR(At(S), At(O), C, OB, P); authorisation is done by checking certain authorisation rule based on subjects' attributes, objects' attributes, certain conditions, obligations and the purpose of use.

For example, a usage policy for medical records sates that "user in role physician is allowed to read blood-test records for purpose of diagnosis" if and only if the physician is on duty and he needs to notify to system when he starts usage session. Then, based on the formulation above, we can express the above policy as follows.

Read(physician, blood-test)=AR(on-duty, notify, diagnose) Supposing that the policy wants to limit the access for only user David in role "physician" for the blood-test of Dara, then we can formulate the following role.

Read(physician, blood-test)=AR(At(physician)=David, At(blood-test)=Dara, on-duty, notify, diagnose)

4.7 REL Profile for Access Control Models

In Section 4.5, we discussed the usage control requirements and we identified the required entities for expressing privacy policies. In Section 4.6 we discussed the selection of access and usage control model based on the requirements in Section 4.5. In this section, we present ODRL and XACML profiles[1] for different access control models presented in Chapter 2 and the model selection presented in Section 4.6. We point out which rights expression languages can be used to express access control policy of which access control model. The main goal of this section is to select a standard policy language given the selected model presented in Section 4.6. In order to do so, we need to study the core model of each policy language and then compare the entities in its core model to those of the access and usage control models we presented in Section 4.6. Although several rights and access control authorisation languages were presented in Chapter 2, we focus only on two languages in this section: ODRL and XACML. For other languages beyond ODRL and XACML, one can refer to our technical report at [69]. Furthermore, the technical reports concerning ODRL profile for PRBAC and XACML profile for RBAC can also be found at [19] and [8], respectively.

	ODRL	XACML
RBAC	Yes	Yes
P-RBAC	Yes	Yes
OrBAC	No	Yes
ABAC	Yes	Yes
HBAC	Yes	Yes
GBAC	Yes	Yes
RelBAC	Yes	Yes
UCON	Yes	Yes

Table 4.2: The list of access and usage control model and the policy languages. "Yes" means the language can be used to express the policy of that model, "No" means otherwise.

4.7.1 ODRL

Given its well-designed core model and its rich vocabularies, ODRL 2.0 can be used to express different types of access control policies of the following models.

[1]REL Profile is the outline or description of the functionalities and components in REL's core model.

RBAC: the classical RBAC grants access based on the role of user, every user in the same role experiences the same degree of control over asset. The core model of the classical RBAC consists of four entities, which are User, Role, Data (asset) and Action. By comparing the entity model of RBAC and the core ODRL 2.0 model (see Figure 2.4), we found that, ODRL can be used to express the RBAC policy by using some entities in its core model, such as party, associate class role, asset, action and permission. In addition, ODRL can also express the context-aware RBAC policy by using others entities in the core model such as constraint or duty.

RBAC's entities	ODRL's entities
User	Party
Role	Associate class role
Action	Action
Data	Asset

Table 4.3: The table mapping the entities of RBAC to those of ODRL. For more details about ODRL, refer to Section 2.2.4.1 and Figure 2.4.

P-RBAC: P-RBAC is an extension of a classical model RBAC provides complete support for expressing highly complex privacy-aware policies. Its focus is to protect the data containing personally identifiable information and as such privacy-sensitive, taking into account characteristics such as purposes, conditions and obligations. These three entities allow for the expression of highly complex privacy-related policy. Using the ODRL to express the P-RBAC access control model is a good match because all the entities required in P-RBAC are available in core ODRL 2.0 model. Beside of the entities such as subject (party in ODRL), role, action and object (asset in ODRL), ODRL introduces others entities such as duty and constraint where duty can be mapped to obligation in P-RBAC and Constraint can be mapped to Condition and Purpose. For more details, one can go to our technical report "ODRL profile for PRBAC" at [19].

OrBAC: OrBAC model is based on three principles: organisation, concrete and abstract level and context. Like other access control models, concrete authorisation in OrBAC relies on three entities, which are subject, action and object. Typically, subject in concrete level is mapped to a role in abstract level where action is mapped to activity and object is mapped to view (see Section 2.1.2.4 for more details). By comparing the data model of ODRL 2.0 and the core model of OrBAC, we found that ODRL can partially be used to express the access policy in

OrBAC model. However, it is not able to cover all requirements and use case scenarios in OrBAC. For example, in OrBAC, there is a possibility to express a separate rule for the obligation (known as duty entity in ODRL). In ODRL, there is no direct link between policy entity and duty. Duty is considered as a part of permission. Thus, it is hard to express a duty as a separate rule in ODRL. For permission and prohibition, they can easily be expressed in ODRL 2.0. One more obstacle is the concept of abstract and concrete authorisation introduced in OrBAC, as ODRL does not have this concept, it is not possible to use ODRL to express both levels of authorisation together. However, ODRL can be used to express the abstract authorisation in OrBAC.

P-RBAC's entities	ODRL's entities
User	Party
Role	Associate class role
Action	Action
Data	Asset
Condition	Constraint
Purpose	Constraint
Obligation	Duty

Table 4.4: The table mapping the entities of P-RBAC to those of ODRL. For more details about ODRL, refer to Section 2.2.4.1 and Figure 2.4.

ABAC: ABAC is an access control model where the access right is decided by a set of the attributes associated with user. Each attribute is a distinct discrete and possibly unrelated field. The access authorisation is based on the comparison between the attributes' values presented by user to the predefined values in the system. Given the characteristic of ABAC, we found that ODRL can be used to express the access policy of ABAC model by considering all the entities such as party, role, asset, duty and constraints as the separate attributes.

GBAC: GBAC is a classical access control model that has been deployed in various application and system. In GBAC, access is granted based on group, and users under the assigned group can exercise the same level of right. GBAC can be expressed by using ODRL, associate class "role" in ODRL 2.0 contains an attribute called "scope", which have two possible values: "individual" and "group". "Individual" indicates that the policy set is for an individual person while group indicates the policy for a group of people. With this feature, we can use ODRL to express the GBAC model. It is important to note that the entities of GBAC are the same as those of RBAC.

P-RBAC's entities	XACML's entities
User	Subject
Role	Subject
Action	Action
Data	Resource
Condition	Condition
Purpose	Condition
Obligation	Obligation

Table 4.5: The table mapping the entities of P-RBAC to those of XACML. For more details about XACML, refer to Section 2.2.5.1.

4.7.2 XACML

Although XACML is designed primarily for attribute-based access control policy, it can be used to express number of other access control models, such as RBAC, P-RBAC and OrBAC.

RBAC, GBAC and P-RBAC: XACML can be used to expresses both RBAC and PRBAC's policies. The elements in both access control models can be mapped to the elements of XACML policy model (see Table 4.5). Moreover, XACML provides also a way to express obligation, which is a perfect match to PRBAC requirements. Condition expression is also available in XACML. The XACML profile for RBAC and P-RBAC can be found at [8]. In XACML [87], roles are expressed as XACML Subject Attributes. The action is expressed "action tag" in XACML while data is expressed in "resource tag". Condition and purpose can be expressed in "Condition" while obligation can be expressed in "Obligation tag". Table 4.5 provides a mapping information between the entities of P-RBAC to those of XACML.

ABAC: rule in XACML is a combination of subject, action, condition, obligation and purpose. These attributes can be related or unrelated to each other and they are considered as separate elements in the rule evaluation process. Given a characteristic of ABAC and the XACML profile, we found that XACML is a good match for ABAC model. XACML controls access based on the policy set defined in the system; request to resource must be complied with the rules defined in the policy set. XACML allows policy-writer to write complex access control policies in different contextual environment by exploiting some of its entities in the core policy language model such as condition purpose and obligation. It is worth noting that XACML is the attribute-based access control language.

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy
      xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
      http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-context-schema-os.xsd"
      PolicyId="urn:oasis:names:tc:xacml:2.0:conformance-test:IIA11:policy"
      RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
    <Description>Subjects to role assignment.</Description>
    <Target/>
    <Rule RuleId="urn:oasis:names:tc:xacml:2.0:conformance-test:IIA11:rule" Effect="Permit">
        <Description>Assigning users to a role</Description>
        <Target>
            <Subjects>
                <Subject>
                    <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Dara</AttributeValue>
                        <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
                         DataType="http://www.w3.org/2001/XMLSchema#string"/></SubjectMatch></Subject>
            </Subjects>
            <Resources>
                <Resource>
                    <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                        <AttributeValue
                         DataType="http://www.w3.org/2001/XMLSchema#anyURI">Physician</AttributeValue>
                        <ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
                              DataType="http://www.w3.org/2001/XMLSchema#anyURI"/></ResourceMatch></Resource>
            </Resources>
        </Target>
        </Rule></Policy>
```

Figure 4.6: Subjects to role assignment policy in XACML: Dara is assigned to role "Physician".

4.7.3 XACML Profile for Privacy-aware UCON

In this section, we define XACML profile for privacy-aware UCON. This specification defines a profile for the use of the OASIS eXtensible Access Control Markup Language (XACML) to meet the requirements for privacy-aware UCON. Use of this Profile requires no changes or extensions to standard XACML Versions 1.0 [87]. We assumes that reader is somewhat familiar with XACML. A brief overview sufficient to understand these examples is available at [87]. It is worth noting that this specification is important for the implementation of our usage control and enforcement system, which will be presented in Chapter 7.

Role and Subjects, in this specification, roles are expressed as XACML Subject Attributes in usage control policy. There is one exception: in a Role Assignment $< Policy >$, the role appears as a Resource Attribute while the "subjects" appears as subject attributes in XACML (see Figure 4.7).

Objects, in this specification, objects are expressed as XACML Resource Attributes in usage control policy.

Authorisation rules contains the actual permissions associated with a given role. It contains $< Policy >$ elements and $< Rules >$ that describe the resources

(objects) and rights that subjects are permitted to use resources, along with any further conditions, obligations and purposes of using objects. The rights (in privacy-aware UCON) are expressed as $< Actions >$ in XACML.

Conditions, Obligations and Purposes, these three elements in privacy-aware UCON are expressed as $< Conditions >$ in XACML policy.

We provide a privacy-aware UCON policy (example from Section 4.6.3) expressed in XACML in Figure 4.6 and 4.7. Figure 4.6 is the policy expressing subjects to role assignment where user "Dara" is assigned to role "physician". Figure 4.7 is the authorisation rule that describes the resources "blood-test" and rights "read" that subjects in role "Physician" are permitted to use resources, along with condition "user must be on-duty", the purpose must be for "diagnose" and user is obliged to "notify" system every access.

4.8 Summary

In this chapter, we introduced a model for processing of private data applied to distributed healthcare. In that, we presented an execution models: physical execution. The presented execution model is based on data processing scenario in healthcare information system where two healthcare institutions share their patient's health records based on a contractual agreement between both parties. The execution models (see Figure 4.3) are for two different scenarios. Firstly, data transfer scenario, it happens when one healthcare institution needs patient's health records from their counterpart for a particular purpose and those health records need to be transferred and temporarily stored on requester's local storage. Secondly, it is the usage scenario where users (e.g. physicians) request to use the records recently transferred. As processing private data requires consideration of various legal, contractual, organisational and technical aspects, we studied different requirements for controlling such data. We presented three requirements, included legal-, contractual- and technical requirements. As the result from our study, we proposed the general usage control requirements for distributed healthcare system. We published this result in 7th International Conference on Health Informatics, Barcelona, Spain and the original manuscript can be found in [10]. Based on the identified usage control requirements, we proposed a usage control model taking into account the aspects of roles and purposes. We termed that model as privacy-aware usage control model or Privacy-aware UCON. We published Privacy-aware UCON in 7th International Conference on Health Informatics, Barcelona, Spain and its original manuscript can be found in [11].

Moreover, we also presented the analysis of access and usage control models based on our defined requirements. We compare each model against a set of requirements we defined and find out which model is best to fit our case. From our study (see Section

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy
      xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
      http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-context-schema-os.xsd"
      PolicyId="urn:oasis:names:tc:xacml:2.0:conformance-test:IIA10:policy"
      RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
    <Description> Usage Policy for users in role physician.</Description>
    <Target/>
    <Rule RuleId="urn:oasis:names:tc:xacml:2.0:conformance-test:IIA10:rule" Effect="Permit">
        <Description>Physician reads patient records</Description>
        <Target>
            <Subjects>
                <Subject>
                    <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Physician</AttributeValue>
                        <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
                         DataType="http://www.w3.org/2001/XMLSchema#string"/></SubjectMatch></Subject>
            </Subjects>
            <Resources>
                <Resource>
                    <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
                        <AttributeValue
                         DataType="http://www.w3.org/2001/XMLSchema#anyURI">http://foi.se/Blood-test
                         </AttributeValue>
                        <ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
                              DataType="http://www.w3.org/2001/XMLSchema#anyURI"/></ResourceMatch></Resource>
            </Resources>
            <Actions>
                <Action>
                    <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Read</AttributeValue>
                        <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                         DataType="http://www.w3.org/2001/XMLSchema#string"/></ActionMatch></Action>
            </Actions>
        </Target>
        <Condition>
         <Apply  FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
               <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                 <EnvironmentAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:environment:purpose"
                       DataType="http://www.w3.org/2001/XMLSchema#string"/>
               </Apply>
               <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Diagnose</AttributeValue>
            </Apply>
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
               <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                 <EnvironmentAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:environment:On-duty"
                       DataType="http://www.w3.org/2001/XMLSchema#string"/>
               </Apply>
               <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">yes</AttributeValue>
            </Apply>
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
               <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                 <EnvironmentAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:environment:notify"
                       DataType="http://www.w3.org/2001/XMLSchema#string"/>
               </Apply>
               <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">yes</AttributeValue>
            </Apply>
         </Apply></Condition></Rule></Policy>
```

Figure 4.7: Authorisation rule or permission rule in XACML: users in role "physician" can read blood-test records if and only if he is on-duty and every time he accesses to data, he needs to notify the system. Moreover, the permission is for patient's "diagnose" purpose only.

4.5) we see that most of access control models (like DAC and MAC) fail to respond to the access control requirements (see Section 4.4) of such systems [10] since such systems generally have very complex access control policies (see our study published in [13]). The basic RBAC model, where access policy is formulated primarily around role, is not also sufficient. Other models, like OrBAC provides more expressing power. However, it is not specifically designed for privacy-aware system. Based on our study published in [13], we conclude that P-RBAC is the best candidate to be used in such system since in that model the concept of purpose and obligation have been introduced and they are well formulated. The two entities (purpose and obligation) (see Section 4.3 and 4.4) are indeed the most important elements required for expressing privacy policies. The result of our survey on access control model for healthcare information system has been published in the Fourth International Conference on eHealth, Telemedicine, and Social Medicine [20].

While P-RBAC is our choice for access control, UCON is our choice as usage control model (see Section 4.5). However, the extension is required in order to make basic UCON to be sufficient to express the usage control policy for private sensitive data. This is because in basic UCON model [41], there is no concept of purpose, which is important for privacy policies. We propose to extend UCON to support purpose expression by introducing purpose as one of the core model components. Finally, in this chapter, we also presented the REL Profile for Access Control Models and the detailed XACML profile for privacy-aware UCON.

Chapter 5

Purpose Modelling

In Chapter 1 and Chapter 4, we presented the role of purpose in privacy policy and the need to limit the usage of private data to only the purpose it intends for. In this chapter, we mainly discuss the purpose definition and the modelling of purpose as workflow. We also present the resources management and assignment for task in workflow and the access control model for the system that use workflows. This chapter is organised as follows. Section 5.1 is about purpose definition and purpose model is discussed in Section 5.2. Section 5.3 presents the modelling of purpose as workflow. The access control for resources in workflow is discussed in Section 5.4. Finally, Section 5.5 is the summary of this chapter.

5.1 Purpose Definition

The goal of this section is to study the meaning of purpose. We aim at providing formal definitions suitable for the enforcement of purpose restrictions. Based on our work published in [11], we find that *planning* is central to the meaning of purpose [76]. We see the role of purpose in the definition of the sense of the word "planning" most relevant to our work in [58].

Planning is the process of thinking about and organising the activities or tasks required to achieve a desired goal [58].

We see also the role of planning in the definition of the sense of the word "purpose" most relevant to our work [76].

Purpose is the object for which anything is done or made, or for which it exists; the result or effect intended or sought, a goal or an aim [76].

Similarly, in dictionary [57], ***purpose is defined as the object towards which one strives or for which something exists, an aim or a goal.*** By observing the definition of the sense of the words "planning" and "purpose", we see the similarities. They are all about reaching a goal. Taking into account the meaning of "purpose" and "planning", it shows that purpose can be represented by a planning of tasks. In order words, we can say that *purpose* refers to a goal behind executing a set of tasks. For example, we withdraw money for *purpose* of shopping. To achieve shopping goal, we may have to complete some intermediate tasks, such as going to withdraw money from bank, driving a car, finding a parking lot, choosing items, paying, etc. Only after completing those tasks, we can say we achieve our goal that is shopping.

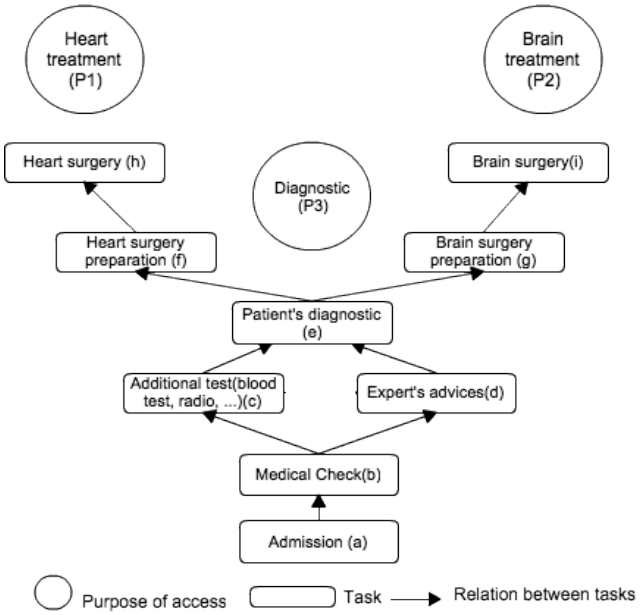

Figure 5.1: Example of task graph containing 3 *purposes*: Heart treatment, Diagnostic and Brain-treatment. Heart treatment is represented by a set of tasks: a, b, c, d, e, f and h. Brain treatment is consisted of tasks: a, b, c, d, e, g and i. Diagnostic is represented by "a, b, c, d and e".

5.2 Purpose Model

"*Purpose*" is abstractedly modelled as a planning of tasks. It is defined by a set of tasks and the relationships among them. The tasks are linked together in the form of network of relationships that capture the intention, or more precisely, the *purpose* of executing tasks. We call such network of relationships as a "*plan*". The abstract

model of purpose is then expressed in the form of task graph. This is the fundamental assumption that forms the basis of our *purpose* model in this thesis. Figure 5.1 is a task graph representing three *purposes*. This graph is based on the general procedure for healthcare processing in hospital in Belgium elicited from the book written by Jean Herveg and Anne Rousseau [44].

A task represents a single unit of work that needs to be executed within a workflow definition. Task can be an automatically executable method (background task) or it can need a user (physical person) to execute it.

5.2.1 Task Graph

Task graph (TG) is a graph in which each node represents a task to be performed. We define task graph as an abstract model to capture a *planning of tasks* with only one types of relationships: "Future task relation", which is represented by the letter "F" for simplicity.

Task graph is a directed graph with one sets of edges corresponding to one type of relationship as discussed above. It is defined as TG = (T, F) in which T is set of vertices each of which corresponds to a task. F is subsets of T × T, and respectively correspond to F-relationships. The task graph satisfies the following condition:

- TG is a directed acyclic graph. It forbids circularity in the graph.

"Future task relation", a task is a future task of another tasks if and only if it is a following task that needs to be executed. For example in Figure 5.1, "e" is a future task of "c".

Definition 5.1: Path

A path is a list of nodes $(n_1, n_2, ..., n_k)$ such that there is an arc from each node to the next, that is $v_i \rightarrow v_{i+1}$ for i= 1, 2, ..., k-1. The length of path is k-1. For example, in Figure 5.1, a path represented P1 consists of 7 nodes, that are a, b, c, d, e, f and h.

Generally, a purpose is represented by a complete set of tasks, in which there is a path from the first node (start node) to the last node (end node).

Definition 5.2: Task purpose mapping

Let ST be a set of tasks. A *purpose* (P) is mapped to a set of tasks and it is denoted by the mapping Set-of-Task-Purpose: ST $\mapsto$ P. For example, Figure 5.1, (a, b, c, d, e, f, h) $\mapsto$ P1.

It is important to note that since our purpose modelling allows a task to be part of more than one set of tasks representing different *purposes*, it implies that there are relationships between purposes. We discuss purpose relationship in next section.

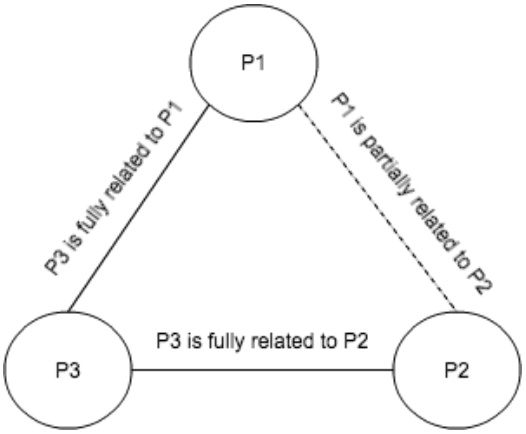

Figure 5.2: A purpose graph derived from Figure 5.1 showing the relationship between P1, P2 and P3. Dashed line represents the partial relationship while solid line indicates full relationship.

5.2.2 Purpose Graph

Purpose graph (PG), that expresses the relationships between purposes, is a labelled graph with two sets of edges, each corresponding to one type of relationship. It is defined as PG = (P, Pa, Fu) in which P is the set of vertices each of which corresponds to a purpose, and Pa and Fu are subsets of P × P, and respectively correspond to Pa- and Fu-relationships. Pa-relationship indicates that a purpose is partially related to other purpose while Fu-relationship indicates that a purpose is fully related to other purpose.

1. **"Fully related"**, a purpose is fully related to another purpose if and only if a set of tasks representing that purpose is a subset of tasks representing the other purpose. In Figure 5.1 and 5.2, P3 is fully related to P1 and P2. This is because P3 is represented by a set of tasks (a, b, c, d, e) that is a subset of a larger set of tasks, which is (a, b, c, d, e, f, h) representing purpose P1 and (a, b, c, d, e, g, i) representing P2.

Definition 5.3: Purpose relationship (fully related)

Let P_1 and P_2 be two different purposes. ST_1 and ST_2 are two sets of tasks where ST_1 and ST_2 are mapped to P_1 and P_2, respectively. P_1 is fully related to P_2, if and only if $ST_1 \subset ST_2$.

2. **"partially related"**, this relationship indicates that there is at least one task being shared by the two or more purposes. For example, in Figure 5.1, P1 and P2 are partially related because both purposes have common tasks, which are: a, b, c, d and e.

Definition 5.4: Purpose relationship (partially related)

Let P_3 and P_4 be two different purposes. ST_3 and ST_4 are two sets of tasks where ST_3 and ST_4 are mapped to P_3 and P_4, respectively. Let t_i be a task where i$\in N$ and N is the set of integers. P_3 is partially related to P_4, if and only if $ST_3 \cap ST_4 \neq 0$.

Definition 5.5: Relations between Purpose

Let P_i and P_j are two different purposes. Let Pa and Fu be two different relations for partial and full relationship respectively. Then, we have.

- (p_i, Pa, p_j), read as, p_j is partially related to p_i.
- (p_i, Fu, p_j), read as, p_i is fully related to p_j.

It is worth noting that we define purpose graph to be used in the "purpose validation process". This is because we argue that, in some cases, we can validate the claimed purpose[1] of access by looking at other purposes having relationship with the claimed purpose. The purpose for an action affects other purposes having relationship with it.

5.3 Purpose as Workflow

In this section we discuss the reason why we use workflow to model the purpose, which is abstractedly modelled early in the form of a planning of tasks in task graph. We

[1]Claimed purpose is a purpose of accessing data.

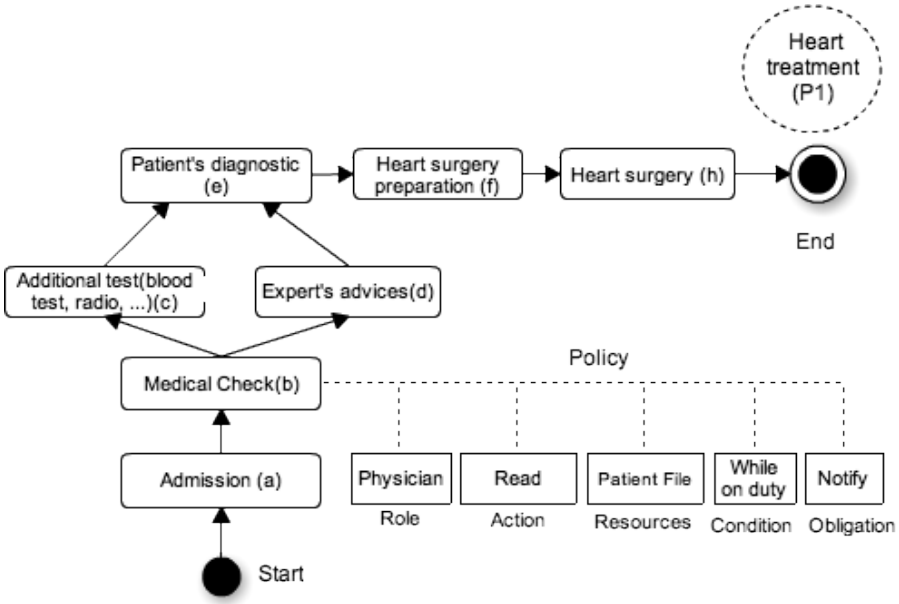

Figure 5.3: Example of workflow representing heart treatment *purpose* (P1). Figure 5.3 is derived from Figure 5.1.

also introduce the workflow definition and some of its properties.

5.3.1 Modelling Purpose with Workflow

A workflow [1] is a representation of an organisational or a business process in which documents, information or tasks are passed from one participant to another in a way that is controlled by rules or policies. A workflow separates the various activities of an organisational process into a set of well-defined tasks. Hence, generally, a workflow is specified as a set of tasks and a set of dependencies among the tasks, and the sequencing of these tasks is also important. The tasks in a workflow are usually executed by several users in accordance with organisational rules relating to the process represented by the workflow. Given the property of workflow, we see that it matches to our modelling of *purpose* in task graph. Workflow can accurately represent the abstract model of *purpose* as the planning of task represented in the form of task graph.

An assumed *purpose* may lead to bindings of future action. For instance, if Edward withdraws money from an account for *purpose* of buying a book, this implies that at a later time, he should buy some books and should not spend money otherwise. These future obligations are in perfect match with the workflow management model wherein execution of a task can only lead to specific future tasks. The importance of relationship between tasks in realising a *purpose* has also been pointed out in Section

5.2. The representation of a business process using a workflow involves a number of organisational rules or policies. Within the scope of security for resources management in workflow, access control policies play a key role, and hence defining and enforcing access control requirements becomes a key function of a Workflow Management System.

5.3.2 Resources Management in Workflow

A resource [53] is an entity assigned to a task and is requested at runtime to perform work in order to complete the task. The assignment of resources to task is performed based on the resource management rule. In general, resource is not limited to digital resources (e.g. digital files); it can be anything. For example, if the task is to buy a table, then the resource can be money. However, in this thesis, we focus mainly on the management of digital resources, other than that is not in the scope of our work.

Resources assignment policies: a dynamic perspective on the resources involved in workflow execution is the handling of the resource assignment to task of the workflow at runtime. There are three different concepts for resources assignment in workflow [53]: direct designation, assignment by role and assignment by formal expression.

1. **Direct designation:** a task is assigned to the resources directly. At run-time, the workflow-engine can directly retrieve these resources in the repository and place them on its work-lists. This kind of assignment is easy to handle for the workflow administrator, because he is concerned with a single entity type: the workflow executor. If a task is to be made available to a group of users, all members of the group have to be assigned to the task one by one. The direct assignment concept provides no independence of workflow model and organisational model. Therefore, the direct assignment mechanism is not a good choice in industrial practice given the dynamic nature of the organisational or structural change in such environment.

2. **Assignment by role:** most workflow management systems provide workflow modellers with a role entity type [53]. Within this domain, one role entity is used as a synonym for one or more resource entities. For example, "Cardiologist" defines a role of users within an application system that inherit a common set of access rights to resources. The main objective of the role model is the separation of workflow and resource model, where changes of the organisational structure do not affect the workflow model directly. The use of roles instead of a direct assignment also provides means of indirect workload balancing, because all members of an authorised role to the task are notified about the pending work-item, but only one member of this group needs to perform the activity.

3. **Assignment by formal expression:** the third form of resource assignment is called the assignment by formal expression. In this case, not only the entity

types of the resource model have to be known to the workflow creator but also the relationship between these entity types and possible functions depending on the workflow execution history. The attributes used in such formal expression can either be dependent on the workflow instance, such as the information about the workflow executor's last activity or other contextual information such as time allowed for task execution.

Based on the issues we outlined in Section 4.1.1 and the resources management requirements presented in Chapter 4 (Section 4.3), we adopt the third resource assignment concept: assignment by formal expression. The detailed formal resource access control model is presented in Section 5.4. The reasons to choose the third concept are: for direct designation, this method can create administration burden because administrator needs to assign each user to the resource. This can pose the problem when there is an organisational change, when there is a change, administrator needs to redo the assignment. For assigning by role, although this method can provide indirect workload balancing because all the members of an authorised role to the task are notified about the pending work-item, but only one member of this group needs to perform the activity, this method provides no granular control on resource since users in the same role can exercise the same level of rights.

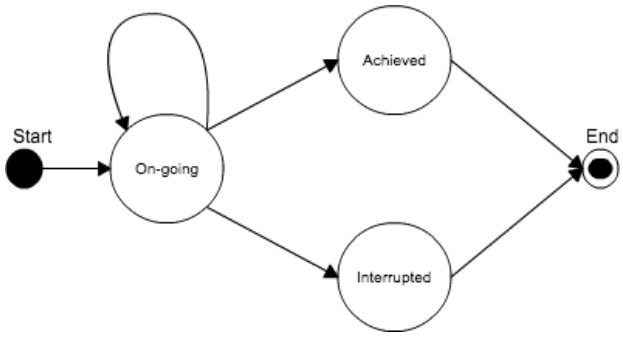

Figure 5.4: State machine representing different states of workflow execution.

5.3.3 Workflow Statuses

We use workflow system to enforce the *purpose* of access. When a workflow instance[1] is created, a unique identification is assigned to it. Then, that unique identification is used as reference for controlling the execution of tasks that are parts of the workflow. The information concerning the creation of workflow and the execution of the corresponding

[1]A workflow instance is a running instance of a workflow definition. Once a workflow instance has been started, it can not be changed.

tasks are logged into access history for later use. Once a workflow instance is created, there are three possible statuses: achieved, interrupted or on-going. These statuses indicate different states of workflow execution (see Figure 5.4).

1. **Achieved** indicates that the workflow instance is completed successfully. In order words, the claimed *purpose* has been achieved because all the tasks of the workflow representing that *purpose* have been executed successfully.

2. **Interrupted** indicates that the workflow instance is broken and the *purpose* is not achieved. We model *purpose* as a sequence of tasks; hence, it is possible that the execution of tasks may be interrupted. The interruption of the task execution can result in the broken workflow instance (see Definition 5.6).

3. **On-going** indicates that the workflow instance is in active mode. In other words, there are tasks that need to be executed and the times allowed for the execution of those tasks are not expired yet (see Definition 5.6). It is worth noting that the "on-going" can be changed to "interrupted", if the immediate next tasks are not executed on time.

By default when a workflow instance is created, "on-going" status is assigned to indicate that the created workflow instance is in progress. Then, the workflow status can be changed from "on-going" to "interrupted" or to "achieved" depending on the execution of tasks belonging to that workflow. To change the workflow status, we need two "status change" modules that analyse access-log and update the status accordingly. Below are the two status change algorithms.

- **On-going to Achieved:** every time a task is completed, "status change" module is activated. This module looks at the last executed task, workflow definition and workflow to purpose mapping. If the executed task is the last task of the workflow and this task is completed successfully, the "status change" module updates in access-log the status of this particular workflow to "achieved"; otherwise, the status is still in "on-going" state.

- **On-going to Interrupted:** every task or workflow has an allowed execution time. The workflow's lifetime indicates the maximum time allowed to complete all the tasks in the workflow while the task's lifetime indicates the maximum time allowed to complete a task. To change the status from "On-going" to "Interrupted", a "status change" module is activated periodically to analyse the access-log for all the workflows with "On-going" status. Firstly, this "status change" module selects from access-log all the information concerning the workflows with "On-going" status. Then, it examines one-by-one the lifetime of the workflow and its corresponding last executed task. If the lifetime of the workflow is not yet expired, it goes further to look at the expiration time of last executed

task. If the task's lifetime is expired, the status of workflow is updated to "Interrupted"; otherwise, "On-going" status is maintained. It is worth noting that, either workflow's lifetime is expired or task's lifetime is expired, the workflow status is updated to "Interrupted".

5.4 Access Control for Resources in Workflow

As mentioned in previous section, we use the "assignment by formal expression" as the resources assignment policy. Thus, in this section, we discuss mainly about the access control model for controlling access to resources for each task of the workflow.

A workflow management system contains a set of workflow description W,

$$W = \{W_1, W_2, ..., W_n\}$$

each of which consists of a set of tasks, $T = \{T_1, T_2, ..., T_n\}$. Every task is carried out by some actors on a set of resources and it is controlled and enforced by an access control policy or resources assignment policies (for short policy). The authorised actors for task and some access constraints are specified in access control policy. The assignment of access control policy to the task is specified by the mapping function:

$$\text{Policy-task: } T_1 \mapsto \textit{policy}$$

that maps each task to an applicable policy (see Figure 5.3). In summary, the workflow description $(_1)$ is defined as a triple containing $\langle T, E, SetPolicies \rangle$, where E is a set of arcs $E \subseteq T \times T$. A *purpose* corresponding to each workflow is denoted by the mapping

$$\text{Purpose-OF: } W_1 \mapsto \text{p.}$$

In general, each workflow instance has limited execution time. Each task of the workflow has also a limited execution time. The workflow instance or task, that spends more than allowed execution time, is considered as interrupted.

Definition 5.6: task and workflow instance execution time

Let L be an execution time variable of task or workflow instance. Then, we can define the following functions.
LT: $t \mapsto L$ is a function that maps a task to an allowed execution time.
LW: $W_i^j \mapsto L$ is a function that maps a workflow instance to an allowed execution time. Where "i" is the workflow definition number and "j" is the workflow instance number.

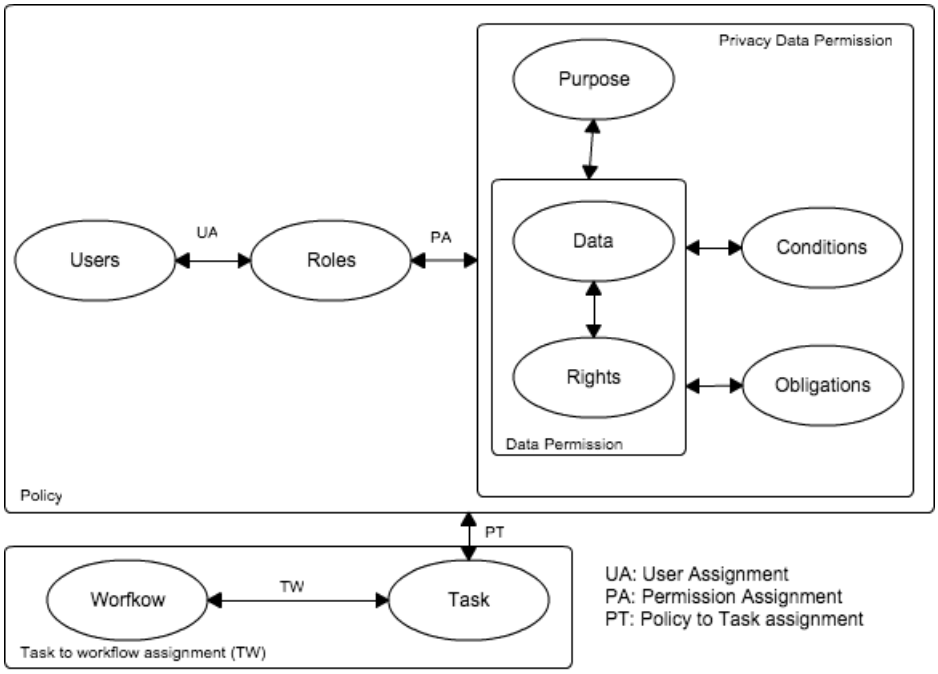

Figure 5.5: Access control model for resource (data) in workflow.

Access Control Model for Resources in Workflow

Every user $u \in U$ is assigned to a set of roles denoted by $ROLE\_OF(u)$ and every task is assigned to a policy. We use rule-centric as our policy model. Rule-centric policy [1] has a complex form compared with data-centric policy [1] where data item are associated with a set of intended *purposes*. Let RP be a set of policies, $RP \subseteq R \times G \times D \times C \times O \times P$ where:

- R is a set of user's roles (r) where $r \in R$. Role refer to a profession or the responsibility of an individual person or a group of people, for instance, cardiologist, radiologist, etc.
- G is a set of riGhts (g) where $g \in G$. "Right" refers to the permitted operation on data. The rights, such as "read", "write", "delete", "transfer", "modify", "copy" are the general operations on digital data performed by user. For example, user in role of cardiologist can "read" past health records of patient, "read" is considered as "cardiologist's right " on patient's health records.
- D is a set of data (d) where $d \in D$. Data is the digital resource that user needs in order to execute a task of a workflow.

- C is a set of conditions (c) where $c \in C$. Condition is a required constraint that needs to be satisfied before allowing user to access data.
- O is a set of obligations (o) where $o \in O$. Obligation is a duty that user needs to fulfil before or after accessing data. For example, a duty to pay before downloading music.
- P is a set of *purposes* (p) where $p \in P$. Purpose is a final goal of using data.

Then, we formulate our privacy-sensitive policy as follows.

1. The set of Data Permission $DP = \{(g, d) \mid g \in G, d \in D\}$
2. The set of Privacy-sensitive Data Permission $PDP = \{((dp, p), c, o) \mid dp \in DP, p \in P, c \in C, o \in O\}$
3. Privacy-sensitive Data Permission to role Assignment: PDPA $\subseteq R \times PDP$, a many-to-many mapping of privacy-sensitive data permission to role.
4. The Privacy sensitive policy to Task Assignment: PTA $\subseteq T \times PDPA$, a many-to-many mapping of privacy-sensitive policy to tasks (T).

Definition 5.7: condition expression

Let C be a set of conditions (c), where $c \in C$. "c" has the finite domain of possible values, denoted as DC where $dc \in DC$. "c" is equipped with the relational operators (Oprs) " $=, \neq, \geq, and \leq$ ". The condition of c has the form (c opr dc).

let c_1 and c_2 are two variables in the form of the atomic condition. Then, $(c_1 \wedge c_2)$ or $(c_1 \vee c_2)$ is also condition. For example, using working-hour of user as variable, if user's working-hour is between 8am and 5pm, we can express: working-hour $\geq$ 8am $\wedge$ working-hour $\leq$ 5pm.

Definition 5.8: obligation expression

Let O be a set of obligation variables (o), where $o \in O$. "o" has the finite domain of possible values, denoted as B where $b \in B$. "o" is equipped with the relational operators (Oprs) " $=, \neq, \geq, and \leq$ ". The condition of "o" has the form (o opr b). For example, a payment obligation has the form: $payment \geq 50\$$.

Example: 5.1: Access control policies for resources assigned to the tasks of purpose "Heart treatment".

Suppose that there is a set of policies applied to tasks "a, b, c, d, e, f, h" (see Figure 5.3) for *purpose* of "heart treatment". We have the following rules.

1. For task "a, b and c", the rule states "users in the role PHYSICIAN can READ David's blood records if they are on duty (between 8 AM to 5PM) and every time they access this information, they need to notify David. Moreover, David's consent to user is required".

Suppose that we have the following variables: Time (u) is a variable expressing the time. Consent (u) is a variable expressing a consent of David to user "u". Consent (u) has two possible values: "Yes, means user is consented by patient and No, means otherwise". Notify (David) is a variable expressing the notification obligation. Notify (David) has two possible values: Yes, means it is required to notify; No, means otherwise. With above information, we can formulate policy expression as follows.

1. PDPA to role "PHYSICIAN":
   PDPA= (PHYSICIAN, ((READ, blood records), heart treatment), ((Time (u)$\geq$ 8am $\wedge$ Time (u) $\leq$ 5pm) $\wedge$ Consent (u)=yes), Notify (David)=yes)

2. The Privacy sensitive policy to Task "a" Assignment:
   PTA = a $\mapsto$ (PHYSICIAN, ((READ, blood records), heart treatment), ((Time (u)$\geq$ 8am $\wedge$ Time (u) $\leq$ 5pm) $\wedge$ Consent (u)=yes), Notify (David)=yes)

3. The Privacy sensitive policy to Task "b" Assignment:
   PTA = b $\mapsto$ (PHYSICIAN, ((READ, blood records), heart treatment), ((Time (u)$\geq$ 8am $\wedge$ Time (u) $\leq$ 5pm) $\wedge$ Consent (u)=yes), Notify (David)=yes)

4. The Privacy sensitive policy to Task "c" Assignment:
   PTA = c $\mapsto$ (PHYSICIAN, ((READ, blood records), heart treatment), ((Time (u)$\geq$ 8am $\wedge$ Time (u) $\leq$ 5pm) $\wedge$ Consent (u)=yes), Notify (David)=yes)

2. For tasks "d, e, f, and h", the rule states "users in the role CARDIOLOGIST can READ David's past heart records if they are on duty (between 8 AM to 5PM) and every time they access this information, they need to notify David. Moreover, David's consent to user is required". With above information, we can formulate policy expression as follows.

1. PDPA to role "CARDIOLOGIST":
   PDPA= (CARDIOLOGIST, ((READ, past heart records), heart treatment), ((Time (u)$\geq$ 8am $\wedge$ Time (u) $\leq$ 5pm) $\wedge$ Consent (u)=yes), Notify (David)=yes)

2. The Privacy sensitive policy to Task "d" Assignment:
   PTA = d $\mapsto$ (CARDIOLOGIST, ((READ, past heart records), heart treatment),
   ((Time (u)$\geq$ 8am $\wedge$ Time (u) $\leq$ 5pm) $\wedge$ Consent (u)=yes), Notify (David)=yes)

3. The Privacy sensitive policy to Task "e" Assignment:
   PTA = e $\mapsto$ (CARDIOLOGIST,, ((READ, past heart records), heart treatment),
   ((Time (u)$\geq$ 8am $\wedge$ Time (u) $\leq$ 5pm) $\wedge$ Consent (u)=yes), Notify (David)=yes)

4. The Privacy sensitive policy to Task "f" Assignment:
   PTA = f $\mapsto$ (CARDIOLOGIST,, ((READ, past heart records), heart treatment),
   ((Time (u)$\geq$ 8am $\wedge$ Time (u) $\leq$ 5pm) $\wedge$ Consent (u)=yes), Notify (David)=yes)

5. The Privacy sensitive policy to Task "h" Assignment:
   PTA = h $\mapsto$ (CARDIOLOGIST,, ((READ, past heart records), heart treatment),
   ((Time (u)$\geq$ 8am $\wedge$ Time (u) $\leq$ 5pm) $\wedge$ Consent (u)=yes), Notify (David)=yes)

5.5 Summary

We presented in this chapter, the definition of purpose. We defined the abstract model of purpose as a planning of tasks [11]. It is defined by a set of tasks and the relationships among them. The abstract model of purpose is then expressed in the form of directed task graph (see Section 5.2.1). Given that the property of task graph has something in common with workflow, finally, we use workflow as a formal representation of purpose. Workflow is generally specified as a set of tasks and a set of dependencies among the tasks, and the sequencing of these tasks is also important. This property matches well to the meaning of purpose we defined. Since workflow is used as a formal presentation of purpose, in this chapter, we briefly introduced the workflow model where the workflow definition and some of its properties were presented. In that, we presented the resources management rules and workflow statuses. It is worth noting that since our main addressing issue is to control the access to resources when user executes a particular task of the workflow, our work mainly connects to resource management rule. In workflow, the resources, which are attached to each task, are not limited to digital resources (e.g. digital files); they can be anything. However, in this thesis, we focus only on the management of the digital resources, other than that is not in the scope of our work. To formally model resource management policy in workflow, we proposed an access control model for resources in workflow, where each task of the workflow is mapped onto a policy. In policy, each permission on resources is assigned to role and access authorisation is further enforced with conditions and obligations. To ensure that user uses resources for the purpose he claims, we need an access control policy enforcement. The enforcement technique will be presented in Chapter 6.

Chapter 6

Enforcing Purpose for Privacy-aware Policies

In Chapter 5, we introduced the purpose modelling and access control model. In this chapter, we address the issue of *purpose* enforcement for privacy-aware policies based on the purpose model and access control model presented in Chapter 5. We propose an approach to enforce *purpose* of access in access control system that uses workflows. In our approach, the access authorisation is based on the estimation of the level of certainty of *purpose*[1] achievement, which is determined by *purpose* achievement prediction [2] module. The prediction module is built using association rule learning method where user's access history and contextual information are used as the input data for rule analysis. We argue that by using the combination of contextual information and *purpose* achievement prediction, we can get a reliable purpose enforcement technique. The rest of this chapter is organised as follows. We discuss the privacy-aware policy enforcement in Section 6.1 in which we outline the purpose enforcement issues. The result from our survey on different prediction and forecasting methods is also presented in Section 6.1. Section 6.2 dedicates to purpose enforcement techniques. Section 6.3 is about purpose achievement prediction and prediction value estimation. Section 6.4 is about related work. Finally, Section 6.4 is the summary.

6.1 Enforcing Purpose for Privacy-aware Policies

In this section, we discuss the issues of privacy policy enforcement and the related work.

[1] *Purpose* here refers to claimed *purpose*, which is the *purpose* of accessing data.

[2] *Purpose* achievement prediction is a probabilistic system estimating how likely user can reach his claimed *purpose* after access permission is granted.

6.1.1 Issues

Enforcing *purpose* of access has been a subject of study in many research literatures [42][75][89][6][35] that address the issues of security and privacy protection for systems dealing with sensitive private data such as healthcare information system [7].

In general, enforcing *purpose* of access means to ensure that user uses requested data complying with claimed *purpose* and the data is not further used for other unauthorised *purposes*. There are two main parts for *purpose* enforcement [42].

1. "Verification" is a process to prove that user has the right to use data for the purpose he claims.

2. "Validation" refers to a process to prove that user can really achieve the *purpose* he claims once access permission is granted.

Purpose enforcement is a complicated task. The intuition behind this is that in most situations, it is not possible to get 100 percent certainty for *purpose* validation. This is because *purpose* is similar to the future goal that user claims to achieve. Normally, we do not know for sure, at the time of request, whether user uses data complying with claimed *purpose* or not once access permission is granted. It is possible that user claims to access the data for a *purpose*, but after permission is granted, he actually uses those data for other unauthorised *purposes*. With above illustration, we can see clearly the necessity to have a technique being able to predict the future achievement of claimed *purpose*. Given that, we propose a *purpose* achievement prediction technique built using the association rule learning [4] method. This technique is able to tell how likely that user can reach his claimed *purpose* given current contextual information and user's past access history as background check. In section 6.1.3, we provide a survey of different prediction and forecasting methods and a reason why we use association rule learning, but not other methods.

6.1.2 Survey of Different Prediction Methods

Before arriving at the conclusion of using association rule learning method for analysing the access log of user, we have studied different prediction and forecasting methods, such as Markov Decision Process [65], Naive Bayes [34], Logistic Regression [47], k-nearest neighbor algorithm [21] and Decision Tree [67]. However, we find that among them, some can be used in our context: Markov Decision Process, Decision Tree, Association Rule Learning and Naive Bayes, but with different degree of effectiveness. We discuss them in detail below. We will point out their advantages and disadvantages when used in our context. We illustrate their weakness through example. Other methods like Logistic Regression and k-nearest algorithms do not fit to our context given our problem formulation and the data type and structure.

6.1.2.1 Markov Decision Process

Markov decision processes (MDPs) [65] MDPS offers a mathematical framework in order to model the decision making in cases where outcomes are partly under the control of a decision maker and partly random. MDPS is a discrete time stochastic control process. In MDPS, at each time step, when the process is in some state "s" the decision maker can choose any action "a" that is available in state "s". At the next time step, the process reacts by randomly moving into a new state "s' ", and it gives a corresponding reward $R_a(s, s')$ to the decision maker. The probability that the process

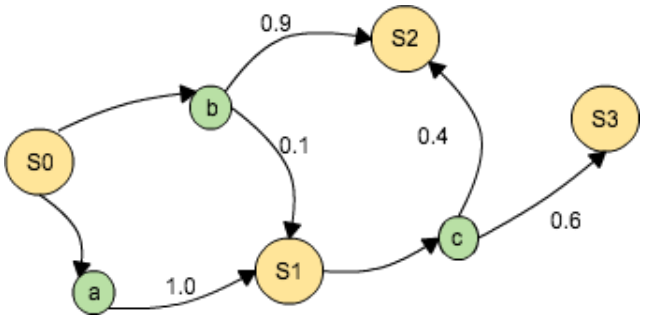

Figure 6.1: Example of MDP with 4 states and 3 actions.

moves into its new state "s' " is influenced by the chosen action and it is given by the state transition function $P_a(s, s')$. Thus, the state "s' " depends on the current state "s" and the action "a" chosen by the decision maker. Markov decision processes are an extension of Markov chains; the difference is the present of actions (allowing choice) and rewards (giving motivation). Conversely, if only one action exists for each state and all rewards are the same (or reward is zero), a Markov decision process reduces to a Markov chain [51].

The main problem of MDPs is to find a "policy" for the decision maker: a function π that specifies the action $\pi(s)$ that the decision maker will choose when in state s.

The goal is to choose a policy π that will maximise some cumulative function of the random rewards:

$$\sum_{t=0}^{\infty} \gamma^t R_{a_t}(s_t, s_{t+1})$$

where we choose $a_t = \pi(s_t)$ and γ is the discount factor and satisfies $0 \leq \gamma < 1$. For example, $\gamma = 1/(1 + r)$ when the discount rate is r. γ is typically close to 1.

How to apply MDP in our case. We can model workflow as MDP process where task represents the state in MDP and actions allowed in each task represent the actions

in MDP. The transition probability from one state to the others can be calculated using user's past access history. In other words, the MDP model is built using the past access of user as the trained data for the model.

When a user requests to execute a task for a particular purpose, we can consider the requested task as the initial state of MDP and the last task in a set of tasks representing the purpose as the ending state. What we want to determine is the "policy [1]" for the decision maker: a function π that specifies the action that the decision maker will choose when in state s. $\pi(s)$ must contain all the tasks and actions that need to be executed in order to reach the claimed purpose. In other words, $\pi(s)$ should be matched to the workflow definition representing the claimed purpose.

In general, MDP policy $\pi(s)$ contains a set of actions and states that policy maker needs to choose in order to reach a goal. What is in the policy is the optimal path or the best choice given by MDP. Considering in our case where a task is a part of different purposes (see Figure 5.1), user may execute that task for different purposes with different frequency. If we use MDP, the policy always points to the purpose that user executes more often and if we use MDP as the only decision factor, it may provide the wrong answer. In order words, when user requests access to less frequent executed purpose, the system will always reject the request because MDP always considers optional path (the most visited path) as a choice.

MDP can be used in conjunction with other model such as association rule learning to re-enforce the decision, but it could not be used as the single decision factor for access authorisation to address our problem. Other drawback of MDP is the performance, MDP requires significant processing time since in order for MDP to function, we need to have the transition probability. The transition probability, in our case, are calculated based on user's past access history. The time required to perform the transition probability calculation increases proportionately to the size of the access log and the complexity of the workflow. This means that the bigger access log the longer time required to calculate the transition probability. According to our experimental work [23], we find that MDP needs significantly more processing time compared with that of association rule learning method.

6.1.2.2 Decision Tree Learning

Decision tree learning [67] is a method, used in data mining, to create a model that predicts the value of a target variable based on a set of input variables. Decision tree learning uses a decision tree as a predictive model which maps observations about an item or a set of items to conclusions about the item's target value. In the tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. In decision analysis, a decision tree can be used to represent

[1] Policy, here, refers to a set of states and actions that user needs to follow and execute in order to reach a given state.

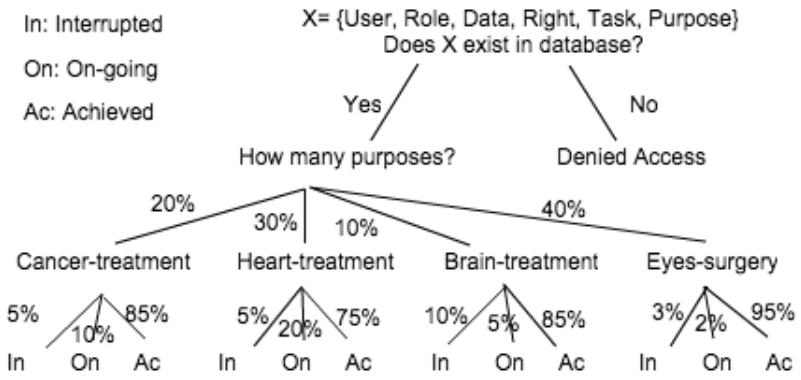

Figure 6.2: Example of decision tree with 4 purposes.

decisions and decision making. An example is shown Figure 6.2. Each interior node corresponds to one of the input variables; there are edges to children for each of the possible values of that input variable. Each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf. There are three types of leaf: Interrupted, On-going and Achieved.
The decision tree can be formulated as:

$$(\mathbf{X}, Y) = (x_1, x_2, x_3, ..., x_k, Y)$$

Where the dependent variable, Y, is the target variable that we want to understand or classify. The vector X is composed of the input variables, x_1, x_2, x_3, x_4 etc., they are used for that task.
Apply Decision Tree. We consider a vector X= {Username, Role, Task, Right, Data, Purpose} representing the features that need to be classified and the "workflow status" is considered as the target variable we want to understand. What we want to determine is the probability of Achieved, On-going and Interrupted given X (see Figure 6.2). The tree is constructed based on the past access history of user for a specific period of time. Decision tree can be used in our case, however, with large access history and highly complex workflow, the construction and analysis of decision tree is a challenge.

6.1.2.3 Naive Bayes

Naive Bayes [34] is a method for constructing classifiers. Naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be a banana if it is yellow and has a length about 20 CM. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is a banana,

regardless of any possible correlations between the colour and length. Naive Bayes is a conditional probability model: given a problem instance to be classified, represented by a vector $\mathbf{x} = (x_1, \ldots, x_n)$ representing some n features, it assigns to this instance probabilities $p(C_k|x_1, \ldots, x_n)$, for each of k possible classes.

$$p(C_k|\mathbf{x}) = \frac{p(C_k)\ p(\mathbf{x}|C_k)}{p(\mathbf{x})} \quad (1)$$

The naive Bayes classifier combines this model with a decision rule. One common rule is to choose the hypothesis that is most probable; this is known as the maximum posteriori decision rule. A Bayes classifier is the function that assigns a class label $\hat{y} = C_k$ for some k classes as follows:

$$\hat{y} = argmax_{k \in \{1,\ldots,K\}}\ p(C_k) \prod_{i=1}^{n} p(x_i|C_k) \quad (2)$$

Apply Naive Bayes. We consider a vector X= {Username, Role, Task, Right, Data, Purpose} representing the features that need to be classified. It is worth noting that X is actually the user's request query (see Section 6.2.1). The workflow statuses (C_k) (Achieved, Interrupted and On-going) are the classifiers. Given X and C_k, we can define the following Bayes functions.

$$\hat{y}_{Achieved} = p(Achieved) \prod_{i=1}^{n} p(x_i|Achieved).$$
$$\hat{y}_{Interrupted} = p(Interrupted) \prod_{i=1}^{n} p(x_i|Interrupted).$$
$$\hat{y}_{On-going} = p(On-going) \prod_{i=1}^{n} p(x_i|On-going).$$

We need to find the argmax[1]$(\hat{y}_{Achieved}, \hat{y}_{Interrupted}, \hat{y}_{On-going})$. The access is authorised if and only if $argmax(\hat{y}_{Achieved}, \hat{y}_{Interrupted}, \hat{y}_{On-going}) = \hat{y}_{Achieved}$. In order words, the user's request X is classified as "Achieved".

According to our study in [23], Naive Bayes classifiers is similar to association rule learning since both methods are built based on the basic probability concept [74]. However, in our context, association rule learning is more suitable. This is because Naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature. This is not the case since in our model where user is related to role, a user may have different roles and each role may contain many users; a task can be a part of different purposes; a data can be accessible for different task.

[1]Argmax is the set of points of the given argument for which the given function attains its maximum value.

This shows that the features are somewhat dependent. This special characteristic can cause Naive Bayes to provide the result which is different from what we intend since Naive Bayes classifier considers each of these features to contribute independently to the probability while association rule learning considers these features to contribute dependently to the probability. Moreover, in some cases, Naive Bayes provides wrong answer. In order words, the answer is different from the result we expected. We prove our claim by an example in discussion section (see Section 6.1.3.4). It is important to note that Naive Bayes can be used in our case if we classify our data set into subsets of features. That subset contains the features, which are dependent. Doing so, Naive Bayes produces the same result as that of association rule.

Transaction Id	Milk	Bread	Butter	Beer
1	1	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	1	1	0
5	0	1	0	0

Figure 6.3: Example database with 4 items and 5 transactions

6.1.2.4 Association Rule Learning

Association rule learning is a method for discovering relations between variables in databases. It aims to identify the relationship rules discovered in databases using different measures of interestingness. Based on the concept of strong rules [62], Rakesh Agrawal [4] introduced association rules for finding the regularities between products in large-scale transaction databases. For example, the rule $\{\text{butter}, \text{bread}\} \Rightarrow \{\text{milk}\}$ found in the sales data of a supermarket would mean that if a customer buys butter and bread together, he is likely to also buy milk. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product arrangement. An association rule has two parts, an antecedent (if) and a consequent (then). An antecedent is an item found in the data. A consequent is an item found in combination with the antecedent.

Association rules are created by analyzing data for frequency if/then patterns and using the criteria support and confidence to identify the most important relationships. Support is the frequencies of appearance of the items in the databases. Confidence represents the number of times the if/then statements have been found to be true.

Definition. The problem of association rule mining is defined as: Let I=$\{i_1, i_2, \ldots, i_n$ $\}$ be a set of n items. Let $DI = \{t_1, t_2, \ldots, t_m\}$ be a set of transactions. Each transaction in DI has a unique transaction identification (ID) and contains a subset of the items in I. A rule is defined as an implication of the form $X \Rightarrow Y$ where X, $Y \subseteq I$ and $X \cap Y = \emptyset$. The sets of items (for short itemsets) X and Y are called antecedent (left-hand-side or LHS) and consequent (right-hand-side or RHS) of the rule respectively.

Concepts. The confidence of a rule is defined as:

$$\text{conf}(X \Rightarrow Y) = \frac{supp(X \cup Y)}{\text{supp}(X)} \quad (3)$$

The support supp(X) of an itemset X is defined as the proportion of transactions in the data set which contain the itemset. An example in Figure 6.3, the itemset $\{\text{milk}, \text{bread}, \text{butter}\}$ has a support of 1/5=0.2 since it occurs in 1 out of 5 transactions. To illustrate the concepts, we use a small example from the supermarket domain. The set of items is I=$\{$ milk, bread, butter, beer$\}$ (1 codes presence and 0 absence of an item in a transaction) is shown in Figure 6.3. An example rule for the supermarket could be $\{\text{butter}, \text{bread}\} \Rightarrow \{\text{milk}\}$ meaning that if butter and bread are bought, customers also buy milk. For example, the rule $\{\text{butter}, \text{bread}\} \Rightarrow \{\text{milk}\}$ has a confidence of 0.2/0.2=1.0 in the database, which means that for 100% of the transactions containing butter and bread the rule is correct (100% of the times a customer buys butter and bread, milk is bought as well). Confidence can be interpreted as an estimate of the probability P(Y|X), the probability of finding the RHS of the rule in transactions under the condition that these transactions also contain the LHS.

Apply association rule learning. We consider a vector X= {Username, Role, Task, Right, Data, Purpose} representing the itemset. The workflow statuses (C_k) (Achieved, Interrupted and On-going) is another itemset. What we want to determine is the relation between X and C_k. Given X and C_k, we can define the following rule confidence.

$$\text{conf}(X \Rightarrow Achieved) = \frac{supp(X \cup Achieved)}{\text{supp}(X)}$$
$$\text{conf}(X \Rightarrow Interrupted) = \frac{supp(X \cup Interrupted)}{\text{supp}(X)}$$
$$\text{conf}(X \Rightarrow On-going) = \frac{supp(X \cup On-going)}{\text{supp}(X)}$$

We need to find the $argmax(\text{conf}(X \Rightarrow Achieved), \text{conf}(X \Rightarrow Interrupted), \text{conf}(X \Rightarrow On-going))$. The first access authorisation step is considered as successful if and only if $argmax(\text{conf}(X \Rightarrow Achieved), \text{conf}(X \Rightarrow Interrupted), \text{conf}(X \Rightarrow On-going)) = \text{conf}(X \Rightarrow Achieved)$. The argmax equals confident of "achieved" means that the vector X is classified as "Achieved". After finding that X is related to "Achieved", we need to compare the confidence of $\text{conf}(X \Rightarrow Achieved)$ to the required confidence value defined in access control policy. If it is greater or equal, the access request is authorised, otherwise, the access request is rejected.

Workflow s-Id	User	User-role	Task	Right	Object	Purpose	Status
1	David	Physician	b	Read	D1	Heart-treatment	Interrupted
2	David	physician	b	Read	D1	Heart-treatment	Achieved
3	David	Physician	b	Read	D1	Heart-treatment	On-going
4	David	Physician	b	Read	D1	Heart-treatment	Achieved
5	David	Physician	b	Read	D2	Heart-treatment	Achieved
6	David	Physician	a	Modify	D3	Brain-treatment	Achieved
7	David	Physician	a	Modify	D3	Brain-treatment	Achieved
8	David	Physician	a	Modify	D4	Brain-treatment	Achieved
9	David	Physician	a	Modify	D4	Brain-treatment	Achieved

Figure 6.4: Example database (access history) with 7 items and 9 transactions.

6.1.2.5 Discussion: Naive Bayes and Association Rule Learning

We claim that association rule learning is better than Naive Bayes classifier if we do not classify our data set into subsets of features that are dependent as discussed in Section 6.1.2.2. We prove by example in this section. Suppose that we have the database recording the user's activities when he executes workflow (see Figure 6.4). The database contains 8 items: workflow instance id, name of user, role of user, task, right on data object, object, purpose of access and the workflow status. Workflow status has three possible values: "Achieved", "Interrupted" and "On-going". In that database, there are 9 records.

At some point in time, user "David" in role "Physician" requests to execute task "b" and perform an right "Read" on data "D1" for purpose "Heart treatment". The provided information forms a user's access request "X".

X={David, Physician, b, Read, D1, Heart Treatment}
Y={Achieved, Interrupted, On-going}

For simplification we use the following abbreviation: David= Da, Physcian= Ph, Read= Re, Heart Treatment= HT, Achieved= Ac, Interrupted=In and On-going=On.

With this information, what we want to find out is the decision (permit or deny access) that system makes: (1) in case Naive Bayes is used and (2) the association rule is used.

Naive Bayes. Based on equation (2), we can define the following Naive Bayes functions.

$$\hat{y}_{Ac} = p(Ac)(Da|Ac)p(Ph|Ac)p(b|Ac)p(Re|Ac)p(D1|Ac)p(HT|Ac)$$

$$\hat{y}_{In} = p(In)p(Da|In)p(Ph|In)p(b|In)p(Re|In)p(D1|In)p(HT|In)$$

$$\hat{y}_{On} = p(On)p(Da|On)p(Ph|On)p(b|On)p(Re|On)p(D1|On)p(HT|On)$$

Based on the database in Figure 6.4, we can calculate the values of $\hat{y}_{Ac}$, $\hat{y}_{In}$ and $\hat{y}_{On}$, as follows.

$\hat{y}_{Ac} = \frac{7}{9}(\frac{7}{7} * \frac{7}{7} * \frac{3}{7} * \frac{3}{7} * \frac{2}{7} * \frac{3}{7}) = 0.0074$

$\hat{y}_{In} = \frac{1}{9}(\frac{1}{1} * \frac{1}{1} * \frac{1}{1} * \frac{1}{1} * \frac{1}{1} * \frac{1}{1}) = 0.1111$

$\hat{y}_{On} = \frac{1}{9}(\frac{1}{1} * \frac{1}{1} * \frac{1}{1} * \frac{1}{1} * \frac{1}{1} * \frac{1}{1}) = 0.1111$

$argmax(0.0047, 0.1111, 0.1111) = 0.1111$, thus, the request X is classified as either "Interrupted" or "On-going", the access is rejected.

Association Rule Learning. Based on equation (3), we can define the following rule confidence.

$$\text{conf}(\{Da, Ph, b, Re, D1, HT\} \Rightarrow Ac) = \frac{supp(\{Da,Ph,b,Re,D1,HT\} \cup Ac)}{\text{supp}(\{Da,Ph,b,Re,D1,HT\})}$$

$$\text{conf}(\{Da, Ph, b, Re, D1, HT\} \Rightarrow In) = \frac{supp(\{Da,Ph,b,Re,D1,HT\} \cup In)}{\text{supp}(\{Da,Ph,b,Re,D1,HT\})}$$

$$\text{conf}(\{Da, Ph, b, Re, D1, HT\} \Rightarrow On) = \frac{supp(\{Da,Ph,b,Re,D1,HT\} \cup On)}{\text{supp}(\{Da,Ph,b,Re,D1,HT\})}$$

Based on the database in Figure 6.4, we can calculate the following rule confidences.

$\text{conf}(\{Da, Ph, b, Re, D1, HT\} \Rightarrow Ac) = \frac{\frac{2}{9}}{\frac{4}{9}} = \frac{2}{4} = 0.50$

$\text{conf}(\{Da, Ph, b, Re, D1, HT\} \Rightarrow In) = \frac{\frac{1}{9}}{\frac{4}{9}} = \frac{1}{4} = 0.25$

$\text{conf}(\{Da, Ph, b, Re, D1, HT\} \Rightarrow On) = \frac{\frac{1}{9}}{\frac{4}{9}} = \frac{1}{4} = 0.25$

$argmax(0.25, 0.50, 0.25) = 0.50$, thus, the request X is classified as "achieved", the access is granted.

From that example, we can see clearly that Naive Bayes and Association Rule provide contradicting decision. According to Naive Bayes, the access request is rejected while association rule provides positive response. But, in our context, which method is suitable to be used? According to the information provided in database (see Figure 6.4), the access request "X" should be granted since among 4 workflow instances (1, 2, 3 and 4) that David has executed, two of them were completed successfully. This implies that the success rate is relatively higher compared with Interrupted and On-going. Naive Bayes provides negative response because Naive Bayes classifier considers each of these features (X) to contribute independently to the probability and since David has achieved other purposes more frequently with some of the features in X, this contributes negatively to the probability. Other drawback of Naive Bayes is the processing time. Naive Bayes has the complexity of O(K*(N-1)), where N is the number of features and K is the number of records in database. Association Rule has the complexity of O(K), where K is the number of records in database.

Naive Bayes could provide the same conclusive result to that of association rule if we classify the data set into subset of dependent features. However, classifying the data into subsets of features requires a pre-processing of data; hence, it is a time consuming process. This can result into a bigger processing time compared with association rule. For Decision Tree, it bears the same challenge to that of Nave Bayes since in order for decision tree to work we need to construct the three. With large access history and highly complex workflow, the construction and analysis of decision tree is a time consuming process. Thus, we conclude to use Association Rule for analysing the user's past access log over Naive Bayes and Decision Tree since Association Rule is a simple process. Association Rule discovers the relationship between features in database by counting frequency and then calculating the confident of that relationship based on the frequency that relationship appears in database.

6.2 Purpose Enforcement

In this section, we focus on access authorisation and purpose enforcement expression.

6.2.1 Concept: access request, authorisation and policy enforcement

User request: in workflow information system, when a user wants to execute a task for a particular purpose, he needs to provide the following information: username, role, task, right, data, purpose (see Section 5.4.1). This information forms the user's request. With the provided information in user's request, system validates the request

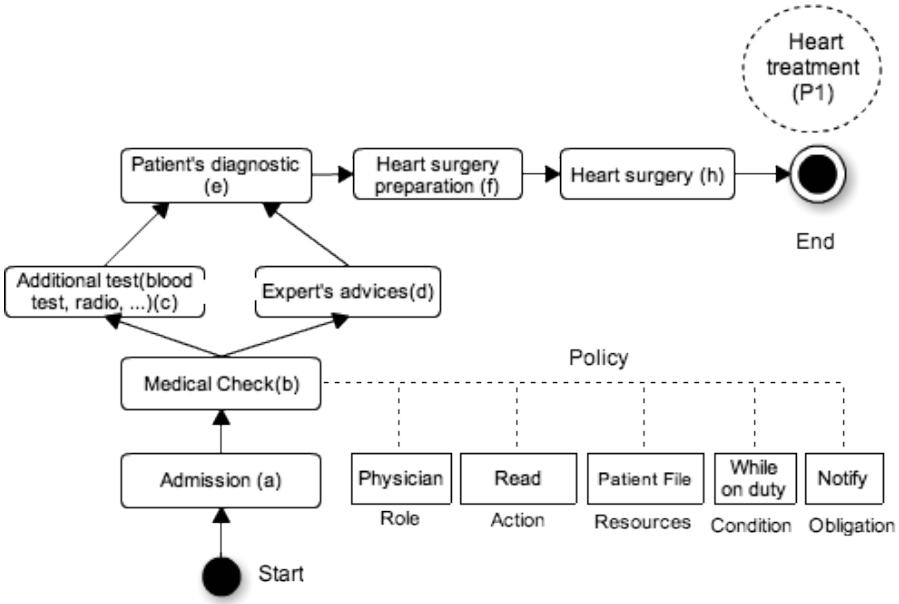

Figure 6.5: Example of workflow representing heart treatment *purpose* (P1). Figure 6.5 is derived from Figure 5.1.

based on the access control policy (see Section 5.4.1) applied to the requested task of the workflow. If the request is valid, user is granted access to data. There are two processing steps when validating an access request (see Section 6.1.1): verification and validation. In Figure 6.6, in verification process, system matches the information provided in user request to that of access control policy applied to the requested task of the workflow. User must be in role authorised to execute the task. The task must be a part of the workflow definition representing the purpose user claims. The requested data must be a part of the resources allocated to the task. Finally, the requested right (or operation) must be the one, which is authorised to perform on requested data. If all the attributes are matched, the system proceeds to purpose validation.

Purpose validation: as mentioned earlier, we use two types of information to enforce the access control to data: contextual information related to task and purpose and the purpose achievement prediction. The contextual information related to task and purpose defined in access control policy must be valid at the time of access. For example, if working-hour is a contextual information, user is authorised to execute the task only during his working-hour. Once contextual information is valid, system needs to calculate the purpose achievement prediction value using user's past access history.

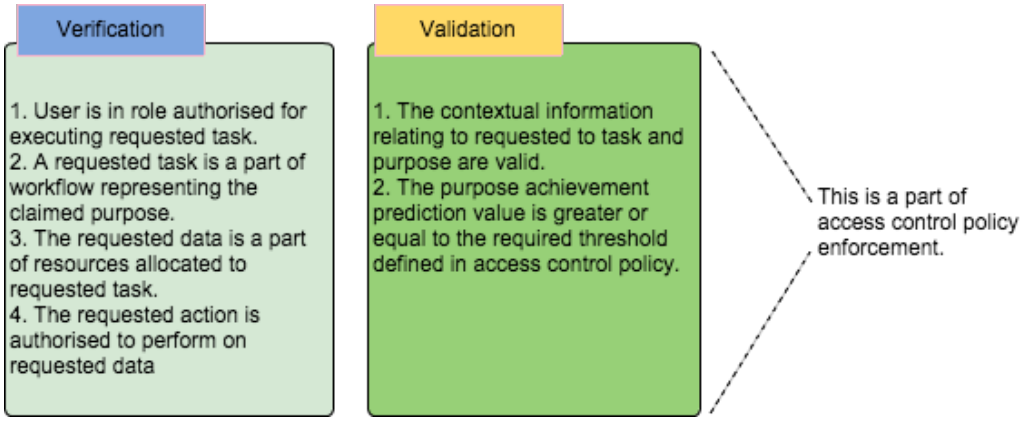

Figure 6.6: Access control policy verification and validation process.

6.2.2 Purpose Enforcement Expression

The purpose achievement prediction module, a probabilistic system, estimates how likely it is that user can achieve their claimed *purpose*. The estimation is done by using association rule learning method [4] with the support of user's past access history. The required prediction value, a probabilistic value, is expressed as condition in policy by policy-maker. Then, during policy validation, the estimated value defined in policy is compared with the value generated by *purpose* prediction module at the time of request. For example, if the policy requires 0.9 of certainties, the access is allowed if and only if the estimated value provided by *purpose* prediction module, at the time of request, is greater or equal 0.9. The second *purpose* enforcement parameter is the contextual information. Each task or *purpose* is associated with a set of contextual variables. When user requests to execute a task for a particular *purpose*, the contextual information for the requested task and claimed *purpose* need to be validated. The contextual variables are also expressed as conditions in policy.

6.2.2.1 Contextual Information

Contextual data [84] are the information surrounding user, data, task and *purpose*. Contextual data can be anything, such as user's personal data, location, time or other environment information. For example, in Figure 6.7, the physical location of user can be considered as contextual information. Each *purpose* or task has its own set of contextual variables that are determined based on specific system's requirements.

Definition 6.1: Contextual variables to task and purpose mapping

---

Let "p" be a *purpose*. Let V be a set of contextual variables (v) where $v \in V$. Let "t"

Admission

Context Information	Description
Physical location of user	Receptionist or phsycian who is responsible for admitting patient to hospital must physically present in hospital when accessing to required data of patient. The identity of the device used to access to system can be used to indicate the physical location of user.
Physical location of patient	To admit a patient to the hospital, patient registration is required. Thus, patient needs to physically present at hospital.
Insurrance information	Before admitting to the hospital, health insurance verification is required.

Figure 6.7: Example of contextual information for task "Admission", refer to Figure 6.5 for more details.

be a task. Then, we can define the following functions.

PF: $p \mapsto V$ is a function that maps a set of contextual variables V to *purpose* "p".

TF: $t \mapsto V$ is a function that maps a set of contextual variables V to task "t".

Basic language for expressing the contextual variables (LC)

In this section, we include a simple language for expressing contextual variables in policy. As mentioned earlier, the contextual variables are expressed as conditions in access control policy (see Chapter 5, Section 5.4).

Definition 6.2: Language for expressing V

Let v be a contextual variable where $v \in V$. "v" has the possible values, denoted as D_v. "v" is equipped with the relational operators (Oprs) " $=, \neq, \geq, and \leq,$". The condition of v has the form (v opr d_v), where $d_v \in D_v$ and $opr \in Oprs$. For example, suppose that payment-amount is the contextual variable and the payment needs to be greater or equal 20$, we can write as follows. payment-amount $\geq$ 20$.

let v_1 and v_2 be two contextual variables in the form of the atomic condition. Then, $(v_1 \wedge v_2)$ or $(v_1 \vee v_2)$ is also condition.

Contextual information mining for task and purpose. Normally, contextual variables vary depending on the nature of task or purpose. Finding the contextual variables is a case-by-case study basic. However, a rule of thumb is, firstly, to de-

fine the meaning of task or purpose. Then, we find the requirements that need to be checked when executing it. An example in Figure 6.7, "admission" consists of three contextual variables: physical location of user (e.g. physician), physical location of patient and information about insurance. Those contextual variables are important in order to execute the task "admission"; without such information, the task cannot be executed. For more details, one can refer to [84].

Workflow s-Id	User	User-role	Task	Right	Object	Purpose	Status
1	David	Cardiologist	b	Read	D1	Heart-treatment	Achieved
2	David	Cardiologist	b	Read	D1	Heart-treatment	Achieved
3	David	Cardiologist	a	Read	D2	Achieved	Achieved
4	John	Physician	a	Modify	D3	Brain-treatment	Achieved
5	John	Physician	a	Modify	D3	Brain-treatment	Interrupted
6	John	Physician	a	Modify	D4	Brain-treatment	Achieved
7	John	Physician	a	Modify	D4	Brain-treatment	Achieved

Figure 6.8: Example database (access history) with 7 items and 7 transactions.

6.2.2.2 Purpose Achievement Prediction

The objective of *purpose* achievement prediction module is to say, given user's past access history, how likely it is that he would achieve his claimed purpose when he requests to execute a particular task for that purpose. An example in Figure 6.8, if cardiologist "David" requests to execute task "medical check" for patient "Edward", for *purpose* of heart treatment; then, the prediction module should tell, after analysing David's past access history, the probability that David could complete heart treatment after executing task "medical check".

Definition 6.3: purpose achievement prediction expression

Let PA be a purpose achievement prediction variable. PA has possible values, denoted as D_{PA}; where $D_{PA} = [0, 1]$. PA is equipped with the relational operators (Oprs) " $=, \neq, \geq, and \leq$ ". The condition of PA has the form (PA opr d_{PA}), where $d_{PA} \in D_{PA}$ and $opr \in Oprs$. For example, PA is greater or equal 0.70 ($PA \geq 0.70$).

There are two issues:

1. How to compute PA's value? When user requests to execute a task for a purpose, system needs to compute PA. The PA's value, which is computed at the time of request, is then compared to the threshold (PA) defined in policy.

2. How to determine the threshold of PA? How policy maker know which PA's value is good for their system?

The two issues are presented in detail in Section 6.3.

6.2.3 Example: Access Policy Expression with PA

We use the example from Figure 6.5. Suppose that there is an access control policy for *purpose* HEART-TREATMENT (P1) at task "medical check (b)". The rule states that every user in role CARDIOLOGIST can READ a patient's HEART TREATMENT HISTORY for *purpose* of P1, if and only if user is present in the hospital at the time of access and the required *purpose* achievement prediction is greater than or equal to 0.95. Moreover, each time CARDIOLOGIST requests access, patient needs to be notified.

Suppose that "user-location" is the contextual variable representing the location of cardiologist and PA is the variable representing the estimated value of *purpose* achievement prediction. "Notify" is the obligation that user needs to fulfil. Then, we can define the following policy (see Section 5.4 for policy expression).

1. PDPA to role "CARDIOLOGIST":
   PDPA= (CARDIOLOGIST, ((READ, HEART TREATMENT HISTORY), HEART-TREATMENT)), (user-location= in-hospital $\wedge PA \geq 0.95$), Notify)

2. The Privacy sensitive policy to Task Assignment:
   PTA = b $\mapsto$ (CARDIOLOGIST, ((READ, HEART TREATMENT HISTORY), HEART-TREATMENT), (user-location= in-hospital $\wedge PA \geq 0.95$), Notify)

It is worth noting that the bigger value of PA, the higher workflow succeeded rate of the system. The required value of PA is generally defined in policy by policy-maker and how to define that value will be presented in Section 6.3.4.

6.3 Calculating PA Value

In this section we present in detail how the system calculates the value of PA at the time of request and how policy maker define PA's value in policy (determining PA's threshold).

6.3.1 Apply Association Rule Learning

Based on our study in Section 6.1.3, we conclude that Association Rule Learning method is suitable to be used in our context. Thus, in this section, we present in detail how we apply association rule to predict the purpose achievement. Since we use access history of user in a system that uses workflows as inputs for rule analysis, we introduce access-log structure defined specifically to be used in our system context.

6.3.1.1 User Access History Structure and Data Items

When user requests to execute a task for a particular *purpose*, system needs to log the necessary information for later use. Based on our modelling of *purpose* and the policy definition (see Section 5.4), the data items, that system needs to record in access history, are: the unique identification of workflow instance (i), user, role, task, right, data, purpose and workflow instance status.

Definition 6.4: access history structure

We define the following data items in access history: (i, u, r, t, g, d, p, s), where "t" is a task, "s" refers to the status of workflow. (u, r, g, d, p) are the elements of policy we defined in Section 5.4. "i" is a workflow instance identification number.

6.3.1.2 PA value Calculation based on Association Rule Learning

The PA value (see Definition 6.3) is actually the value of rule confidence, the confidence of the relationship between the data provided by user in access request with the status of the workflow instances that user has executed in the past.

Let J={u, r, t, g, d, p, s} be a set of items. Let H=$\{i_1, i_2, \ldots, i_m\}$ be a set of workflow instance identifications. Each identification, in H, contains a subset of the items in J. What we want to find is the confidence of the following rule.

$\mathbf{conf}(\{u, r, t, g, d, p\} \Rightarrow Achieved)$
$\mathbf{conf}(\{u, r, t, g, d, p\} \Rightarrow On - going)$
$\mathbf{conf}(\{u, r, t, g, d, p\} \Rightarrow Interrupted)$

In order for the system to grant access for a particular access request, the confidence of the rule, {u, r, t, g, d, p} ⇒ Achieved, must be bigger than that of $\mathbf{conf}(\{u, r, t, g, d, p\} \Rightarrow On - going)$ and $\mathbf{conf}(\{u, r, t, g, d, p\} \Rightarrow Interrupted)$. Moreover, the confidence of the rule, $\mathbf{conf}(\{u, r, t, g, d, p\} \Rightarrow$ Achieved), must be greater or equal to the PA's value defined in policy.

Log-elements	Abreviation	Description
Workflow Instance Id	i	The unique identification of workflow instance
User-name	u	The unique identification of user
Role	r	The role of user
Task	t	The task is an element of workflow
riGht	g	The right/operation allowed on data
Data	d	Data is the resource allocated for task
Purpose	p	Purpose of using data
Workflow Status	s	The status of the workflow

Figure 6.9: Log Structure.

For example (see Figure 6.8), David, in role cardiologist, requests to execute task "b", a right "Read" on data "D1", for *purpose* of heart treatment. Suppose that M is a set of items containing {David, Cardiologist, b, Read, D1, Heart-treatment} (this data set is actually David's request query (see Section 5.4.2)). Then, based on the database in Figure 6.8, we can calculate the confidence of the rule.

$\mathbf{conf}(M \Rightarrow Achieved) = \frac{supp(M \cup Achieved)}{\text{supp}(M)} = \frac{2/7}{2/7} = 1$
$\mathbf{conf}(M \Rightarrow Achieved) = \frac{supp(M \cup On-going)}{\text{supp}(M)} = \frac{0/7}{2/7} = 0$
$\mathbf{conf}(M \Rightarrow Achieved) = \frac{supp(M \cup Interrupted)}{\text{supp}(M)} = \frac{0/7}{2/7} = 0$

This means that the rule $\mathbf{conf}(M \Rightarrow Achieved)$ is 100% true since every time M appears in the databases, "Achieved" is also appeared. In other words, every time David requests access to data for that *purpose*, the claimed *purpose* is achieved. In this example, confidence of the rule $\mathbf{conf}(M \Rightarrow On-going)$ and $\mathbf{conf}(M \Rightarrow Interrupted)$, both are zero because, in database (see Figure 6.8), M does not appear with either "On-going" or "Interrupted".

6.3.1.3 Algorithm: Calculate Support and Rule Confidence

In this section we introduce an algorithm for calculating the support "supp()" and the confidence of the rule.

Terminology. DS is the data sequence. A-log is the access-log. Minimum support (for short min-supp(M)) is the required minimum number of M appearing in the database [5] .

In order to calculate the confidence of the rule, **conf**$(X \Rightarrow Y)$, we need to calculate the support $supp(X \cup Y)$ and supp(X) (see Section 6.1.3.3).

Definition 6.5: Data Sequence in access-log

Let DS be the data sequence containing the following data items: {i, u, r, t, g, d, p, s}. Let DS-1, DS-2, DS-3 and DS-4 be the sub-sequences of DS where DS-1= {i, u}, DS-2={u, r, t, g, d, p}, DS-3={u, r, t, g, d, p, s} and DS-4={s}.

The algorithm: we split the problem of mining the support and calculating the rule confidence into the following phases:

1. **Sort Phase**: A-log is sorted, with user-id (u) as major key and workflow instance identification (i) as the minor key. The DS-1 is used in this case. This step implicitly converts the global access-log into an access-log of user (AU).

2. **Mining support DS-2 Phase**: counting the number of occurrences of DS-2 in AU. The support of a data sequence is defined as the fraction of transactions in which a data sequence is present.

3. **Mining support DS-3 Phase**: counting the number of occurrences of DS-3 in AU.

4. **Rule confidence Phase**: calculating the rule confidence based on the information in Phase 2 and 3 and the defined min-supp. If number support of DS-3 is less than min-supp, the rule confidence **conf**(DS-2 ⇒DS-4) is invalid (or zero). If number of supp(DS-3) is greater or equal min-supp, the rule **conf**(DS-2 ⇒DS-4)= supp(DS-3)/supp(DS-2).

The algorithm 1 provides the detailed calculation of the support of data sequence and rule confidence.

Remark: Association Rule with non-existing data.

As user's past access activities are used as the input data for rule analysis, there is a drawback if we calculate PA's value based entirely on association rule. This is because association rule fails in case of new user who does not have any access history or new

data that user has never accessed. Taking an example in healthcare information system, suppose that there is a new patient admitted to the hospital for the first time, a physician is assigned to this new patient, but he has never treated this new patient; hence, physician does not have any past access to the health records of the patient. If physician requests access to this patient's health records, the system will always deny his request; this is because patient's health records does not appear in access log of physician. How to solve this problem? We address this problem in Section 6.3.2. We developed a method with the support of association rule learning to solve the problem we described above.

Algorithm 1 Calculating support and rule confidence **conf**(DS-2 ⇒DS-4)

```
Inputs: A-log, min-supp, DS, DS-1, DS-2, DS-3, DS-4
Outputs: Rule-conf // rule confidence
Variables: AU // access-log of a user (U).
Rule-conf ← 0
// Begin sorting phase
while Not the end of A-log do
   Search A-log and select data from A-log using DS-1 as key (see Definition 7.5).
   Insert the selected data to AU
end while
// End sorting phase
//Begin calculating support DS-2 and DS-3
while Rule-conf==0 and not the end of AU do
   Count supp(DS-2)
   Count supp(DS-3)
end while
// End calculating support DS-2 and DS-3
// Begin calculating rule confidence
if if supp(DS-3) >= min-supp then
   Rule-conf ←DS-3/DS-2
end if
//End calculating rule confidence
Return Rule-conf
```

6.3.2 PA's Value Calculation

According to Definition 6.4, PA's value is given by a function that analyses the past access history of user using association rule learning method, based on past access

analysis variables. Past access variables are the units of data used for analysing the access history of user.

6.3.2.1 Past Access Analysis Variables

We define four past access analysis variables.

- **Past access of user for claimed *purpose* with requested data object (PAV-1)**: this past access variable is used to find out if user has ever accessed the requested data for the claimed *purpose* or not. If he did, was the claimed *purpose* achieved? The confidence of the rule, $\mathbf{conf}(\{u, r, t, g, d, p\} \Rightarrow Achieved)$, is calculated. If he has never accessed requested data with claimed *purpose* or the confidence of the rule is below the expected value defined in policy, we proceed to second variable; otherwise system returns the "valid" response with the rule confidence value.

- **Past access of user for claimed *purpose* with other data (PAV-2)**: this past access variable allows us to observe the past access of user indirectly (not with the data being requested). For example, a patient (Charlie) comes to hospital for heart-treatment. A cardiologist (David) is assigned for Charlie. Then, before treatment, David requests access to Charlie's health records for heart-treatment *purpose*. David has never treated Charlie; hence, no past access records to Charlie's health records. However, David has experiences with other patients with the heart-treatment *purpose*. We consider his credits with other patients for the heart treatment *purpose* and take it into account. Again we need to calculate the confidence of the rule (claimed purpose with other data) and if the confidence of the rule is below minimum requirement, we proceed to the next variable; otherwise system returns the "valid" response with the rule confidence value.

- **Past access of user for other *purpose* (having relationship with the claimed *purpose*) with requested data object (PAV-3)**: this past access variable allows us to observe the past access of user indirectly, particularly the observation based on the relationship between claimed *purpose* and other *purposes*. In our modelling, a *purpose* may be a subset of other *purpose*. For example, Figure 5.1, "brain surgery preparation" may be considered as a subset of "brain surgery". If user requests access for "brain surgery" and he has good records for *purpose* "brain surgery preparation", it is highly likely that he would achieve "brain surgery" *purpose*. Like other past access variable, the confidence of the rule is calculated. If the confidence of the rule is below the minimum requirement, we proceed to the next variable; otherwise system returns the "valid" response with the rule confidence value.

- **Past access of user for other *purpose* with other data object (PAV-4)**: this variable is for *purpose* of verifying if user is a new user or the old one. The result from this analysis does not provide the conclusive response since it does not relate to the object and the *purpose* user claims. However, it can tell the past activities of user working with the system. The confidence of the rule, the rule that contains neither requested object nor claimed *purpose*, is calculated. If the rule is below the minimum requirement, the system returns the "invalid" response; otherwise system provides the valid response with precaution. Any response with precaution may need extra control.

6.3.2.2 PA's Value Calculation and Validation Algorithm

The detailed procedure for calculating the value of PA is presented in Figure.6.9. Suppose that we have a set of items {u, r, t, g, d, p} as antecedent (left-hand-side or LHS) and {status (s)} as consequent (right-hand-side or RHS) of the rule respectively.

- N is an integer representing the number of past access analysis variables. In this case, we define N=4.
- RPA is a required value of PA defined in Policy.
- VPA is a PA's value that system estimates at the time of request.
- DiffFrom(k) is a function that returns the value that is different from k. For example, DiffFrom(p) returns the *purposes* that are different from p.
- A-log is the access-log.
- p is *purpose* of access.
- R-d is the requested data (d) that user wants to access.
- Relationship(p) is a function returning the *purposes* having relationship with p.

Based on the definition of the four past access analysis variables defined in Section 6.3.2.1, we can define the following rules for the four past access analysis variables:

1. PAV-1: {u, r, t, g, R-d, p} ⇒ Achieved
2. PAV-2: {u, r, t, g, DiffFrom(R-d), p} ⇒ Achieved
3. PAV-3: {u, r, t, g, R-d, Relationship(p)} ⇒ Achieved
4. PAV-4: {u, r, t, g, DiffFrom(R-d), DiffFrom(p)} ⇒ Achieved

The algorithm 2 shows the detailed procedures for calculating and validating PA's value.

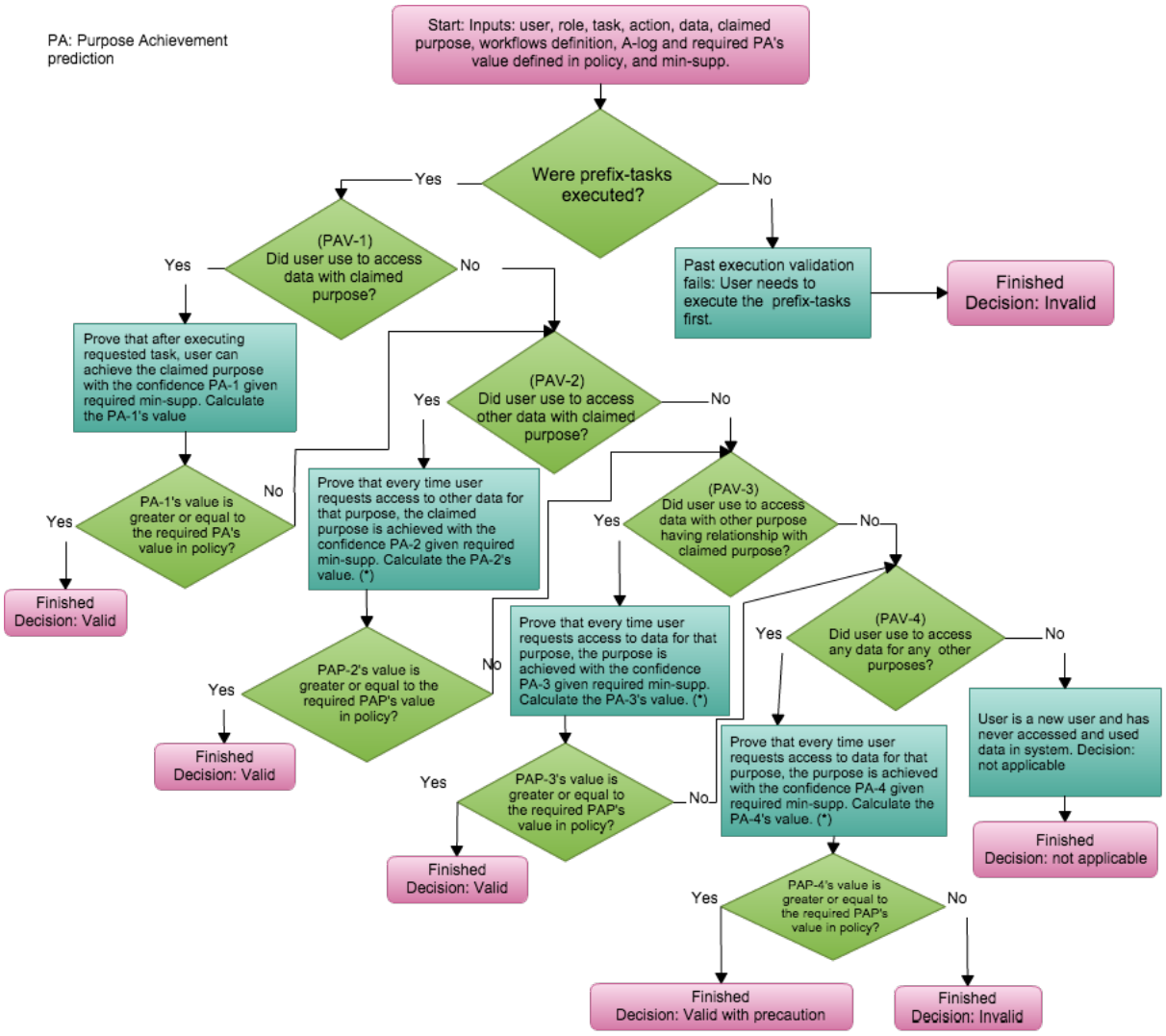

Figure 6.10: Flowchart: PA's value calculation and Validation

Algorithm 2 Calculating and validating PA's value

Inputs: p, A-log, RPA, R-d, PAV.
Outputs: Response
Variables: VPA
$Response \leftarrow Invalid$
$VPA \leftarrow 0$
$i \leftarrow 1$
while Response==Invalid and i<=N **do**
  Calculate the rule confidence of PAV(i) using algorithm 1 and see also PAV's rule structure in Section 6.3.2.2.
  $VPA \leftarrow Confidence(PAV(i))$
  if if VPA >= RPA **then**
    $Response \leftarrow Valid$
    Exist the loop.
  end if
  i←i+1
end while
Return *Response*

6.3.3 Example: PA's Value Calculation

In this section we provide an example on how to calculate the PA's value. Suppose that a workflow definition (a, b, c, d, e, g and i) (see Figure 5.1) represents "brain treatment" *purpose*. The policy mapped to task "a" is: PTA = a $\mapsto$ (Physician, ((modify, D3), brain-treatment), $PA \geq 0.75$). John (physician) wants to execute task "a" in order to modify data (D3) for *purpose* of "brain-treatment". According to the policy, John is permitted to access if and only if after analysing his past access records, the system returns the PA's value being greater or equal 0.75.

Suppose that we have the past access records (databases) like in Figure 6.8. Then, based on the algorithm 1 and 2 in Section 6.3.2, we can calculate PA's value as follows.

1. Past access of user with claimed *purpose* and requested data (PAV-1). Suppose that we have a set G={John, Physician, a, Modify, D3, Brain-treatment}. We need to find the confidence of the rule $\mathbf{conf}(G \Rightarrow Achieved)$.

   $\mathbf{conf}(G \Rightarrow Achieved) = \frac{supp(G \cup Achieved)}{\text{supp}(G)} = \frac{1/7}{2/7} = 1/2.$
   $\mathbf{conf}(G \Rightarrow On-going) = \frac{supp(G \cup On-going)}{\text{supp}(G)} = \frac{0/7}{2/7} = 0.$
   $\mathbf{conf}(G \Rightarrow Interrupted) = \frac{supp(G \cup Interrupted)}{\text{supp}(G)} = \frac{1/7}{2/7} = 1/2.$

   With 0.5 rule confidence, it is not possible because the policy requires 0.75 con-

fidences. Thus, we need to proceed to the next past access variable (PAV-2).

2. Past access of user for the claimed *purpose* with other data (PAV-2). G={John, Physician, a, modify, DiffFrom(D3), Brain-treatment} **conf**$(G \Rightarrow Achieved)$= $\frac{supp(G \cup Achieved)}{\text{supp}(G)} = \frac{3/7}{4/7} = 3/4 = 0.75$. With 0.75 rule confidence, John is permitted to access and further verification of other past access variables (PAV-3 and PAV-4) is not required.

6.3.4 Determining Threshold Value of PA

PA's threshold value, that policy maker uses as condition in policy, is determined based on the long time observation and learning of the system's access-log. As mentioned earlier, in access-log, each workflow is marked with three possible statuses: On-going, Interrupted or Achieved. The PA's threshold value is basically the number of achieved within a defined periods of observation. It is important to note that the observation time must be long enough and observation should be repeated number of times to have a correct PA's value. For example, within one-month observation of access-log, among 1000 workflow instances, there are 900 Achieved, 20 Interrupted and 80 On-going. Thus, the PA's threshold value is set to 900/1000= 0.90. The PA's threshold value may vary from time to time depending on users' activities in the system, hence, the adjustment should be done accordingly.

PA's threshold value depends on two factors: Interrupted and On-going. If the number of Interrupted and On-going increases more than the observed threshold for a particular user, system does not authorise user to access and auditing may need to be done in order to find the cause.

- *The increase of interrupted workflows*: there are two possible explanations. Firstly, the workflows are interrupted because of the change of procedure and user does not have bad intention to do so. For example, a doctor may wrongly diagnose patient and needs to change from one to another purpose. The first workflow instance execution may be marked as interrupted. Secondly, user creates many workflow instances in order to get access to data for an illegal *purpose*. He created many workflow instances, but he has never completed them.

- *The increase of on-going workflows*. there are also two possible explanations. Firstly, there are the increases of activities in the system. For example, the increased number of patient visiting doctors in hospital; hence, there are many on-going dossiers, which are being processed. Secondly, user creates many workflow instances in order to illegally access to data.

6.4 Related Work

Many *purpose* enforcement techniques have been proposed [38][68] [17][20]. However, they do not provide satisfactory solution to cope with many important issues in purpose enforcement. One of which is how to ensure that users use data complied with the purpose it intends for. Below are some works having direct connection with ours.

Byun et al [17][20] proposed a purpose-based access control of private data, a model that relies on the well-known RBAC [24] access control model. They use user's role as the condition to enforce *purpose* of access by mapping roles to purposes. However, this method is not so reliable and it is criticised to be inefficient in capturing *purpose* of an action since roles and *purposes* are not always aligned and members of the same organisational role may practice different *purposes* in their actions. For example, a user in role administrator can access customer's data for *purpose* of marketing and planning.

Carl et al [81] proposed an automated method for enforcing privacy policies. The authors modelled *purpose* as a planning of actions and they used a modified version of Markov Decision Processes (MDPs)[1] to determine if the requested action is really for claimed *purpose*. They argue that an action is for a *purpose* if and only if that action is part of a plan for optimising the satisfaction of that *purpose* under the MDP model. The proposed technique is for auditing. Auditing may be able to detect policy violations after-the-fact, but it cannot prevent unauthorised access. Thus, this technique is not suitable for controlling the highly sensitive data, such as health records. The difference between the author's work and our work is that we focus on pre-enforcement of *purpose* (a priori control of access), not a posteriori control (auditing).

Jafari et al [42] models *purpose* as the inter-related actions, which are expressed in the form of action graph. The action (proposed by Jafari et al) has similar meaning to "task" in our purpose modelling. In their method, the enforcement of *purpose* is achieved by looking at the actions having relationship with requested action. The authors argue that the *purpose* of an action is determined by its situation among other inter-related actions. For example, in Figure 5.1, if a cardiologist requests to execute the action "heart surgery preparation" for *purpose* of heart surgery, it is important to check other actions that need to be executed before "heart surgery preparation"; in this example "patient's diagnostic" needs to be executed first. The proposed technique does make sense, but it has many drawbacks. One of which is that it does not guarantee that user could achieve the claimed *purpose* (use data for the purpose he claims) after the permission is granted. It has no way to predict the future activities of user. User may have executed the previous tasks correctly, but he may now have the bad intention for the requested task and after the access permission is granted; he would never complete the next tasks in order to achieve claimed *purpose*. To complement the work of Jafari

[1] http://en.wikipedia.org/wiki/Markov_decision_process

et al, we introduce the *purpose* achievement prediction and contextual information as other dimensions to enforce the *purpose* of access.

The closest work to ours is the one proposed by Jafari et al [1] [6] that is similar to the work in [42]. They proposed an approach to enforce *purpose* in access control systems that use workflows. They proposed to encode *purposes* as properties of workflow; a *purpose* is mapped to a sequence of tasks. In the proposed technique, the access authorisation bases solely on the mapping between user's role and action in workflow. However, this method cannot work effectively with the model where one action is a part of the actions that can lead to different *purposes*. This is because we cannot predict the future action of user since *purpose* of access is the future action; user with bad intention can change their mind after permission is granted to execute other tasks that are not part of a sequence of tasks representing the *purpose* that user claimed earlier. Thus, similar to the work in [42], we complement their work with our *purpose* enforcement technique.

6.5 Summary

In this chapter, we mainly discussed about *purpose* enforcement technique for privacy-aware access control policies. The enforcement of purpose is achieved by means of *purpose* achievement prediction, which is built based on association rule learning method with the support of user's access history and contextual information. The algorithms on how to calculate the predictive value were also presented in this chapter. The proposed purpose enforcement technique has significant improvement compared with the existing techniques [17][42] (see Section 6.1.4). One of the improvements is the ability to predict the *purpose* achievement of user based on the observation of user's past access history and contextual information. We argue that our proposed technique is better than the existing techniques. Firstly, the existing techniques are not able to predict the achievement of user's claimed *purpose*. Secondly, all the proposed techniques simply use role or simple workflow to enforce the *purpose* of access and they work under the assumption that all users in the system are the trusted entities; we assume otherwise. The use of those simple constraints (role or workflow control) to enforce the *purpose* is not sufficient as illustrated in Section 6.4.

In order to show the usefulness of our proposed *purpose* enforcement technique, we use a simple internal attack (internal intruder) scenario in healthcare information system and illustrate that while our technique can detect and prevent access to data in such scenario, the existing techniques [17][42] fail to do so.

Scenario: the data in Figure 6.5 and Figure 6.8 are used in this scenario. Suppose that David and John are in roles "Cardiologist" and "Physician", respectively. David and John have a bad intention to collect the health records concerning heart of patients and sell them to an insurance company. Such act is not authorised according to

hospital's policy. Suppose that we have three different access control policies for the three *purpose* enforcement techniques.

1. Byun et al [17] uses role as purpose enforcement constraint. The policy states that all users in role physician can admit patients to hospital while users in role cardiologist can access patients' health records concerning heart.

2. Jafari et al [1] uses workflow to enforce *purpose* of use. The policy states that every user in role physician can execute task " a to d (see Figure 6.5)" while role "cardiologist" can execute task "e", "f" and "h" for *purpose* of heart treatment.

3. In our technique, we use workflow and the result of the *purpose* achievement prediction as the access authorisation constraints. The policy states that any user in role "physician" can execute tasks "a" to "d" and user in role "cardiologist" can execute tasks "e", "f" and "h" if and only if the *purpose* prediction module indicates that user could achieve their claimed *purpose* with 0.90 of certainty (or PA=0.90).

Now David and John design a plot to illegally collect patients' health records concerning heart. The first technique by Byun et al [17] cannot prevent both of them from accessing data since David and John are in the roles authorized to access those data. Both of them can collect data as much as they want.

Second technique, Jafari et al [1], David can tell John to create many fake workflow instances, as John has rights to admit patients to hospital. John has rights to execute task "a" according to the policy. Then, David with the role as cardiologist executes tasks "e" or "f" to collect required patients' health records. The workflows created by John have never been completed; hence, *purposes* have never been achieved. Again, David and John can collect data as much as they want since there are no other constraints on access authorisation. For both techniques, auditing may be a choice to further enforce the policy, but not without drawback. Auditing may be able to detect policy violations after-the-fact, but it cannot prevent unauthorised access.

For our proposed technique, John can create many fake workflow instances and each time he creates it, it is marked as "On-going" if the remaining tasks have not been executed. The more John has workflow instances with "On-going" status, the more it affects *purpose* prediction's value since the prediction module takes into account all the three statuses: "Achieved", "Interrupted" and "On-going". The increase of either "Interrupted" or "On-going" workflows will impact on the confidence of the rule $\mathbf{conf}(G \Rightarrow \mathit{Achieved})$, which is the value of *purpose* prediction. When the confidence of the rule goes down to below 0.90, the system blocks the access for John. Consequently, limit the ability of John and David from accessing data.

With above example, we can see clearly that our technique is able to not only detect the unusual behavior of user, but also prevent unauthorised access and minimise damage to the system.

Chapter 7

Usage Control Architecture and Implementation

We concluded in Chapter 2 that we use a policy-based with the support of trusted client application as a mechanism to control and enforce the usage of private data in distributed environment. In Chapter 6, we introduced a privacy-aware policy enforcement technique. In this chapter, we propose a usage control architecture supporting purpose enforcement technique and the prototype implementation of such system in the healthcare information system environment. The validation of the purpose enforcement algorithms based on different data usage scenarios in healthcare system is also presented in this chapter. The rest of this chapter is organised as follows. Section 7.1 presents a general concept of usage control and enforcement. Section 7.2 is about usage control architecture. Section 7.3 presents in detail the prototype and implementation of the usage control architecture. Section 7.4 is about the validation and performance test of the proposed purpose enforcement mechanise. Finally, Section 7.5 dedicates to the summary of this chapter.

7.1 Usage Control and Enforcement

A data provider gives sensitive data to a data consumer with conditions, which latter become the requirements that restrict the future usage of data. When data provider releases data, he would like to have a mechanisms on the consumer's side to enforce his requirements. He would also like to check consistency of policies, and if mechanisms are capable of enforcing them. In general, usage control requirements are negotiated between data provider and consumer, and enforced using consumer-side mechanisms; upon successful negotiation, data is transferred from the provider to consumer and the usage control requirements are activated. From this point onward, mechanisms on the consumer's side will enforce the requirements (which is, in general, not fully possible

for all requirements, e.g. taking photographs of a monitor will always be an option). We assume the consumer possesses a secure data storage and that, prior to usage, data is routed through usage control mechanisms whenever it leaves the store. Figure 7.1 shows a general usage control state transaction.

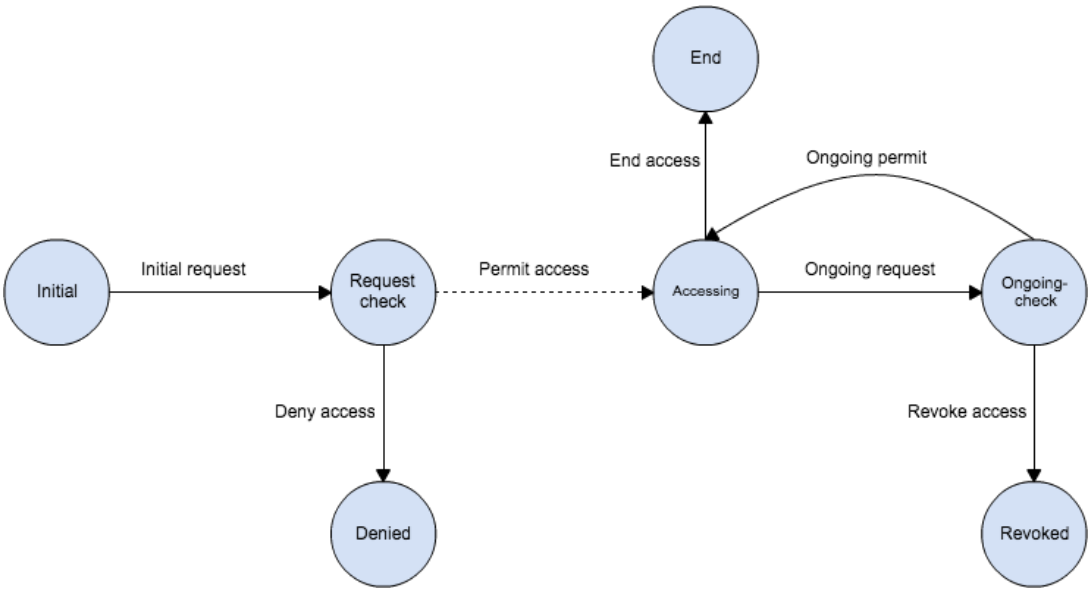

Figure 7.1: A general usage control state transaction.

7.2 Usage Control Architecture

We provide the functional and service architecture, for usage control supporting purpose enforcement, which is used as the control platform at client side control domain. As illustrated in Figure 7.2, system consists of three main parts: Workflow Management System, Policy Enforcement Point and Usage Control and Decision Point. Each part contains a set of components that are detailed in the following sections.

1. **User** is a Man Machine Interface acting as the intermediate layer between system and physical person.

2. **Resource** is the digital resource that is securely stored at the client side application domain.

3. **Policy Enforcement Point (PEP)** handles request from user and forwards it to usage control and decision and workflow management system. If the usage

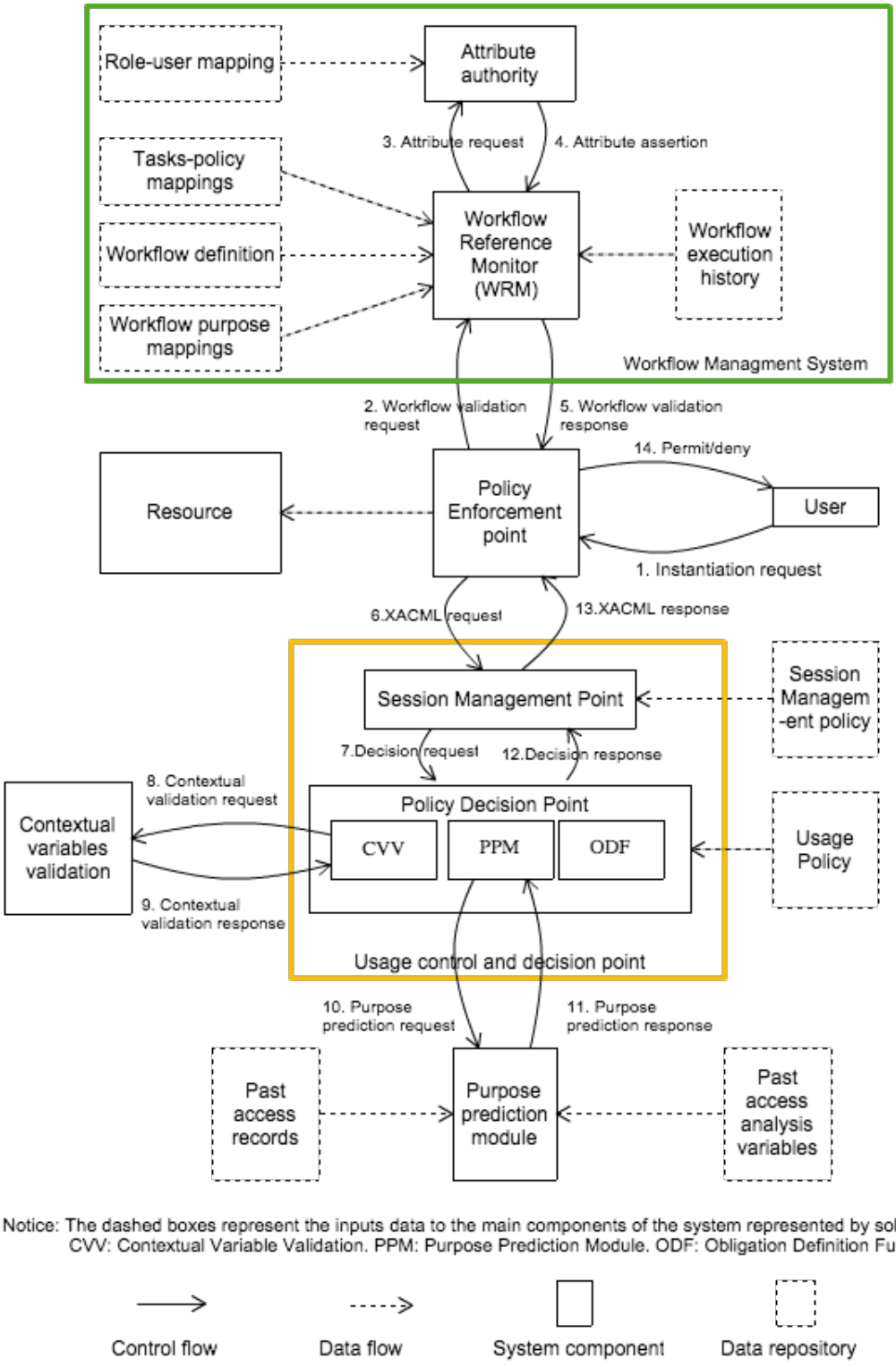

Figure 7.2: Usage control architecture supporting purpose enforcement for system using workflows.

request is granted by usage control and decision module, PEP allows user to access resource, otherwise, the access is blocked.

4. **Workflow Reference Monitor (WRM)** handles request from PEP for workflow validation process. It checks if the task and purpose requested by user are matched or not. It also checks the past executed tasks performed by the user for the requested workflow instance. WRM's functions include: checking user's role, subject's attributes or object's attributes. It also checks the validity of task and *purpose*. WRM contains the following modules.

    - **Workflow definition** defines sequence of tasks representing *purpose*.
    - **Workflow-purposes mappings** provides mapping information between workflows and *purposes*. It tells which workflow corresponds to which *purpose*.
    - **Tasks-policy mappings** provides information concerning the assignment of usage policies to each task.
    - **Attribute Authority (AA)** is responsible for providing information required to be used in workflow validation process.
    - **Role-user mapping** provides the information concerning the role of user.
    - **Workflow Execution History** provides information concerning the execution history of every workflow instance. The state (e.g. ongoing, interrupted or achieved) of each workflow instance is also provided.

5. **Usage Control and Decision Point (UCDP)** is responsible for controlling the usage session and also deciding the usage permission. This module consists of session management point and policy decision point.

    - Session Management Point (SMP) manages individual usage session. It also performs some other functions such as requesting decision to usage control decision module for each state of usage control session (see Figure 7.1). SMP also monitors continuously user, object and environment attributes, as well as any further actions requested by user. Based on decision received from usage decision point module, SMP either revokes or permits the ongoing usage session. In case of revoke, the ongoing usage session needs to be terminated immediately. The SMP validates each usage session based on the usage session management policy. The usage management policy can be expressed in a separate file or embedded in usage control policy.

- Usage Decision Point (UDP) is responsible for validating the usage control policy. Its role is to tell whether the access can be granted or not based on the analysis of all concern variables.

6. **Usage Decision Point** consists of three main components: Contextual Variable Validation, Purpose Prediction Module and Obligation Definition Function.

   - **Contextual Variables Validation (CVV)** is responsible for the validation of all contextual variables of tasks and *purposes*.
   - **Purpose Prediction Module (PPM)** is responsible for examining access history of user and estimating the level of certainty that user could achieve his claimed *purpose*. It takes the inputs from two different modules.
     - "Past access analysis variables " provides a list of variables used for analysing past access history.
     - "Past access records" provides the information concerning the access history of user.
   - **Obligation Definition Function (ODF)** is responsible for validation the obligation of user or system if any. For example, an obligation of notifying user during usage session.

7. **Usage policy** is the usage control policy expressing how user should use data.

Explanation of the data flow in Figure 7.2

Figure 7.2 depicts the architecture of usage control system supporting purpose enforcement mechanism presented in Chapter 6. We explain the data flow between different components of the system. We start with the creation of workflow instance. Once workflow instance is created, users who are supposed to be part of workflow execution process can request access to use any resource assigned to a particular task in the created workflow instance (call (1)). The access request (containing the right, task and purpose) is passed to policy enforcement point (PEP) where a preliminary verification and validation is performed. PEP sends a workflow validation request to workflow reference monitor (WRM) where the relationship between right, task and purpose are checked (call (2)). If the requested task do not belong to a workflow presenting the claimed purpose, the system provides the deny message to user. Otherwise, further verification is required such as checking the past executed tasks of the workflow (call (3) and (4)). After performing workflow validation request, WRM sends a response

to PEP (call (5)). If the response is positive, PEP proceeds to the next step by requesting to usage control and decision point. PEP sends a request directly to session management point (SMP) (call (6)). SMP further contacts policy decision point (call (7)). After validating the usage policy, policy decision point sends the decision response back to SMP (call (12)). SMP forwards the decision response to PEP (call (13)) and then PEP forwards it to user (call (14)). If the response is positive, client application allows user to use resource.

Policy Decision Point (PDP) validates a usage request based on usage policy. There are three important modules in PDP: CVV, PPM and ODF. CCV is responsible for validating the contextual variables required in usage policy (call (8) and (9)). PPM is responsible for calculating the purpose prediction value (call (10) and (11)) and ODF is responsible for validating the obligation requirement.

Remark: the modules that are responsible for purpose enforcement in Figure 7.2

In Figure 7.2, the enforcement of purpose of use is achieved at two levels. The first level is at workflow management system where the requested task and claimed purpose are verified. The idea is to ensure that the task being requested is authorised for the purpose being claimed by user. Moreover, the past executed tasks of the workflow need also to be checked to ensure that the sequence of the workflow is respected. The second enforcement level is at policy decision point where contextual variables and the likelihood of purpose achievement are checked.

7.3 Prototype and Implementation

In order to test our concept, we need to implement and validate it. A prototype of usage control system supporting purpose enforcement using our functional and service architecture in Figure 7.2 is built. The implementations include the workflow reference monitor and usage control and decision point. In usage control decision point, we focus on the implementation of purpose prediction module. The prediction module is built using our proposed enforcement technique presented in Chapter 6. We use Java as our implementation language and XML as the format for communication messages between different modules in the system. We have XACML version 2 [87] as the format for access control requests, responses and policies. We also make use of Java Enterprise XACML library [1] as the policy decision point engine.

[1] https://code.google.com/p/enterprise-java-xacml/

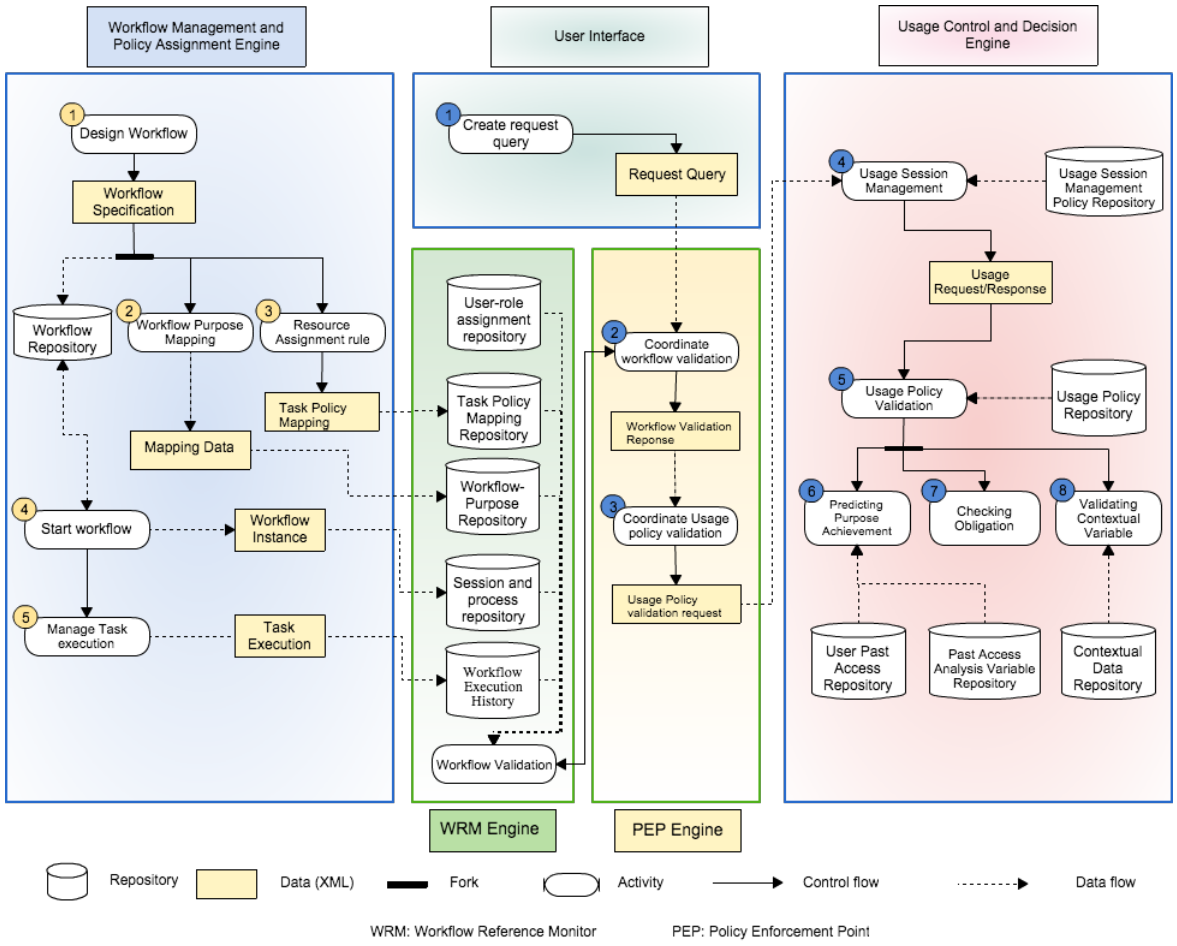

Figure 7.3: Overview of essential components and use case scenario.

7.3.1 Architecture

Figure 7.3 shows the essential components of our integrated toolset as well as a typical usage scenario. All the elements under workflow management, workflow reference monitor (WRM), policy enforcement point (PEP) and usage control and decision engine are Java-based application. The newly developed usage control and decision engine are the extension of the standard XACML [87], a Java-based access control and authorisation engine. For clarity, we discuss the workflow management and validation and usage control and decision. Then, we elaborate on purpose enforcement technique and explain how purpose prediction is achieved. We furthermore explain the communication protocol between different components of the system and also the storage structure. Actually, workflow management system and usage control system are two independent modules. They are connected by workflow reference monitor and policy enforcement point module. In order words, the system engineers are free to separate the management domain of the two modules.

Figure 7.3 shows three different processing phases: workflow creation and management, usage request and usage control and decision.

7.3.1.1 Workflow Creation and Management Phase

Figure 7.3 shows the essential components of workflow creation and management phase. The design of workflow is the first activity that needs to be done (call **(1)**). Workflow specification is defined using XML based on the defined task graph (see Section 5.2.1). Basically, the workflow specification contains the description of the necessary tasks that need to be executed and the sequencing order of those tasks. Moreover, the execution expiration time of each task and that of workflow are also defined in the workflow specification. Once the workflow is defined, it is stored in the workflow repository. The following activity is to assign the defined workflow to a purpose (call **(2)**). Then, the mapping information of purpose and workflow is stored in the workflow-purpose mapping repository for later use (e.g. during purpose validation phase). The next activity is the resource assignment rule (call **(3)**). In general, each task of the workflow is mapped to (or allocated to) a set of resources required for task execution. However, to ensure that the task performer use resources correctly, we need to have a resource assignment and management rule. The resource management rule defines the resource access policy for every task of the workflow. Once workflow specification is created, user can start a workflow instantiating from created workflow specification (call **(4)**). Each workflow instance is stored in session and process management repository. After workflow instance is created, system starts monitoring the execution of each task of that workflow instance (call **(5)**) according to the defined workflow specification. In general, the information concerning the task execution (e.g. on-going or interrupted task) is stored in workflow execution repository for later use.

7.3.1.2 Usage Request Phase

Figure 7.3, user interface for usage control is the Human Machine Interface (HMI) that allows physical person to interact with usage control system. In order to form a usage request, user needs to choose the following data: workflow instance, data user wants to access, right user wants to perform on data, task of the workflow and purpose. With that information, HMI produces a request query (call **(1)**). Then, the request query is sent by HMI to Policy Enforcement Point (PEP) engine where the preliminary validation is required. The first module in PEP that the request is validated is the workflow validation (call **(2)**). A module in PEP coordinates the workflow validation. Workflow validation looks at the past executed tasks of a requested workflow instance and other information such as workflow to purpose mapping and task to policy mapping (see Figure 7.3, WRM Engine). If the previous tasks have not been executed and they are required to be executed before requested task, the system rejects the request immediately and deny message is sent to HMI. Otherwise, the request is forwarded further to usage control and decision engine (call **(3)**).

7.3.1.3 Usage Control and Decision Phase

After receiving usage request forwarded by PEP engine, usage control and decision engine starts its task (call **(4)**). Usage session management (USM) starts a new usage session, but it is in inactive mode. Then, USM forwards the usage request to usage policy validation module (call **(5)**) where the usage request is validated against the usage policies in policy storage. All the values of the attributes in usage policy need to be validated: purpose achievement prediction (call **(6)**), obligation verification (call **(7)**) and contextual data validation (call **(8)**). Once the usage validation is completed, a response is sent to usage session management. If the request is granted, the usage session is activated. From that point onward, user can use the requested data. The usage session management periodically controls user's activity during the usage session. The usage right can be revoked if usage policy violation occurs. In case of negative response from usage decision point, a deny message is sent to usage session management point. With deny message, usage session management point removes the session that has been created. The deny message is forwarded further to PEP, and then to HMI.

7.3.1.4 User Management

The actual assignment of user to a role is done trough a separate module for user creation and management. Once the user to role assignment is done, assignment information, expressed in XML format, is stored in "user-role assignment repository" (see Figure 7.3). In our model, each task is assigned to a resource usage policy. Resource usage policy is assigned to role, not directly to user. Thus, at runtime, when user logs in, he can only access and start the tasks that have been assigned to roles that he

belongs. Every time user starts a task, a resource usage policy is applied and enforced. The user ID is part of the data that is communicated to user management system and workflow reference monitor engine for role validation. Assigning role to a task instead of user provides system engineer an easy way to manage user since system engineer needs only to maintain the user to role relation. For example, if a user is no longer in role to execute a particular task of the workflow, system engineer simply takes him out of the role he currently holds.

7.3.1.5 Repository Management and Implementation

Figure 7.3 shows the essential components of system, among which there are the repositories used to store the information for workflow validation, user management as well as usage control and decision. To simplify our implementation, we use XML and text file to store those information. The user-role assignment, workflow-purpose mapping, task-policy mapping are expressed in XML-based documents. Workflow definition is defined in XML-based workflow [1]. The access log data is stored in text file format. The past access analysis variable and contextual data are expressed in XML. For access-log data, at first, we used the access-log defined in Java Enterprise XACML. Java-XACML has its own access-log that records all user activities. However, since we need to have large enough log-data to test the performance of our purpose prediction module and it is difficult to manually making user's request with java-XACML, we decided to build our own access-log generator module that is able to create million lines of record. The access-log generator module generates the log-data having structure like in Figure 6.8.

7.3.2 Validation and Performance Test

Validation is an important component of algorithm development. Validation is the process by which developers confirm that a given algorithm meets acceptable levels of accuracy and performance. Achieving effective validation requires a dataset with known input and output parameters, whereby algorithm outputs can be directly compared against the already established output values. To validate and assess our purpose achievement prediction algorithm (see Chapter 6, Section 6.4), we use datasets and challenges as the validation and assessment method. Different sizes of access-log are created and tested against different set of usage control policies.

Since our main focus is the usage control enforcement, to be precise the purpose prediction model, we focus our implementation on this module in this Chapter. The input data in Figure 7.4 are used for validation and testing.

[1] http://en.wikipedia.org/wiki/XPDL

7.3.2.1 Access-log Generator

The structure of the access-log, that is used as input data for association rule learning method, consists of the following elements: workflow-instance-id, user, user-role, task, right, data, *purpose* and status (see Chapter 6, Section 6.4 for the definition of each element). However, the real raw access-log data do not contain only those elements; hence, we need to create a data mining module that mines only the required data elements from the raw access-log. The objective of the mining is to simplify the input data for association rule learning method. There are two steps for access-log generation. Firstly, we generate the raw access-log containing similar log structure to that of JAVA-XACML. Second step is to mine the raw access-log. The final product is the access-log containing the data elements as presented in Figure 6.7. To create an access-log, we need to have workflow definition, workflow to *purpose* mappings, access policies applied to each task, user, user's role and request query. The user's request query, that contains user, task, right, data and *purpose*, is constructed randomly. Each element of the user's request query is chosen randomly from our predefined sets of those elements.

Experiement number	Workflow complexity	Number tasks	Size of access log	Size of access log in MB	Workflow status (AC, ON, IN)	Worfklow instance creation rate
1	Level 1	30	18350 workflow instances	4	95%, 3%, 2%	50/day
2	Level 1	30	36500 workflow instances	7.8	90%, 7%, 3%	100/day
3	Level 2	80	73000 workflow instances	19.4	95%, 4%, 1%	200/day
4	Level 2	80	146000 workflow instances	38.3	90%, 4%, 6%	400/day
5	Level 3	160	182500 workflow instances	56.1	92%, 6%, 2%	500/day
6	Level 3	160	365000 workflow instances	109.5	85%, 10%, 5%	1000/day
7	Level 4	160	730000 workflow instances	231	97%, 2%, 1%	2000/day
8	Level 4	160	1215000 workflow instances	387.9	98%, 1%, 1%	3000/day

* We simulate the size of access log based on one year system activities.
AC= Achieved, ON= On-going, IN= Interrupted

Figure 7.4: Experiments' inputs.

7.3.2.2 Testing Input Data

In order to test and validate the performance of our implemented prototype, we have done 8 experiments with different workflow complexity and size of access log. We used the input data as presented in Figure 7.4.

Workflow Complexity. We defined 4 different levels of workflow complexity. The

complexity of the workflow is represented by the number of workflows, number of tasks for each workflow and relationship between tasks in the workflow.

- Level 1, there are at most 20 workflow definitions and each workflow has at most 5 tasks.
- Level 2, there are at most 50 workflow definitions and each workflow has at most 10 tasks.
- Level 3, there are at most 100 workflow definitions and each workflow has at most 15 tasks.
- Level 4, there are at most 100 workflow definitions and each workflow has at most 20 tasks.

Size of access log. There are two ways to represent the size of access log: (1) by the number of workflow instances or (2) the size of access log in MB (Mega Byte). It is worth noting that the size of access, in Figure 7.4, is calculated based on one year system activities.

Workflow instance creation rate is the number of workflows created per day.

7.3.2.3 Requirements and Scenarios

We defined two access scenarios in healthcare information system for validation. Our main objective is to find out how good our *purpose* prediction achievement module is. Could *purpose* prediction module predict the future achievement of *purpose* and capture the mal-intended user as expected? Since access-log involves in decision process, does the size of access-log affect the general performance of the system?

First scenario: David (physician) requests to execute task "T1" for *purpose* of heart-treatment (P0). David wants to read data (D1). David is in hospital at the time of request. We use in this scenario the access-log as presented in Figure 7.4 (experiment 1-8). There are two separate tests in this scenario. Firstly, we test with the given data set in Figure 7.4 . Secondly, we manually change some access-log parameters for David, such as increasing the number of Interrupted 1% and On-going 1% to simulate the unusual activities of David and keep the same required PA's value (0.90) in policy.

Second scenario: in general, when the access-log becomes larger and larger, the time required to analyse the log becomes important. To find out the impact of the size of access-log on the general performance of system, we use the access-log in Figure 7.4. The same user's request information used in scenario 1 is also used in scenario 2.

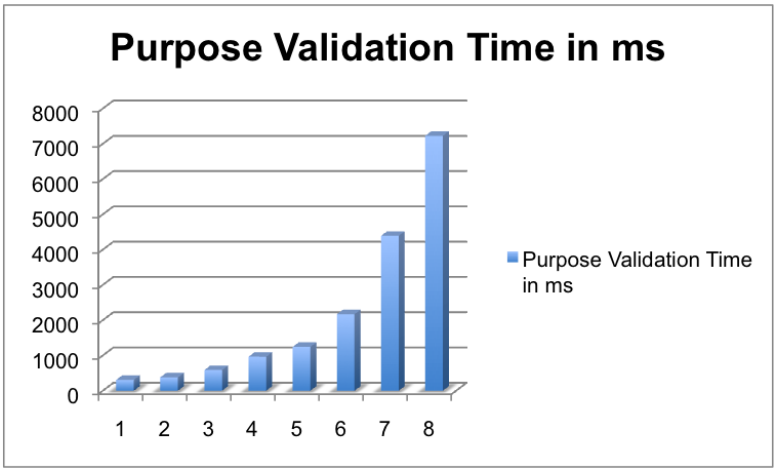

Figure 7.5: Experiment results. (Y) axis represents the purpose validation time in milliseconds while (X) axis represents the experiment number (see Figure 7.4).

7.3.2.4 Performance Analysis

We developed and tested our java packages in Eclipse Standard/SDK (version Kepler) installed on Macbook air OS version 10.8.4, processor 1.3 Ghz Intel Core i5 with memory 8GB DDR3.

First scenario: the system validated the rule "**conf**$(\{David, Physician, T1, read, D2, P0\} \Rightarrow Achieved)$" using the experiment data in Figure 7.4. For each experiment data set, we repeated the test with different set of access-log data, but keep relatively the same access-log's size. We found that if the access history of David maintains 90% of achieved and other 10% for Interrupted and On-going, the system always provides positive response. For the second test, we manually modified the access-log by increasing the number of Interrupted and On-going by 1%. We modified the access-log record-by-record for "Achieved" and "On-going" till they reached 1% change. We observed the result for each change. We found that with the large access records, the slightly change of the number of Interrupted or On-going does not affect the decision, only after it reaches a certain threshold, the affect starts to take place. This implies that policy maker must be careful when defining the value of PA; the value of PA must represent the current system's activities.

Second scenario: the aim of the second scenario is to find out how the rule validation time affects the general performance of the system when access-log grows larger and larger. We did 8 experiments with different levels of workflow complexity. We started with a small access log with the size of a few MBs to hundred MBs (see Figure 7.4). The results of the 8 experiments are presented in Figure 7.5. We see that the time required to validate a request increases in relation with the size of the access-

log. This is as expected. In case of loaded system, this issue can be a big challenge. However, there are two possible ways for reducing the request validation time: the first option is to minimise the size of the access-log; another is to increase the computation power of the system (e.g. parallel computing).

To minimise the size of access-log, we need to minimise the size of observation interval. One solution is to divide a large observation interval into many smaller intervals (equal size). Then, we define the PA's value of each interval. The final PA's threshold value, which is used in access policy, is an average of the values (PA) from the smaller intervals. With this method, the size of the access-log used to valid the rule is the size of the access-log for one interval (the most recent access-log), not the entire log. For example, instead of using one-year access-log data, we can use a month access-log to validate the rule. However the PA's threshold value is defined based on the observation of one-year interval.

7.4 Summary

This chapter brought together the contributions of Chapter 4-5-6 into a usage control system supporting purpose enforcement. Our ambition is to design the usage control architecture supporting purpose enforcement and to demonstrate that our proposed purpose enforcement technique works. For implementation and validation, we developed different modules of the proposed usage control architecture (see Figure 7.2 and 7.3). These include: the usage control interface, usage control and decision point and workflow reference monitor. In addition to that, since we use datasets and challenges as the validation and assessment method for proving our purpose enforcement algorithms, we need large datasets for access-log. To achieve this goal we built an access-log generator module that is able to simulate access log for different workflow complexity. Finally, we did 8 experiments with different level of workflow complexity and different access log size. We concluded our finding in performance analysis section (see Section 7.4.4).

Chapter 8

Protecting Personal Data in Privacy-Preserving Perimeter Protection System

In Chapter 1(Section 1.3), we listed two application domains for the implementation of our finding. The first application domain is distributed healthcare where we presented the prototype of such system in Chapter 7. The second application domain is the privacy preserving perimeter protection system. Since 2013, we have been participating in a European research project, the Privacy Preserving Perimeter Protection Project (P5). The goal of the P5 is an intelligent perimeter proactive surveillance system that works robustly under a wide range of weather and lighting conditions and that has strong privacy preserving features. P5's research has similar in goal to that of our thesis. The main of objective of P5 is to protect personal data of individual generated by different surveillance tools (e.g. CCTV). We have contributed to the project ranging from the design of a global system architecture to the design of access control model and its implementation. Given its similarity to our doctoral research, some part of our work have been used as a platform for the design and implementation of access control system for P5. This chapter dedicates to the work done in P5. The rest of the chapter is organised as follows. Section 8.1 is the introduction. Section 8.2 is about the motivation and the description of P5 project. Section 8.3 introduces the privacy-aware access control and Trusted Third Party (TTP) module. Section 8.4 is about privacy-aware access control model. Section 8.5 presents the access control scenarios and policies definition for P5 system. Section 8.6 talks about access control architecture and implementation. Section 8.7 is related work and contributions while Section 8.8 is summary.

8.1 Introduction

Critical buildings and infrastructures (e.g. nuclear power plants, military operation zones, governmental or private institutions) require strong and unlikely breakable physical security protection from physical or forceful attacks. To protect such infrastructures beyond the use of conventional methods such as fence, they normally use different surveillance tools, such as visual cameras, thermal cameras or radars, to observe and detect activities around the protected infrastructures. In most of the cases, the surveillance covers only the private zones belonging to the institution, but sometimes it goes beyond by covering a larger area, for instance public area, in order to have an early warning and enough time to react in case of attack. However, including the public area into the surveillance perimeter poses challenges for personal data protection since surveilling the public areas, especially people moving around the areas are not permitted in some countries like in EU or USA. EU Directive 95/46/EC [28] does not allow any government or private organisation to do surveillance in public area without the approval from concern government authority.
There are issues related to privacy when covering the public areas, such as roads or residential areas [28] [54]. For residential area, it can pose threat to people living in that area because one can observe the daily life of a given person or a group of people by analysing the data generated from surveillance tools. Thus, when designing perimeter protection system, one needs to take into account the data protection aspect.
In general, there are two data protection phases.

1. Firstly, protecting real-time data streamed from sensors around facility, one needs to ensure that data streamed from sensors are well protected and they have not been altered before they arrive at the control room. Furthermore, one needs to filter out all privacy-related information before showing them to guards[1] in the control room. For example blurring the face of people.

2. Secondly, protecting data in the storage, in some cases, data generated from sensors need to be stored for a while for forensic purposes. For example, if there is the criminal scene in the coverage area, the authorities may require getting access to those data for investigation.

Given all above illustration, we can see clearly the need to protect personal data in such system. We introduce, in this chapter, an access control system designed particularly for managing and controlling access to private data in perimeter protection system. The proposed access control system is to ensure that personal data are properly protected and only authorised people can use those data for purpose they intend for. As it is

[1] Guards refer to those who are working in control room or surveillance room.

required by laws [28] that only the authorised entity is allowed to process personal data, we also introduce the concept of Trusted Third Party (TTP)[1].

8.2 Motivation and P5 Project

In this section we present the motivation and the introduction of P5 project, which is the driving force behind our work presented in this chapter.

8.2.1 Motivation

Using conventional methods such as fences with the support of guards to protect the critical buildings and infrastructures is no longer enough given the sophisticated tools and technologies that we have nowadays. Skilful attackers can exploit the use of such advanced tools and technologies for their advantages to break the security barrier. To provide a reliable security protection to such area, one needs to use a more advanced surveillance system, with the support of the tools such as visual or thermal camera. The existing perimeter protection systems [61][29][30] assume that surveillance should take place only inside the private protected area; hence, the privacy concern can be ignored since the surveillance is limited within the private area. This is because within the perimeter of facility, facility manager has the right to do surveillance on their employees and the guests visiting the facility. They normally use information boards or signs informing the guests or employees about the surveillance. However, ignoring privacy issue is no longer possible when the surveillance perimeter includes the area beyond the private area (e.g. covering the public area). The privacy issue must be taken into account in such case and facility manager is bound by laws [28] to ensure that data are properly protected. There are good reasons to extend the surveilling area. One of which is that it gives facility guards enough time to react if the attack is to occur. With conventional system where the surveillance covers the area not farther than the fences or the boundary of the protected area, it may be too late for security guards to react if there is an attack with the sophisticated tools (e.g. missiles or high speed vehicles). Thus, securing the facility and at the same time protecting personal data of people affected by the surveillance are the main motivations.

8.2.2 P5 Project

P5, the Privacy Preserving Perimeter Protection Project, is the European and FP7 FUNDED (http://www.foi.se/p5) project for the protection of critical infrastructures to benefit the sustainability of society and future well-being of the European Citizens.

[1]TTP is an entity outside the protected facility and it is responsible for authorising the access to personal data.

The goal of the P5 project is an intelligent perimeter proactive surveillance system that works robustly under a wide range of weather and lighting conditions and that has strong privacy preserving features. The system will monitor the region outside the security area of critical buildings and infrastructures, and give early warning if terrestrial or airborne threats are detected. The system will support, rather than replace, a human operator. A low false alarm rate due to animals or other innocuous events, combined with high threat detection sensitivity and privacy standards, are the central ambition of this project. To achieve these goals, a multispectral sensor suite comprising both passive and active sensor is envisaged (e.g. a system based on radar, visual and thermal sensors). The sensor suite will be complemented with advanced algorithms for information fusion, object detection and classification, privacy preservation and high level modeling of intent and behavior analysis.

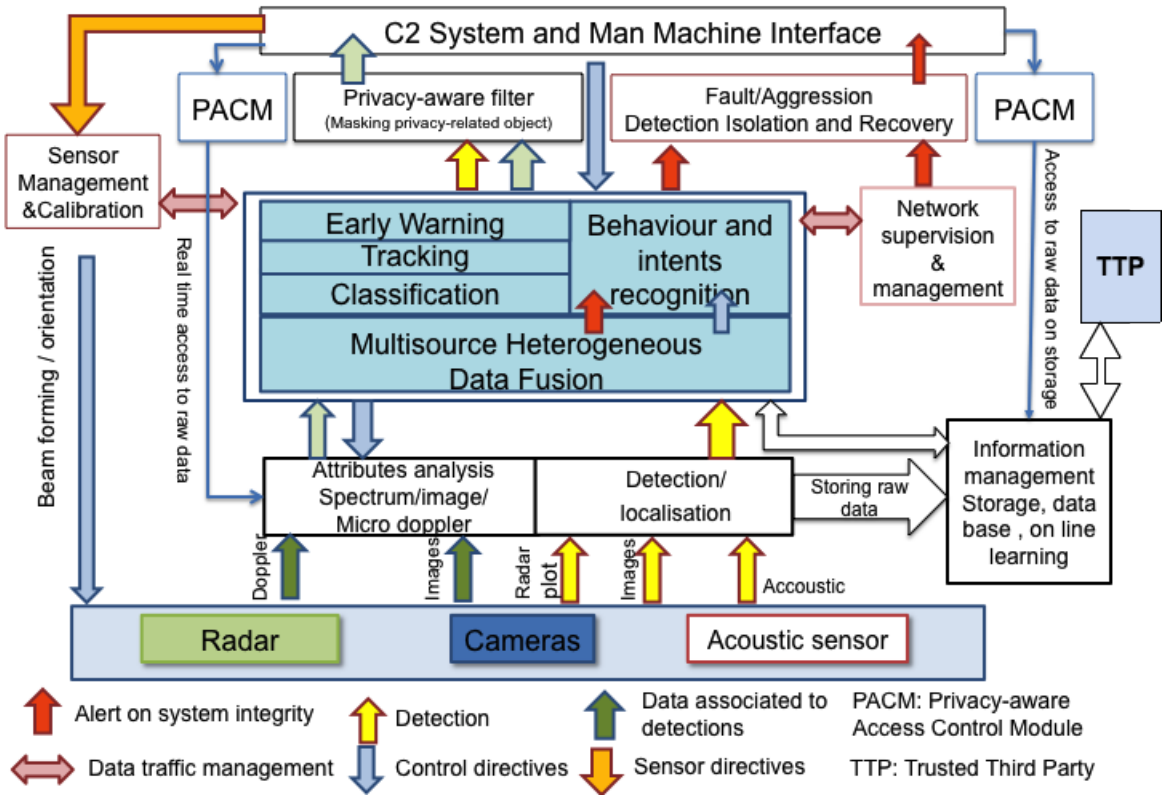

Figure 8.1: Privacy preserving perimeter protection system architecture

8.2.2.1 P5 System Architecture

In this section we present the global architecture of P5. We provide a brief description of each component and the details of three components that we address in this chapter: privacy-aware access control module, privacy-aware filter and TTP.
P5's team designed global system architecture for P5 (see Figure 8.1) where different layers of system components are introduced. The lowest layer hosts the sensors that provide different data types to the upper layer modules. Sensors can be managed and calibrated remotely through "man machine interface" module. The data from sensors are provided to the "attributes analysis" module where data are processed. Data from different sensors are detected by a module called "detection/localisation" where data are tagged with their location identification and privacy concern level.

After "attributes analysis and detection/localisation" does its job, data are passed to a module called "multi-source heterogeneous data fusion" where the data from three different sensors (e.g. radar, thermal and visual camera), at the same location, are fused together. The main idea of fusing data from different types of sensor is to increase the detection precision for different surveillance conditions. For example, visual cameras may have bad visibility during the bad weather (e.g. raining or snowing) and at night. While visual cameras have such drawbacks, thermal cameras do well in such conditions. In some situations, using visual or thermal camera could not help us to detect and separate the objects that are in front of camera and are positioned in series closed to each other. Both visual and thermal cameras may detect as single object. With this reason, radar is used as complement for object detection since radar can provide precise position of every single object.

After performing data fusion at "multi-source heterogeneous data fusion", the fused data are passed to "object classification, tracking and behaviour and intents recognition" and then to "early warning module" (see Figure 8.1). The "object classification" module is responsible for identifying the objects (e.g. people, animals or vehicles). After classifying the objects, the tracking module plays a role, it follows up any suspicious object. Both, "object classification and tracking" modules pass data to "behaviour and intents recognition module" where the movement or activities of the object is analysed. If the movement pattern is considered as threat (e.g. object moving fast close to fence), a warning message is generated.
The "fault/aggression detection isolation and recovery" module provides the procedure on how to tackle the attack and how to recover. "Network supervision and management" module is responsible for monitoring the entire system (e.g. network and connectivity). "Information management, storage, database and online learning" module is responsible for storing and retrieving data for later use.

There are three more modules, in the proposed architecture, which ensure privacy preservation and personal data protection: "Privacy-aware Filter (PF)", "Privacy-aware Access Control Module (PACM)" and "TTP". The details of the three modules

are presented in next section.

8.2.2.2 PF, PACM and TTP

In P5, privacy issues happen when people in the control room at protected facility want to view raw data[1] bearing privacy-related information directly from sensors in real-time or access past raw data from storage. Any access to raw data needs a special control to ensure that data are not used excessively. With this reason, we propose to insert two modules for controlling access to raw data and for filtering privacy-related information (see Figure 8.1).

1. Privacy-aware access control module (PACM) is responsible for controlling access to raw data. This module is responsible also for enforcing access control policies. The access control policies are generally defined by the Trusted Third Party (TTP). The idea of using TTP to define privacy-aware access control policies instead of allowing people in facility to do the work is to avoid the uncontrolled data manipulation by those people. The details of TTP will be discussed in the following section.

2. Privacy-aware filter is responsible for filtering the privacy-related information (e.g. a car's license plate or person's face). In general, the data filtering module performs based on object-masking policy, which defines masking procedure and criteria for different type of objects. The details of privacy-aware filter are presented in Figure 8.2. Privacy-aware filter takes two inputs. The first input is from the object classification module and the second input is from object masking policy. Object classification identifies different kinds of objects while object-masking policy tells which object needed to be masked.

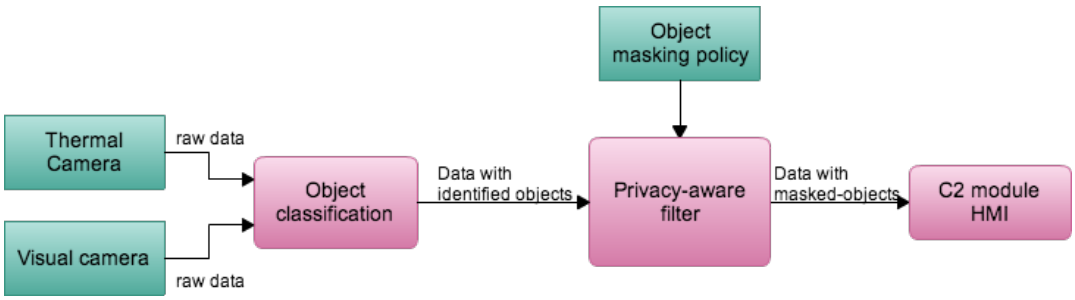

Figure 8.2: Architecture of privacy-aware filter

[1]Raw data are data generated from sensor without alteration.

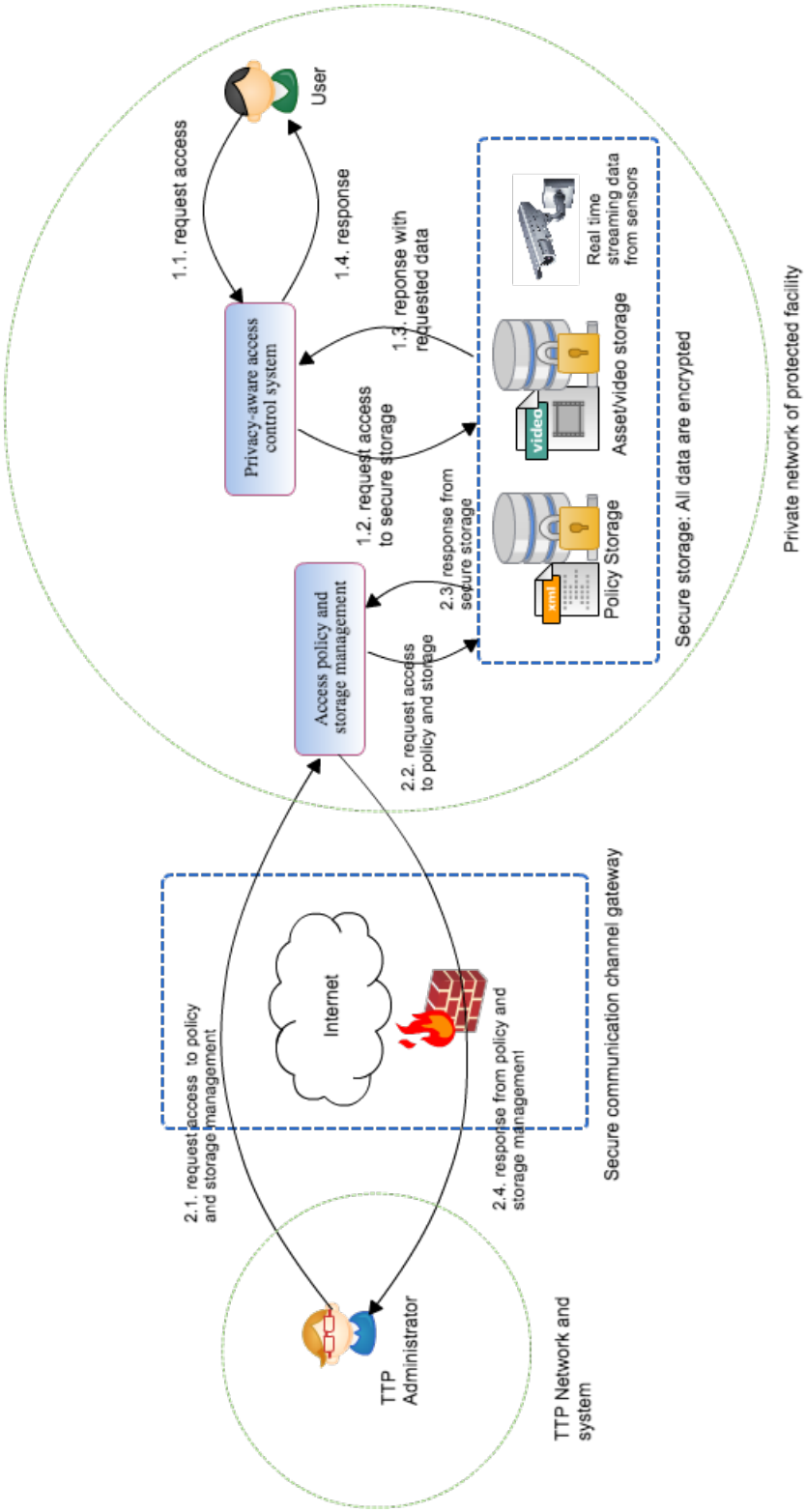

Figure 8.3: Global architecture for privacy-ware access control module and TTP, data flow

8.3 PACM and TTP

We start with global architecture containing privacy-aware access control and TTP modules. In the detailed functional architecture presented in Figure 8.3, there are two main components: TTP and privacy-aware access control. TTP is managed by TTP administrator. Accesses to raw data are controlled by the privacy-aware access control module installed in the private network of protected facility.

- TTP module provides to TTP administrator a way to manage access control policies, to manage and to protect the raw data and to audit the access to raw data.
- Privacy-aware access control module controls access to raw data in the protected facility. The access permission is determined based on access control policies defined by TTP administrator.

It is worth noting that the whole system is generally installed in the protected facility. However, TTP administrator can remotely control system through secure communication channel.

Details of Figure 8.3:

1. *TTP administrator* can be a trusted private or government entity, which is authorized for the job.
2. *Secure communication channel* is a secure communication medium between TTP system and the system installed at the facility.
3. *Access policy and storage management* provides some functional features for TTP administrator to manage access control policies and the storage of raw data.
4. *Secure storage* is responsible for protecting data in storage (e.g. access control policies and data from sensors).
5. *Privacy-aware access control module* is responsible for controlling access to raw data in storage or real-time data from sensors.
6. *User* is a physical person allowed to access raw data for a particular purpose.

Figure 8.3: data flow explanation

We explained previously the details of each component of the system (see Figure 8.3). In this section we provide a step-by-step explanation of the data flow. There are two different access scenarios in Figure 8.3.

- Firstly, TTP administrator requests access to policy and storage management module (call 2.1). After successfully authenticated TTP administrator, "policy and storage management" module replies. From that point on, TPP administrator can manage access policies or data in storage. Through "access policy and storage management" module, TTP administrator can request access to policy and data storage (call 2.2) and gets the reply (call 2.3). Finally, access policy and storage management module forwards the data to TTP administrator (call 2.4).
- Secondly, user in facility requests access to raw (call 1.1). Privacy-aware access control system validates the request. If the permission is granted, it contacts the database management system storing the requested data (call 1.2). Then, the database management system replies with requested data (call 1.3). After getting data from database management system, privacy-aware access control system forwards the data to user (call 1.4).

It is important to note that since our main addressing issue in this paper is the design of privacy-aware access control system, we do not go into the details of TTP's architecture.

8.4 Privacy-aware Access Control

In this section, we present in detail the access control system taking into account the privacy aspect. It includes the access control requirements for privacy preserving perimeter protection system, access scenarios and access control model designed specifically for such system.

8.4.1 Access Control Requirements

To identify the access control requirements for privacy preserving perimeter protection system, we conducted two different studies. Firstly, we worked with legal group to study the EU Directive 95/46/EC concerning the protection of personal data of individual. Secondly, we did a formal survey and also conducted a broad range of data collection and analysis. For the field works, we visited existing perimeter protection systems installed in the protected facilities in United Kingdom (UK) and Sweden, such as National Air Traffic control in UK and OKG Nuclear Power Plant in Sweden. A list of questions, concerning the management of access control and data storage, were proposed to people in the control room as well as technical people working in security department of those facilities. The responses were analyzed and then refined. Based on the result of our survey and legal studies, we can classify the access control and data protection requirements into four main points.

1. **Legal requirements:** Directive 95/46/EC specifies rules for handling personal data. The directive defines objectives for the legislation of the member states of the European Union and it is binding on the member states as to the result to be achieved but leaves them the choice of the form and method they adopt to realise the community objectives within the framework of their internal legal order. The directive uses the terms "controller", "data subject" and "processing". Article 2 defines the terms in the context of the directive.

   "Controller" shall mean the natural or legal person, public authority, agency of any other body which alone or jointly with others determines the purposes and means of the processing of personal data, ...

   "Data subject or data owner" is an identified or identifiable natural person, ...

   "Processing" shall mean any operation or set of operations whether or not by automatic means, such as collection, storage, adaption or alteration, use, disclosure by transmission, ...

   Article 10 specifies the obligation of data controller when processing personal data. Article 10 lists the following information that data controller must provide to data subject.

   - (a) the identity of the controller and of his representative, if any;
   - (b) the purpose of the processing for which the data are intended;
   - (c) any further information, such as the recipients or categories of recipients of the data, ...

   Article 11 compels the data controller to notify the data subject, even if the data are not obtained from them. Article 11 describes the need for an information service attending to the information rights of private persons. Such an information service can support the process of notifying the data subject by providing the needed information.

   In summary, from those articles, there are three main requirements:

   - Data controller needs to notify data subject every time of access.
   - Processing of private data is limited to the purpose for which data are intended; excessive use is not allowed.
   - Consent from data subject is required when processing personal data.

2. **User management requirements:** security department personnel manages access to control room as well as sensors. An assigned group of users, while they are on duty, are allowed to be in control room to view and analyse real time data streamed from sensors. By default, data are filtered out all the privacy-related information. In case of emergency where there are intruders attacking the facility, users in control room are allowed to access raw data. Other assigned group of users can access raw data in storage or transfer recorded raw data to third party, but special access permission is needed. The main purpose of storing data from sensors is for forensic purpose.

3. **System performance requirements:** since we deal also with real-time data, access control to such data stream must be reasonably fast to avoid the delay to data stream. In order words, the time required for validating access control policy must be small.

4. **Security and data protection requirements:** processing of private data must be secured. We need to make sure that only authorized people can get access to data. Data controller should be a trusted entity that overlooks the management of access control policies as well as data in storage. To avoid data manipulation by people inside the protected facility, we require data controller to be a trusted entity outside the facility, either authorised government or private entity.

With those access control requirements, we are able to define the access control model presented in Section 8.4.2.

8.4.2 Access Control Model

We introduce Context- and Privacy-aware RBAC [82] (CP-RBAC), an access control model designed for controlling access to private data in privacy-preserving perimeter protection system. In CP-RBAC, access authorisation is based not only on the user's role, but also on contextual information, such as temporal-, spatial- and environment-context. Furthermore, the concept of privacy is also introduced into the model.

8.4.2.1 Role Model

We propose to extend Role-Based Access Control (RBAC) (see Figure 8.4). Role-based access control model (RBAC) has been traditionally used for designing access control systems for organisation. In such systems, users are assigned to a role by system administrator, and such memberships also tend to have long duration. In contrast to this, in our model roles are defined as part of an application's design. Such roles come into existence only when that application is deployed and executed and they last during the application lifetime. The access control system admits a user to a role based on

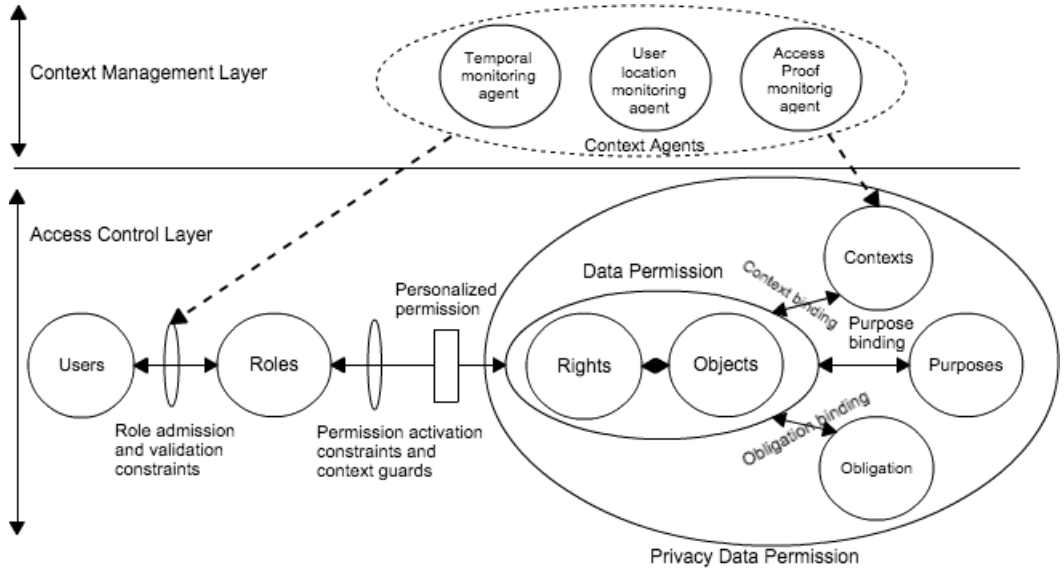

Figure 8.4: Context- and Privacy-aware Role-Based Access Control Model (CP-RBAC)

the admission constraints. For example, the admission constraint based on the working hour, user is admitted to a role only when he/she is on duty.

In the proposed model, contextual information is used as constraint for both the role admission and data permission assignment. The purpose of access and obligation are also the binding constraints on data permission to preserve and protect the privacy of data subject.

8.4.2.2 Access Control model

CP-RBAC (see Figure 8.4) consists of the following entities.

- U is a set of users (u) where $u \in U$.
- R is a set of roles (r) where $r \in R$.
- G is a set of rights (g) where $g \in G$. For example, right to "read", "copy", "delete".
- D is a set of data where $d \in D$.
- P is a set of purposes (p) where $p \in P$.
- O is a set of obligations (o) where $o \in O$.

- C is a set of contextual variables (c) where $c \in C$.

Then, we formulate the privacy-sensitive policy (RP): $RP \subseteq R \times ((G \times D) \times (P \times C \times O))$. The detailed formulation of privacy-sensitive policy is as follows.

- The set of Data Permission $DP = \{(g, d) \mid g \in G, d \in D\}$
- The set of Privacy-sensitive Data Permission $PDP = \{(dp, p, c, o) \mid dp \in DP, p \in P, c \in C, o \in O\}$
- Privacy-sensitive Data Permission to role Assignment PDPA $\subseteq R \times PDP$, a many-to-many mapping privacy-sensitive data permission to role.

8.4.2.3 Context and Obligation Expression

In this section, we provide a simple, but sufficient way for expressing contextual variables and obligation in our model. It is worth noting that the context and obligation expression we present in this section are sufficient for use in P5 based on the requirements we presented in Section 8.4.1. It is not necessary a general context and obligation expression.

Contextual data are the information surrounding user, data[1] and reason that user needs to execute the action. Contextual information can be anything, such as user's personal data, location or time. For example, the physical location of user can be considered as contextual information.

Definition 1: contextual variables expression

Let C be a set of contextual variables (c), where $c \in C$. "c" has the finite domain of possible values, denoted as DC where $dc \in DC$. "c" is equipped with the relational operators (Oprs) " $=, \neq, \geq, and \leq$ ". The condition of c has the form (c opr dc).

let c_1 and c_2 are two contextual variables in the form of the atomic condition. Then, $(c_1 \wedge c_2)$ or $(c_1 \vee c_2)$ is also condition. For example, using working-hour of user as contextual variable, if user's working-hour is between 8am and 5pm, we can express: working-hour $\geq$ 8am $\wedge$ working-hour $\leq$ 5pm.

Obligation is defined as the action that user or system needs to fulfil before or after accessing data. For example, paying before listening music is a form of obligation or notifying data provider every access is also considered as a form of obligation.

---

[1] "Data" refers to the data that user wants to access.

Definition 2: obligation expression

Let O be a set of obligation variables (o), where $o \in O$. "o" has the finite domain of possible values, denoted as B where $b \in B$. "o" is equipped with the relational operators (Oprs) " $=, \neq, \geq, and \leq$ ". The condition of "o" has the form (o opr b). For example, a payment obligation has the form: $payment \geq 50\$$.

8.4.2.4 Context-aware Role Admission and Personalised Role Permission

We use context-aware role admission to admit users to a role. A complementary aspect of context-based role admission is the need to revoke a user's role memberships when specified context conditions fail to hold. For example, in the perimeter protection system, user in the role "guardian" membership must be revoked when they are out of control room or off duty. The location of user or working hours is called "role validation constraint".

The services, that are accessible through role permission, may be different for different role members and may depend on the contextual information associated with a role member. Personalised role permission allows system to revoke role membership on different object instances based on each role member's individual context. An example in privacy preserving perimeter protection system: suppose that there is a crime happened in or around protected facility. The authority wants to view the past videotapes from that area. The facility manager assigns a user "Edward", in role "security manager" to assist and facilitate the authority. This means that only Edward can access the videotapes, not all the members in "security manager" role. In other words, the videotapes are personally assigned to Edward. If the videotapes are assigned to role "security manager" under traditional RBAC, all the members in that role can access to the tapes. This is not what we want.

8.5 Access Scenarios and Policy Definition for P5

In this section, we provide the access scenarios for P5. Then, with those scenarios we define the access control policies. Finally, we express those policies with the access control model we presented in Section 8.4.2.

8.5.1 Access Raw Data in Real Time

Scenario: this happens when system detects a threat in protected perimeter. In such situation, guard in the control room may trigger the emergency button to get access to raw data in order to get a clear view of the target objects. However, in order to prevent

guard from unnecessarily triggering the emergency button, guard is allowed to trigger emergency if and only if there is a positive acknowledgement from "early warning module" (see Figure 8.1). That module is responsible for providing a warning message when it detects the abnormal behaviour of the objects. The positive acknowledgement means it really detects the abnormal activities of the target objects. Without such positive acknowledgement, guard cannot trigger emergency situation to access real-time raw data.

Policy definition (P1): we define role "guardian". The users in role "guardian" are able to view real-time raw data streamed from sensors. However, they can do so only in case of emergency. Otherwise, they can only view filtered data where privacy-related information is filtered out. In addition to that, users can trigger emergency if and only if there is a positive acknowledgement from early warning module. Moreover, to be able to keep track user's activities, users are required to notify system every access to raw data.

With above policy description, we are able to mine the following information.

- Role of user: "Guardian".
- Action: "View".
- Data: "streaming video".
- Context: "acknowledgement-from-early-warning".
- Purpose: "Observing-suspicious-object".
- Obligation: "notify".

With the above information and policy expression in Section 8.4.2, we can formulate the following access control policy.

PDPA to role "Guardian": P1

P1= (Guardian, (View, streaming video), Observing-suspicious-object, (acknowledgement-from-early-warning= positive), Notify=yes))

8.5.2 Review or Replay Recent Past Raw Data

Scenario: this happens when guard in the control room wants to review or replay recent past video stream. We define "recent past video stream" as the video stream that has been recorded within the last 30 minutes (It can be different number depending on requirements). The replay occurs when system detects abnormal activities of objects

and guard wants to observe the recent past activities of those objects.

Policy definition (P2): users in role "guardian" are allowed to review recent past video stream. The same rule in P1 is applied in P2. However, one more context is required that is the life of video stream, which is set to be 30 minutes. The life of video stream context limits the access to the past video streams, which are older than 30 minutes. Any access to older past video streams needs to be controlled by policy P3. With the above policy description, we are able to define the policy (P2) as follows.

PDPA to role "Guardian": P2

P2= (Guardian, (View, streaming video), Observing-suspicious-object, (acknowledgement-from-early-warning= positive $\wedge$ life-of-video $\leq$ 30 minutes), Notify=yes))

8.5.3 Access Raw Data in Storage

Scenario: Government authority may need to access past raw data for an investigation purpose (e.g. if there is a crime scene in the coverage area around the protected facility, authority can request access to raw data generated from the cameras installed in that area). Authority can request raw data from facility manager. However, in order to get access to raw data facility manager needs to send the request to TTP with proof. Proof is an official document justifying the mission. Then, with the valid proof, TTP can grant facility manager an access to raw data in storage for the limit periods of time.

Policy definition (P3): we define the role "Facility-security-manager". Users in role "Facility-security-manager" are able to request TTP for accessing raw data in storage. However, they need to provide the proof to justify their request. In addition to that, users need to mention their purpose of request. We define three possible purposes: (1) Internal auditing, (2) Investigation and (3) Observing-suspicious-object. Moreover, to be able to keep track users' activities, users are required to notify system every time they access to raw data.
With above policy description, we are able to mine the following information.

- Role of user: "Facility-security-manager".
- Action: "View".
- Data: "raw-video-in-storage".
- Context: "proof".
- Purpose: "Investigation".

- Obligation: "notify".

With the above information, we can formulate the access control policy for P3 as follows.

PDPA to role "Facility-security-manager": P3

P3= (Facility-security-manager, (View, raw-video-in-storage), Investigation, (proof= yes), Notify=yes))

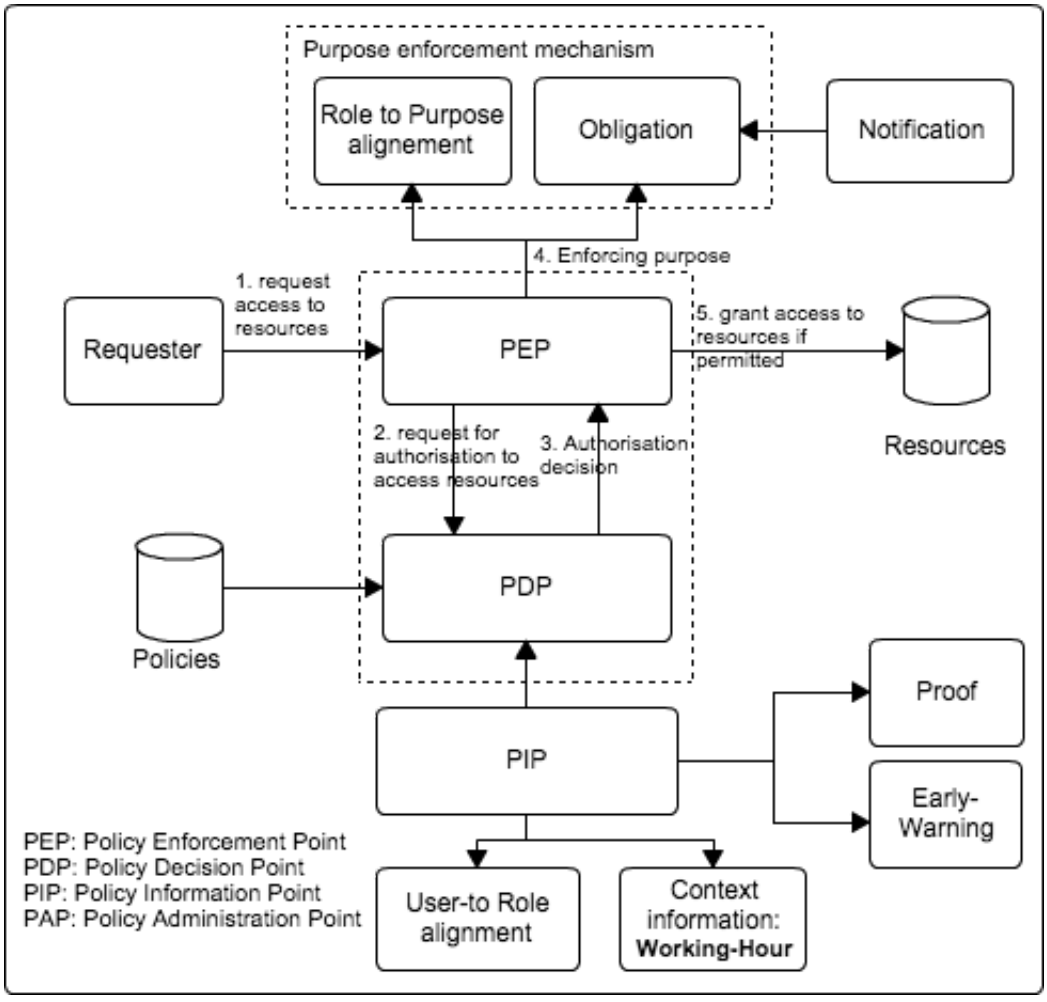

Figure 8.5: Privacy-aware Access Control Architecture for Privacy Preserving Perimeter Protection System

8.6 Access Control: Architecture and Implementation

In this section, we present the access control system architecture supporting purpose enforcement and the implementation of such system in Java.

8.6.1 Access Control Architecture

As illustrated in Figure 8.5, access control system consists of the following components.

1. Policy Enforcement Point (PEP) handles request from user and forwards it to PDP. Beside that it also has other role that is to enforce the policy by using different policy enforcement mechanism.
2. Purpose Enforcement Mechanism: we propose two mechanisms to enforce purpose in the scope of P5 project. They are role to purpose alignment and notification.
    - Role to purpose alignment is the information indicating which roles are allowed for which purposes. For example, security manager is allowed to access raw data in storage for purpose of internal investigation.
    - Notification is a kind of obligation that user needs to fulfil when accessing data (e.g. sending a notification message to data subject every access).
3. Policy Decision Point (PDP) is responsible for validating access control policies with the support of information provided by Policy Information Point.
4. Policy Information Point (PIP) is responsible for providing all needed information to PDP during policy validation phase. It is worth noting that contextual information is also expressed in PIP. In the scope of P5 project, we define four contextual variables.
    - Proof is an official mission document that user needs to provide when requesting access to raw data in storage.
    - Early warning, this module is a part of P5's architecture (see Figure 8.1). It provides a warning message when it detects intruders. The warning message is used as one of the constraints on access permission in case of emergency.
    - User to role alignment provides information concerning the assignment of users to roles.
    - Working hours is the timetable of each user. This contextual information is used as constraint in role admission.

8.6.2 Implementation

We have implemented a context- and privacy-aware access control (CP-RBAC) system based on our proposed architecture (see Figure 8.5) using Java. The implemented system is capable of enforcing the purpose-based policies and is able to validate access control policies in accordance with the access control model we proposed (see Figure 8.4). We also developed PIP module that is able to communicate with other external modules to get the needed information (e.g. getting early warning message from early warning and behaviour analysis module (see Figure 8.1)). We have used XACML version 2 as the format for access control requests, responses and policies [8]. We have also made use of Java Enterprise XACML library [1] as the policy decision point engine. We developed our program in Eclipse Standard/SDK (version Kepler) installed on Macbook air OS version 10.8.4, processor 1.3 Ghz Intel Core i5 with memory 8GB DDR3.

The user-to-role alignment, role to purpose alignment and contextual information are expressed in XML-based documents (they will be replaced by database for final integration of the system). Notification is developed by using mailing module in Java.

8.6.2.1 Testing Inputs and Scenarios

We created 48 access control policies that represent the data permissions for 48 different user roles. We used standard XACML policy language to express all the 48 policies. The policies P1, P2 and P3 (see Section 8.5) are used as the models for the 48 policies. For example, the policy in Figure 8.8 is the formal policy expression in XACML of P1 (see Section 8.5). We performed six different tests with the same request structure (see Figure 8.6), but different number of policies in the policy storages. Since XACML [87] policy engine checks all the policies in the policy storage during policy validation phase, the policy validation processing time depends not only on the complexity of policy, but also on the number of policies in the storage. Thus, Our objective is to observe the policy validation time for each test scenario.

8.6.2.2 Assessment and Validation

There are two criteria we want to assess. The first criterion is the accuracy of the access control system we developed. This means it should provide 100% correct policy evaluation. The second criterion is the time required to evaluate a request. We need it to be as small as possible since our access control system needs to work with real time data. To evaluate second criterion, we created different access control policies with different level of complexity; then we find out the validation time for each request with different number of policies in storage. After several performance tests with the

[1] https://code.google.com/p/enterprise-java-xacml/

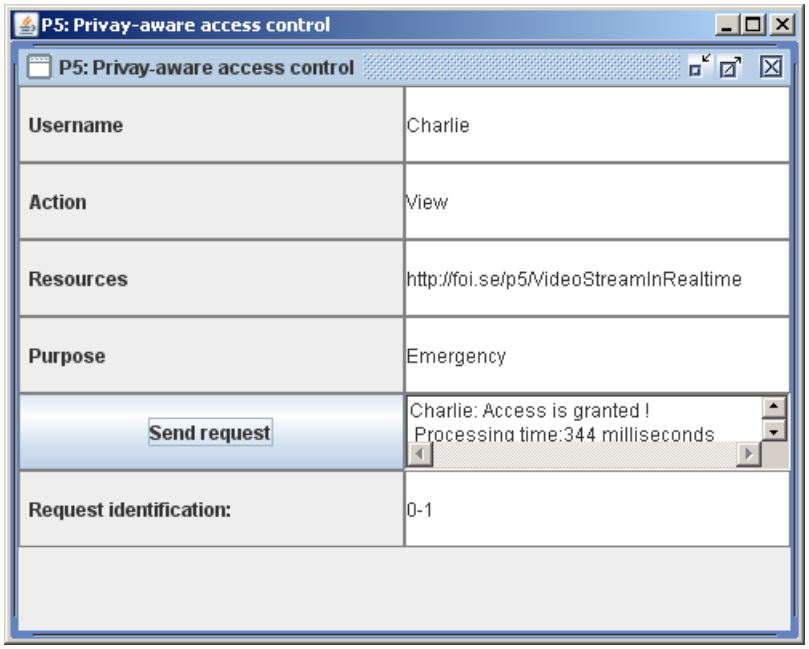

Figure 8.6: A snapshot of a prototype of privacy-aware access control module, it is the user's access request form.

6 different scenarios, we have the result presented in Figure 8.7. We found that as the number of policies in storage increases, the time required to evaluate a request also increase; this is as expected. The first test scenario where there is only one policy (see Figure 8.8) in the policy storage, the time required to validate the policy is 344 milliseconds. With 48 policies, it takes 954 milliseconds (see Figure 8.7).

Given the result in Figure 8.7, we conclude that in order to get small response time for an access request, we need to have small number of policies in storage. Since data permission is generally assigned to role; hence, reducing number of user roles in the system can minimise the number of policies. Other solution is to extend XACML policy decision point engine by adding a module "role to policy mapping" that is able to instruct XACML policy decision point engine to select only the concerned policies from policies storage instead of searching and validating all the policies in the storage. For P5 project, we choose the first option where we minimise number of user roles. We allow only two groups of people to access data: "Guardian" and "Facility-security-manager". Thus, the average response time we can achieve for P5 is 344 ms.

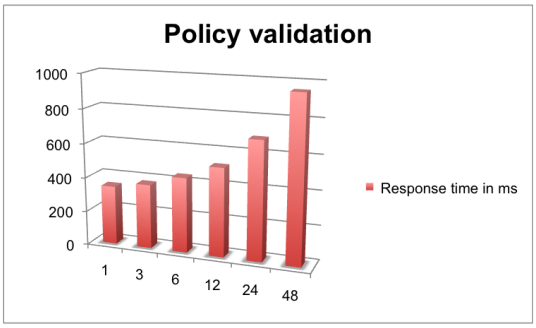

Figure 8.7: The chart shows the relationship between number of policies in storage and policy validation processing time. Axis (X) represents number of policies in storage while axis (Y) is the response time in millisecond.

8.7 Related Work and Contributions

In this section, we discuss two different points for related work. Firstly, we present some existing perimeter protection, threats prevention and detection systems that have similar aims to that of P5 project [32]. Secondly, we discuss the access control models.
Many research projects have been conducted in the area of perimeter protection by using the advanced sensing tools for treats detection and prevention [29][31][30]. However, they do not address the privacy issue in their system. ARENA project [29] aims to develop methods for automatic detection and recognition of threats, based on multi sensory data analysis, for mobile platform such as trucks. But privacy issues have not been addressed in this project. Other project such as CO-FRIEND[30] aims at designing a framework for understanding human activities in real environments, through an artificial cognitive vision system, identifying objects and events and extracting sense from scene observation. Again, CO-FRIEND does not provide privacy preserving feature. ISCAPS [31], another European project, aims at reinforcing security for the European citizen and to downsize the terrorist threat by reducing the risks of malicious events. ISCAPS provides efficient, real-time, user-friendly, highly automated surveillance of crowded areas, which are significantly exposed to terrorist attacks. Some commercial systems such as CIAS [61] is designed for perimeter protection using different type of sensors such as CCTV, but the detection and surveillance depend mostly on human intervention. Moreover, it does not provide privacy preserving feature. The existing systems provide different features, but they are not complete. P5 unifies all the needed features, especially the privacy preserving issue. P5 is also a less human dependable system where object detection and treats analysis are done automatically with the support of fused data from different heterogeneous sensors, such as visual and thermal camera and radar. The system will support, rather than replace, a human operator.

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy
      xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
      http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-context-schema-os.xsd"
      PolicyId="urn:oasis:names:tc:xacml:2.0:conformance-test:IIA10:policy"
      RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
    <Description> Policy for Conformance Test IIA010.</Description>
    <Target/>
    <Rule RuleId="urn:oasis:names:tc:xacml:2.0:conformance-test:IIA10:rule" Effect="Permit">
        <Description>Guardian view video stream</Description>
        <Target>
            <Subjects>
                <Subject>
                    <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Guardian</AttributeValue>
                        <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
                         DataType="http://www.w3.org/2001/XMLSchema#string"/></SubjectMatch></Subject>
            </Subjects>
            <Resources>
                <Resource>
                    <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
                        <AttributeValue
                         DataType="http://www.w3.org/2001/XMLSchema#anyURI">http://foi.se/p5/VideoStreamInRealtime
                         </AttributeValue>
                        <ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
                              DataType="http://www.w3.org/2001/XMLSchema#anyURI"/></ResourceMatch></Resource>
            </Resources>
            <Actions>
                <Action>
                    <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">View</AttributeValue>
                        <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                         DataType="http://www.w3.org/2001/XMLSchema#string"/></ActionMatch></Action>
            </Actions>
        </Target>
        <Condition>
         <Apply  FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
               <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                 <EnvironmentAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:environment:purpose"
                      DataType="http://www.w3.org/2001/XMLSchema#string"/>
               </Apply>
               <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Observing-suspicious-object</AttributeValue>
            </Apply>
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
               <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                 <EnvironmentAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:environment:early-warning"
                      DataType="http://www.w3.org/2001/XMLSchema#string"/>
               </Apply>
               <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">yes</AttributeValue>
            </Apply>
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
               <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                 <EnvironmentAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:environment:notify"
                      DataType="http://www.w3.org/2001/XMLSchema#string"/>
               </Apply>
               <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">yes</AttributeValue>
            </Apply>
         </Apply></Condition></Rule></Policy>
```

Figure 8.8: Formal policy expression in XACML of P1 (see Section 8.5.1)

Access control plays an important role in ensuring that the data generated from privacy concern sensors are well protected. Thus, the careful design of access control model is required to ensure that the created model addresses the requirements. After our thorough study of different access control models, we arrive to the conclusion of using RBAC [82] in P5 project, but with extension. We have also studied different models, such as DAC, MAC [82], ABAC [88] and OrBAC [24][8][17]. In context of P5 environment setting, most of access control models (like DAC and MAC) fail to respond to the requirements since such systems generally have very complex access control policies as we have illustrated in Section V. OrBAC [33] provides more expressing power. However it is not specifically designed for privacy-aware system. The basic RBAC model, where access policy is formulated primarily around role, is also not sufficient. P-RBAC [68][20] is another family of RBAC where the concept of privacy is introduced, but it does not address the contextual information. Moreover, it does not have the concepts of context-aware role admission and personalised role permission. With above reasons, we propose an access control model that takes into account the aspects like privacy [43] and contextual information. To provide privacy preservation feature, the concept of purpose and obligation are introduced and they are well formulated. The two entities (purpose and obligation) are indeed the most important elements required for expressing privacy policies.

Contributions: There are three main contributions in this chapter. Firstly, we propose the privacy preserving perimeter protection system architecture taking into account not only the security of the protected infrastructure, but also the privacy of people who are affected by the surveillance. The second contribution, the most important part in this paper, is the proposed access control model that can be used to express privacy-aware access control policies in privacy preserving perimeter protection system. The third contribution is the implementation of such access control system. It is worth noting that the implemented access control system is integrated into P5 system.

8.8 Summary

In this chapter, we presented a detailed architecture of privacy preserving perimeter protection system, our second application domain for implementing our finding. We also presented the context- and privacy-aware access control model that is designed specifically for such system. The access control system implemented in Java was also presented. Furthermore, we presented a brief description of the role of TTP in P5 system. Our future work is to focus on the development of TTP and the privacy-aware filter module.

Chapter 9

Conclusion

9.1 Our Vision

Given the raise of privacy issues and a demand from government authority for the better protection of private data of individual when sharing them between different parties in the network, many private and state entities are in demand for data usage control system that is able to provide sufficient protection (e.g. healthcare institutions). Early in this research, we thought of using the existing digital rights management technologies to solve the challenge, but after studying the requirements for the protection of personal data of individual (e.g. EU Directive 95/46/EC) and the properties of the existing DRM technologies [71], we came to the conclusion that all the existing DRM technologies do not have sufficient functionalities that respond to the needs for the processing of private data. This is because the existing DRM technologies are not specifically built for private data. They are built to protect commercial contents (e.g. multimedia contents); they are content-specific and lack of generalness. This rules out the possibility of using DRM, without complement or extra support functionalities, to control the processing of private data.

Controlling the usage of private data in distributed environment needs a lot more attention. When we started studying it we saw only a few researches addressing the issue [89][81][72]. Surprisingly, many researches focused on access control, not usage control and assumed that the client side domain is the trusted domain and when data reside on client side domain, they are safe and secured. However, this assumption is not always feasible when we consider the distributed processing environment where data are processed without a direct control of data owner. So far we are not aware of a complete solution that designs for managing and enforcing privacy-aware usage control policies in distributed environment. Consequently, it would be best to design a dedicated system in a way that addresses the processing requirements of such data. We summarise below the responses to our research questions we listed in Chapter 1.

9.2 Summary of Contributions

The central argument of this thesis is that the usage control for private data can be achieved by using the policy-based technique with the support of secured client application; and the enforcement of purpose of use for privacy-aware policies is achieved by controlling the execution of the tasks (and their sequencing) of the workflow representing the purpose in conjunction with the purpose achievement prediction. Our contribution rests on the requirements for access and usage control for private data in distributed environment, a sound purpose modelling for privacy-aware policies, a purpose enforcement technique and a sound mathematical definitions, properties and theorems, which have been implemented in an open source toolset. The prototype implementation produces evidence of efficiency and reliability of our definitions and algorithms. More precisely, the five research questions posted on the thesis objectives (See Chapter 1, Section 1.3.2) have been answered as follows.

RQ1.1: what are the requirements for the protection of private data in distributed system? We focus on distributed healthcare. At the early state of our research, we have participated in a few meetings with Wallonie Healthcare Network (WHN) [85] team, which was responsible for developing the distributed healthcare information system for all the healthcare institutions in French speaking region in Belgium. Our participation provided us a platform upon which we started our work on access and usage control requirements analysis. Beside of working with the real world project like WHN, we had also studied the European laws concerning the protection of private data of individual, especially EU Directive 95/46/EC [28]. We analysed the requirements from WHN and those of EU Directive 95/46/EC and came up with the usage and access control requirements for distributed health. The result of our study was published in the 7th International Conference on Health Informatics, Barcelona, Spain, 2013 [10].

RQ1.2: how to model the purpose of use in such a way so that it can be easily managed and effectively enforced in distributed environment? To grasp a better understanding of the meaning of purpose in the context of access and usage control for private data, we have conducted a formal study of the meaning of purpose in legislation like EU Directive 95/46/EC and the definition of purpose used in the day-to-day basic (e.g. Dictionary). We found that both in legislation and in dictionary, purpose often refers to a future goal, a final destination that someone wants to reach; and to reach the final goal one needs to complete some intermediate tasks. The completion of all the tasks to reach a goal is called the goal achievement or purpose achievement. Based on the definition of purpose, we observe that purpose has similar property to workflow. With this reason we modelled the purpose as the workflow. From that formulation, we built our purpose enforcement technique presented in RQ1.4. The

proposed purpose model was published in the 7th International Conference on Health Informatics, Barcelona, Spain, 2013 [11].

RQ1.3: what access and usage control model should be used to effectively control the usage of private data in distributed environment? And what access and usage policy languages are appropriate to be used in our context? Starting from the requirements in RQ1.1 and the purpose modelling in RQ1.2, we conducted our study on existing access and usage control model from the basic models like DAC and MAC [82] to a more complex model like RBAC and its extension. For usage control model, we studied UCON model [41]. All the models are studied and analysed thoroughly; model by model, we listed its advantages and disadvantages and then compared them against the requirements we identified in RQ1.1 (see Chapter 4, Section 4.5). From our study we concluded that PRBAC is appropriate to be used in our context as access control model where UCON is a good model for usage control. However, UCON needs to be extended in order to fully support our requirements. The result of our work was published in the Fourth International Conference on eHealth, Telemedicine, and Social Medicine (eTELEMED 2012) [11]. For policy language, after our thorough study, we found that XACML is very suitable to be used in distributed setting, not only it is the XML-based language, but also it is the most developed language among all. XACML goes a cross many reviews, implementations and regular updates. XACML is able to not only express access control policy, but also the request and response query. XACML research group has developed Java-enterprise-XACML, a Java-based engine that is able to validate XACML policies. Java-enterprise-XACML is extensible and can be adjusted to our requirements.

RQ1.4: what are the techniques that can be used to enforce privacy-aware usage policy in distributed environment? So far, researchers have used "Auditing" as method to enforce the usage control policies in distributed environment [81][72]. Auditing may be able to detect policy violations after-the-fact, but it cannot prevent unauthorised access. Thus, a posteriori control (e.g. auditing) is not a good choice for enforcing the privacy-aware policies since according to the laws [28], private data are limited to only the authorised people and we need to know the data processor in advance before releasing data to them. Moreover, we need also to ensure that data processor uses data for the purpose it intends for. Given this reason, we focus on a priori control (control happens before releasing data to requester). To enforce privacy polices using formal or automated methods requires a semantics of purpose restrictions to determine whether an action is for a purpose and that purpose could be achieved or not once access permission is granted. We model purpose as a workflow and we argue that an action is for a purpose if and only if that action is part of a plan for the satisfaction of that purpose. Based on that formalisation, we propose an approach to enforce purpose. In our approach, the access authorisation is based on the estima-

tion of the level of certainty of purpose achievement, which is determined by purpose achievement prediction (a probabilistic system estimating how likely user can reach their claimed purpose after access permission is granted). The prediction module is built using association rule learning method where user's access history and contextual information are used as the input data for rule analysis. The semantics of purpose with our enforcement approach enable us to create and implement an algorithm for enforcing the privacy policies.

RQ1.5: what are the existing usage control technologies that can be used to control access and usage of private data in distributed environment? We started from a broad survey of the existing data protection techniques and technologies [70] and studied their varying degree of effectiveness when used in our context. Our survey covered the usage control techniques (e.g. encryption, watermark and digital signature) and technologies (e.g. DRM). Concerning DRM, our finding shows that the existing technologies cannot provide the security we need for protecting private data. We conclude that we need a dedicated system that is able to support our proposed usage control and enforcement technique. Consequently, we proposed a usage control architecture supporting purpose enforcement in Chapter 7. The details of this study can be found in Chapter 2.

9.3 Perspective

Our contributions give a good foundation to the emerging requirements of usage control of private data in distributed environment: from the usage control requirements analysis to usage control model, purpose enforcement mechanism, usage control architecture and prototype of usage control application. These contributions pave a way for the development of a secure platform for the protection of private data in distributed environment. Our prototype implementation of usage control application is more of a proof of concept rather than a customer-ready front-end. As such, it is too generic for practitioners to use it. However, it provides a platform upon which a professional-grade tool can be developed. We discuss below in greater detail the perspectives of this thesis along three dimensions. We firstly discuss the limitations of our purpose enforcement technique and the prototype we implemented. Then, we discuss the interoperability issues. Since our technique is designed for system that uses workflows and the data needs to be shared between different systems in the network, the interoperability issue of the workflow definition is not avoidable. Finally, we explore the avenues for further work on workflow and role mapping.

9.3.1 Limitations

Issue of user access history. Since our purpose achievement prediction technique uses user's access history and contextual information as the input data, the performance of the technique depends largely on user's past activities recorded in access log. Our method fails to conclude in case of new user who has never accessed to the system in the past. However, this issue can be solved by providing a special key to new user. With the special key new user can bypass some security barriers. When special key is used, we need a mechanism to manage such key, but at this state we have not studied yet the issue of special key management. Other solution is to observe the activities of other users in the same role and use those information as the knowledge based on which the decision can be made. Both proposals need to be further studied in our future work.

Probabilistic system. Our purpose enforcement model uses association rule learning method to discover in access log the relation between the requested task and the purpose user claims. The aim of finding the relationship between requested task and claimed purpose is to determine the probability that user could achieve the purpose he claims if the task is allowed to be executed. Since our technique use a probabilistic method, the wrong conclusive result is possible.

Hierarchical purpose. In Section 5.2.2 (Chapter 5), we introduced the concept of multi-purposes and the relationship between purposes. However, the implementation of it has not been included in our prototype. The management of data in case of hierarchical purpose is left for the future work.

Integration between workflow management system and usage control system. Ultimately, we need to integrate the workflow management system into our usage control system. However, at this state we implemented only the usage control system and assumed that we have all the workflow information (see WRM Engine in Figure 7.3, Chapter 7) required for providing to usage control system we developed.

Analogue attack issue. We use policy-based method with support of secure client side application to control and enforce the usage of data. Although our proposed technique can solve most of the issues in usage control, it is not able to protect completely the data from misuse since user can still copy the data by using some tools such as photo or video camera. Thus, we still need a secure data protection at client side control domain. The issue of attack with photo and video cameras was addressed in our unpublished paper in [18]; watermark and digital fingerprint can be the solutions for re-enforcing the usage control of data.

Contextual data mining. In our proposed purpose enforcement technique, con-

textual data play an important role in usage control enforcement. We have provided in Section 6.2.2.1 (Chapter 6) the definition of contextual information related to task and purpose. However, we did not provide a formal method for mining the contextual data for purpose and task. We assume that the contextual data exist for a given set of tasks and purposes. Concerning formal method for contextual data mining, we consider in our future work.

9.3.2 Interoperability

Our main objective is to allow private data to be shared in a secure manner between different systems in the network. Different systems may have different levels of control on data usage, hence, different level of policy complexity; the data management may also be different from system to system. Making them to be able to communicate and share data between each other in the interoperable way is a challenge. One of the interoperability challenges is the user management issue. When data with its usage policy are moved to other system away from the source system, in general, only users stipulated in usage policy could use data if a direct user to data permission assignment method is used. Using direct assignment between user and data permission in context of distributed environment can pose a burden for user as well as policy management since every system in the network needs to have a full list of users, not only the users in their system, but also the lists of users of other systems in the network. Any change in user management structure in a system, other systems in the network need also to update it; this causes the management overload. Seeing the difficulties more than the eases, we propose to use "role to data permission assignment" instead of "user to data permission assignment". Using role allows system engineer to separate the user management domain and the management can be done internally, only "role" needs to be managed globally.

Since different systems may have different role structure, it is possible that the role incompatibility can happen; hence, role-mapping module is required for the systems that have different role structure. We address role-mapping issue in future work.

In Chapter 6, we have clearly stated that our proposed purpose enforcement technique is designed specifically for system that uses workflows. This means that the basic requirement for using our method is that those systems must support workflows. Different systems may have different workflow definition although it may serve the same purpose. This poses other interoperable issue in workflow level. To address this issue we need a workflow-mapping module. The workflow-mapping module will be addressed in future work.

9.3.3 Future Work

Beside of a complete implementation of our usage control system, there are three more issues that need to be addressed as a part of our future work.

Workflow mapping issue. Workflow mapping issue happens when different systems in the networks have different workflow definitions representing the same purpose; the difference can be the number of tasks in the workflow or the names of tasks. For example in healthcare system, two hospitals (A and B) have different procedures for heart surgery; this can mean that the number of tasks in the workflow (or names of the tasks) representing the purpose of heart surgery of system A is different from that of system B. Since the task execution needs to be recorded in access log and the local access log of each system is the input data for purpose enforcement technique, incorrect workflow mapping can lead to wrong purpose achievement prediction value.

Role Mapping Issues. Different systems may have different role management structure. A role's name may exist in one system, but may not exit in other system. Or different names are used to refer to a role. This kind of problem can happen since different system may adopt different management style. Role mapping is an important issue in order to be able to allow systems to work in the interoperable way.

Formal method for contextual data mining. Finding which contextual data goes for which task of the workflow and purpose is an important issue since contextual data is one of the important attributes in usage policies. However, to create a formal method for mining the contextual data for task as well as purpose needs a thorough study. This challenge still requires some work.

References

[1] Mohammad Jafari, Reihaneh Safavi-Naini, and Nicholas Paul Sheppard . Enforcing purpose of use via workflows. In *Proceedings of the 8th ACM workshop on Privacy in the electronic society*, WPES '09, pages 113–116, New York, NY, USA, 2009. ACM. 86, 91, 121, 122

[2] A.Abou El Kalam, R.El Baida, P.Balbiani, S.Benferhat, F.Cuppens, Y.Deswarte, A.Miège, C.Saurel, and G. Trouessin. Organization Based Access Control. In *"4th IEEE International Workshop on Policies for Distributed Systems and Networks (Policy'03)"*, June 2003. 26

[3] Adobe and digital content for e commerce. Website: http://www.adobe.com. latest access: July 2014., 2014. 46

[4] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in largedatabases. *SIGMOD Rec.*, 22(2):207–216, June 1993. 9, 96, 101, 107

[5] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, ICDE '95, pages 3–14, Washington, DC, USA, 1995. IEEE Computer Society. 113

[6] Al-Neyadi, Fahed, Abawajy, and Jemal H. Context-based e-health system access control mechanism. In *Advances in Information Security and Its Application*, volume 36 of *Communications in Computer and Information Science*, pages 68–77. Springer Berlin Heidelberg, 2009. 96, 121

[7] Alexander Pretschner, Manuel Hilty, Florian Sch, Christian Schaefer, and Thomas Walter. Usage control enforcement: Present and future. *IEEE Security and Privacy*, 6:44–53, 2008. 20, 96

[8] Ann. Anderson. XACML profile for Role Based Access Control. *http://www.oasis-open.org/committees/xacml, latest access: July 2011*, February 13 2004. 26, 73, 76, 155, 159

[9] Annanda Th. RATH and Jean-Noël Colin. Protecting personal data: Access control for privacy preserving perimeter protection system". booktitle = The 29th Annual International Federation for Information Processing (IFIP), WG 11.3 Working Conference on Data and Applications Security and Privacy, Fairfax, VA, USA , July 13-15, 2015, springer. 7, 11

[10] Annanda Th. RATH and Jean-Noël Colin. Access and usage control requirements for patient controlled record type of healthcare information system. 2013. 7th International Conference on Health Informatics, Barcelona, Spain, pp.331-336. 4, 9, 50, 61, 68, 78, 80, 162

[11] Annanda Th. RATH and Jean-Noël Colin. A purpose model and policy enforcement engine for usage control in distributed healthcare information system. 2013. 7th International Conference on Health Informatics, Barcelona, Spain, pp. 174-180. 9, 68, 78, 81, 94, 163

[12] Annanda Th. RATH and Jean-Noël Colin. Towards purpose enforcement for privacy policy in distributed healthcare. pages 881– 886, 2013. CeHPSA - 2013 : 3nd IEEE International Workshop on Consumer eHealth Platforms, Services and Applications, Las Vegas, USA. 10

[13] Annanda Th. RATH and Jean-Noël Colin. Towards purpose enforcement model for privacy-aware usage control policy in distributed healthcare. *International Journal of Security and Networks, No. 1, ISSN print: 1747-8405, Publisher: Inderscience.*, 8, 2013. 10, 80

[14] Annanda Th. RATH and Jean-Noël Colin. Modelling and expressing purpose validation policy for privacy-aware usage control in distributed environment. *ACM 8th International Conference on Ubiquitous Information Management and Communication, Siem Reap, Cambodia. ISBN: 978-1-4503-2644-5, DOI:10.1145/2557977.2557991*, 2014. 33, 37

[15] Open Mobile Alliance (OMA) DRM architecture 2.0. Website: http://www.openmobilealliance.org, 2014. 45

[16] Bakar, A. Ismail, R. Ahmad, A.R. Abdul, J.-L. Jais, and J. Coll. Group based Access Control scheme (GBAC): Keeping information sharing secure in mobile adhoc environment. *Digital Information Management, 2009. ICDIM 2009. Fourth International Conference*, 13:1–6, July 2009. 26

[17] Byun Ji-Won, Bertino Elisa, and Li Ninghui. Purpose based access control of complex data for privacy protection. In *Proceedings of the tenth ACM symposium on Access control models and technologies*, SACMAT '05, pages 102–110, New York, NY, USA, 2005. ACM. 120, 121, 122, 159

[18] Annanda Thavymony RATH and Jean-Noël Colin. Analogue attacks in e-health: Issues and solutions. CeHPSA - 2012 : 2nd IEEE International Workshop on Consumer eHealth Platforms, Services and Applications (CeHPSA). Computer Science Faculty, University of Namur. Url: https://directory.unamur.be/staff/rthavymo/publications, 2012. 165

[19] Annanda Thavymony RATH and Jean-Noël Colin. ODRL profile for privacy-aware role-based access control (P-RBAC). Computer Science Faculty, University of Namur. Url: https://directory.unamur.be/staff/rthavymo/publications, 2012. 30, 73, 74

[20] Annanda Thavymony RATH and Jean-Noël Colin. Patient privacy preservation: P-RBAC vs OrBAC in patient controlled records type of centralized healthcare information system. case study of walloon healthcare network, belgium. *The Fourth International Conference on eHealth, Telemedicine, and Social Medicine eTELEMED 2012*, 4:111–118, 2012. 24, 25, 80, 120, 159

[21] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theor.*, 13(1):21–27, September 2006. 10, 96

[22] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital Watermarking and Steganography*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2 edition, 2008. 40

[23] Tith Dara and Annanda Thavymony RATH. Prediction methods: Predicting access goal based on user's access history. *The third international conference on inclusive innovation and innovative management, ICIIIM 2015, Pathumthani, Thailand.*, 25- 26 Nov 2015. 98, 100

[24] D.F.Ferraiolo, R.Sandhu, S.Gavrila, D.R.Kuhn, and R.Chandramouli. Proposed NIST Standard for Role-Based Access Control. In *ACM Transactions on Information and System Security*, pages 4(3):222–274, August 2001. 21, 22, 24, 25, 26, 27, 120, 159

[25] Light Weight digital rights management system. Latest access: March 2013. Website: http://p2pfoundation.net, 2013. 47

[26] High Bandwidth Content Protection(HDCP) DRM. Website: http://www.digital-cp.com/hdcp technologies, 2014. 47

[27] EPAL. *Enterprise Privacy Authorization Language,version 1.2, latest access: January-2013.* http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/. xv, 35, 36

[28] EUdirective. *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.* https://www.cdt.org/privacy/eudirective/EU %Directive%.htmlHD%NM%1. Latest access: March 2012., 1995. 1, 2, 3, 7, 50, 58, 60, 138, 139, 162, 163

[29] European Community Research and Development Information Service. *Automatic detection and recognition of threats, based on multi sensory data analysis, for mobile platform.* http://cordis.europa.eu, latest access: August 2014. 139, 157

[30] European Community Research and Development Information Service. *Cognitive and Flexible learning system, Robust Interpretation of Extended real sceNes by multi-sensors Datafusion.* http://cordis.europa.eu, latest access: August 2014. 139, 157

[31] European Community Research and Development Information Service. *Integrated surveillance of crowded areas for public security including detection of a range of threat scenarios including unattended objects.* http://cordis.europa.eu, latest access: August 2014. 157

[32] European Community Research and Development Information Service, 7 framework program. *Privacy Preserving Perimeter Protection System.* http://www.foi.se/p5, latest access: September 2014. 10, 157

[33] Frédéric.CUPPENS and Nora.CUPPENS. Modeling Contextual Security Policies in OrBAC. *International Journal of Information Security (IJIS)*, 7(4):285 – 305, august 2008. 26, 159

[34] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Mach. Learn.*, 29(2-3):131–163, November 1997. 10, 96, 99

[35] Giovanni Russello, Changyu Dong, and Naranker Dulay. A workflow-based access control framework for e-health applications. *Advanced Information Networking and Applications Workshops, International Conference on*, 0:111–120, 2008. 3, 96

[36] Susanne Guth. A sample digital rights management system. In Eberhard Becker, Willms Buhse, Dirk Gnnewig, and Niels Rump, editors, *Digital Rights Management*, volume 2770 of *Lecture Notes in Computer Science*, pages 150–161. Springer Berlin Heidelberg, 2003. xv, 43

[37] HL7 PHR. *Health Level International Seven.* http://www.hl7.org. Latest access: July 2012., 2011. 3

[38] Hung, Patrick C. K, and Zheng. Yi. Privacy access control model for aggregated e-health services. In *Proceedings of the 2007 Eleventh International IEEE EDOC Conference Workshop*, pages 12–19, Washington, DC, USA, 2007. IEEE Computer Society. 3, 120

[39] FairPlay in iTunes store. Website: http://en.wikipedia.org/wiki/itunes, 2014. 47

[40] MPEG-21 IPMP. Website: http://mpeg.chiariglione.org, 2014. 45, 46

[41] Park Jaehong and Sandhu Ravi. Towards usage control models: beyond traditional access control. In *Proceedings of the seventh ACM symposium on Access control models and technologies*, SACMAT '02, pages 57–64, New York, NY, USA, 2002. ACM. xvi, 27, 61, 68, 70, 80, 163

[42] Jafari Mohammad, Fong Philip , Safavi-Naini Reihaneh, Barker Ken, and Sheppard Nicholas Paul. Towards defining semantic foundations for purpose-based privacy policies. In *Proceedings of the first ACM conference on Data and application security and privacy*, CODASPY '11, pages 213–224, San Antonio, TX, USA, 2011. ACM. 96, 120, 121

[43] Jawad Mohamed, Alvarado Patricia Serrano, and Valduriez Patrick. Design of priserv, a privacy service for dhts. In *Proceedings of the 2008 international workshop on Privacy and anonymity in information society*, PAIS '08, pages 21–25, New York, NY, USA, 2008. ACM. 159

[44] Jean Herveg, Anne Rousseau. *Manuel d'informatisation des urgencies hospitalires.* Presses Universitaires de Louvain (Louvain University Press, 2003), ISBN-10 2-93034-432-6 ISBN-13 978-2-93034-432-4. xv, 52, 53, 55, 83

[45] Barbara Kitchenham. Procedures for performing systematic reviews. url: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.122.3308, 2004. 11

[46] Carl E. Landwehr. Formal models for computer security. *ACM Comput. Surv.*, 13(3):247–278, September 1981. 23

[47] Jun Liu, Jianhui Chen, and Jieping Ye. Large-scale sparse logistic regression. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 547–556, New York, NY, USA, 2009. ACM. 10, 96

[48] Qiong Liu, Reihaneh Safavi-Naini, and Nicholas Paul Sheppard. Digital rights management for content distribution. In *Proceedings of the Australasian Information Security Workshop Conference on ACSW Frontiers 2003 - Volume 21*, ACSW Frontiers '03, pages 49–58, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc. 43

[49] Qiong Liu, Reihaneh Safavi-Naini, and Nicholas Paul Sheppard. Digital rights management for content distribution. In *Proceedings of the Australasian Information Security Workshop Conference on ACSW Frontiers 2003 - Volume 21*, ACSW Frontiers '03, pages 49–58, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc. 49

[50] Lorenzo D. Martin, Qun Ni, Dan Lin, and Elisa Bertin. Multi-domain and privacy-aware role based access control in e-Health. *IEEE, Second International Conference on Pervasive Computing Technologies for Healthcare*, (10090047):131 – 134, Jan. 30 2008-Feb 2008. 3

[51] Sean Meyn and Richard L. Tweedie. *Markov Chains and Stochastic Stability*. Cambridge University Press, New York, NY, USA, 2nd edition, 2009. 97

[52] Mohammad H. Yarmand, Kamran Sartipi, and Douglas G. Down. Behavior-based access control for distributed healthcare environment. *Computer-Based Medical Systems, IEEE Symposium*, pages 126–131, 2008. 3

[53] Michael Zur Muehlen. Resource modeling in workflow applications. In *Proceedings of the 1999 Workflow Management Conference (WFM99)*, pages 137–153, 1999. 87

[54] Walloon Healthcare Network. Règlement relatif à la protection de la vie privée (regulations for the protection of privacy). https://www.reseausantewallon.be/Documents%20partages/R%C3%A8glement%20Vie%20Priv%C3%A9e.pdf, latest access: March 2012, 2010. 2, 7, 9, 52, 60, 61, 63, 64, 138

[55] CORPORATE NIST. The digital signature standard. *Commun. ACM*, 35(7):36–40, July 1992. 42

[56] ODRL. *Open Digital Right Expression Language,version 2.0. Latest access: January-2013*. http://www.w3.org/community/odrl/two/model/. xv, 28, 30, 31, 32

[57] Definition of purpose. Website: http://opensdrm.allofads.com, 2014. 82

[58] Definition of the word planning. Website:http://en.wikipedia.org/wiki/planning, 2014. 81

[59] Office for Civil Rights. *Summary of the HIPAA privacy rule. OCR Privacy Brief, U.S. Department of Health and Human Services*. 1, 2, 58

[60] Ken Peffers, Tuure Tuunanen, Marcus Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. *J. Manage. Inf. Syst.*, 24(3):45–77, December 2007. 3

[61] Perimeter Protection Company. http://www.cias.it/, latest access: September 2014. 139, 157

[62] Gregory Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In *Knowledge Discovery in Databases*, pages 229–248. AAAI/MIT Press, 1991. 101

[63] Chung ping Wu and C.-C. Jay Kuo. Fast encryption methods for audiovisual data confidentiality. In *Multimedia Systems and Applications III, ser. Proc. SPIE*, pages 284–295, 2000. 41

[64] Press play DRM. Website: http://apps.microsoft.com/windows/en-us/app/pressplay-video/14524998-116a-406f-994b-fefc4caa91dc, 2014. 45

[65] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994. 10, 96, 97

[66] Daniel Pahler and Rüdiger Grimm. A formal digital rights model without enforcement. *9th International Workshop for Technical, Economic and legal aspects of Business Models for Virtual Goods*, pages 75–89, 2011. 4, 50

[67] J. R. Quinlan. Introduction of decision trees. *Mach. Learn.*, 1(1):81–106, March 1986. 10, 96, 98

[68] Qun Ni, Dan Lin, Elisa Bertino, and Jorge Lobo. Conditional privacy-aware role based access control. In *European Symposium on Research in Computer Security*, pages 72–89, 2007. 120, 159

[69] Annanda Thavymony RATH and Jean-Noël Colin. A survey of existing rights expression and access control markup languages: Copyright, contract and license management and control over access and usage. Computer Science Faculty, University of Namur. Url: https://directory.unamur.be/staff/rthavymo/publications, 2012. 28, 30, 32, 33, 37, 73

[70] Annanda Thavymony RATH and Jean-Noël Colin. State of the art in usage control enforcement techniques. Computer Science Faculty, University of Namur. Url: https://directory.unamur.be/staff/rthavymo/publications, 2013. 39, 40, 41, 42, 164

[71] Annanda Thavymony RATH and Jean-Noël Colin. Taxonomy and state of the art in digital rights management technologies. Computer Science Faculty, University of Namur. Url: https://directory.unamur.be/staff/rthavymo/publications, 2013. xv, 3, 4, 28, 42, 48, 49, 50, 161

[72] Christoph Ringelstein. *Data Provenance and Destiny in Distributed Environment.* University of Koblenz, Germany. Url: https://kola.opus.hbz-nrw.de/frontdoor/index/index/year/2012/docId/591, 2011. 55, 60, 161, 163

[73] Christoph Ringelstein and Steffen Staab. Logging in distributed workflows. Technical Report, ISWeb Working Group, University of Koblenz, Germany. Url: http://ceur-ws.org/Vol-320/paper3.pdf, 2009. 39

[74] Sheldon M. Ross. *Introduction to Probability Models, Ninth Edition, isbn = 0125980620.* Academic Press, Inc., Orlando, FL, USA, 2006. 100

[75] Rostad, Lillian, and Edsberg Ole. A study of access control requirements for healthcare systems based on audit trails from access logs. In *Proceedings of the 22nd Annual Computer Security Applications Conference*, pages 175–186, Washington, DC, USA, 2006. IEEE Computer Society. 96

[76] John A. Simpson and Edmund S. C. Weiner. In the oxford english dictionary. oxford university press, 2nd edition, 1989. 81

[77] AXMEDIS specification and consortium. Website: http://www.axmediatech.com, 2014. 47

[78] Chillout specification. Website: http://chillout.dmpf.org, 2014. 49

[79] OpenSDRM specification. Website: http://opensdrm.allofads.com, 2014. 49

[80] Intertrust technologies corp. Website: http://www.intertrust.com, 2014. 49

[81] Michael Carl Tschantz, Anupam Datta, and Jeannette M. Wing. Formalizing and enforcing purpose restrictions in privacy policies. In *IEEE Symposium on Security and Privacy*, pages 176–190. IEEE Computer Society, 2012. 120, 161, 163

[82] Vincent C. Hu, David F. Ferraiolo, and D. Rick Kuhn. Assessment of access control system. *National Institute of Standards and Technology*, September 2006. 19, 20, 21, 147, 159, 163

[83] WENJUN ZENG, HEATHER YU, and CHING-YUNG. *Multimedia Security Technology for DRM.* ACADEMIC PRESS, 978-0-12-369476-8, 2006. 3

[84] Ryen W. White, Peter Bailey, and Liwei Chen. Predicting user interests from contextual information. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 363–370, New York, NY, USA, 2009. ACM. 107, 109

[85] WHN, 2009. Espace développeur de RSW (Walloon Healthcare Network): https://www.reseausantewallon.be/ developpement/ default.aspx, latest access: January 2012. 52, 53, 162

[86] Microsoft windows media rights management. Latest access June 2014. Website: http://www.microsoft.com, 2014. 45

[87] XACML. *eXtensible Access Control Markup Language,version 3.0. latest access: January-2013.* http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-en.html. xv, 10, 33, 76, 77, 128, 130, 155

[88] Eric Yuan and Jin Tong. Attribute based access control a new access control approach for service oriented architectures (soa). *Workshop, Ottawa, ON, Canada*, April 2005. 26, 159

[89] Zhang Xinwen, Parisi-Presicce Francesco, Sandhu Ravi, and Park Jaehong. Formal model and policy specification of usage control. *ACM Trans. Inf. Syst. Secur.*, 8:351–387, November 2005. 96, 161

[90] Gansen Zhao and David W. Chadwick. On the modeling of bell-lapadula security policies using rbac. In *Proceedings of the 2008 IEEE 17th Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, WETICE '08, pages 257–262, Washington, DC, USA, 2008. IEEE Computer Society. 23