An Approach for Managing Access to Personal Information Using Ontology-Based Chains

Esraa Omran

A thesis submitted in partial fulfilment of the

requirements of the University of Sunderland

for the degree of Doctor of Philosophy

February 2013

Abstract

The importance of electronic healthcare has caused numerous changes in both substantive and procedural aspects of healthcare processes. These changes have produced new challenges to patient privacy and information secrecy. Traditional privacy policies cannot respond to rapidly increased privacy needs of patients in electronic healthcare. Technically enforceable privacy policies are needed in order to protect patient privacy in modern healthcare with its cross organisational information sharing and decision making.

This thesis proposes a personal information flow model that specifies a limited number of acts on this type of information. Ontology classified Chains of these acts can be used instead of the "intended/business purposes" used in privacy access control to seamlessly imbuing current healthcare applications and their supporting infrastructure with security and privacy functionality. In this thesis, we first introduce an integrated basic architecture, design principles, and implementation techniques for privacy-preserving data mining systems. We then discuss the key methods of privacy-preserving data mining systems which include four main methods: Role based access control (RBAC), Hippocratic database, Chain method and eXtensible Access Control Markup Language (XACML). We found out that the traditional methods suffer from two main problems: complexity of privacy policy design and the lack of context flexibility that is needed while working in critical situations such as the one we find in hospitals. We present and compare strategies for realising these methods. Theoretical analysis and experimental evaluation show that our new method can generate accurate data *mining models and safe data access management while protecting the privacy of the data being mined. The experiments followed comparative kind of experiments, to show the ease of the design first and then follow real scenarios to show the context flexibility in saving personal information privacy of our investigated method.*

Keywords: Patient privacy policy, Chain method, Policy enforcement, ontology, Privacy policy framework

Dedication

To Allah Almighty who bestows all love, graces, happiness and success in my life;

To the prophet Mohammed who guided me to happiness and success in my life;

To my mother Aisha Dawood the meaning of love and happiness in my life;

To my father Chassib Omran who encouraged me to continue my lifelong dream;

To my sister and brother Alaa and Mohammed who share with me all the lovely and difficult times with love and care.

My special dedication goes to my faithful friend Inas Mahfouz who always supports me with great care and pushes me towards success.

Finally I would like to dedicate this thesis to my beloved cat “Tota”.

Acknowledgments

I would like to express my sincere appreciation and gratitude to my dissertation director, Dr David Nelson who made this work possible, and encouraged me during my difficult times.

Many thanks to Dr Albert Bokma, who has enlightened and guided me throughout my doctoral studies. Special appreciation and thanks to Dr Shereef Abu Almaati, who has advised and supported me during my years of doctoral study.

My appreciation also goes to Dr Tyrone Grandison- Research manager in IBM and Dr Nicola Zannone from Eindhoven University for their help and collaboration in a number of my publications. Their comments and suggestions are very constructive, and I look forward to working with them in the future.

Table of Contents

List of Figures

Chapter 1
Introduction

1.1 Background

Information systems are a pervasive feature of everyday life and most of the services people use would cease to be able to function without them. There are great benefits associated with them in terms of quality, speed and ubiquity of service delivery. Information systems are involved when people use the phone, the Internet, financial services such as insurance and banking, and even shopping. Increasingly, information systems are used in public services, such as education and healthcare.

To deliver these services they need to hold increasing amounts of personal information including personal details, service related usage and history information, which are needed for the associated customer service and billing information. In addition, information services display a potential for other purposes such as marketing. At times this information is passed on to third parties for vetting purposes, third party service support and government agencies where there are reporting requirements (e.g. sometimes governments need to check legal issues about suspected people).

The fact that these service providers and their information systems are ever more accessible over the Internet has benefits in terms of accessing up-to-date information remotely or connecting systems together to deliver improved and more sophisticated services. However, there is a downside to these developments in terms of increased exposure of systems to the open Internet and by consequence of hacking. In addition, there are also potential problems in terms of undesirable disclosure of the information these systems hold to third parties when companies do not purely use the information for service delivery purposes and sell on records to other companies for profit. What this thesis seeks is that only authorised people would get the exact and correct information at the right time and for the intended purpose. And this information shouldn’t be disclosed to people inside the organisation who have no right or reason to access it. Furthermore, such information should not be disclosed to third parties for the same reason. What the researcher is looking for is a way to ensure that this occurs in information systems and is not left purely to chance. In this chapter, the researcher highlights the following topics briefly:

- The definition of Personal Information and the need for Protecting it
- The specific case of the Healthcare
- The Need for Technical Enforcement to Data Protection
- Focus of the investigation
- Thesis organisation

1.2 Personal Information and Security Implications

1.2.1 Definition of Personal Information

This section will mainly discuss issues related to personal information and the variant definitions of personal information in order to determine the definition that most clearly meets the thesis objectives.

Personal information, in popular understanding, is a term whose scope varies significantly from person to person, from law to law and from Act to Act. This section will highlight the most significant personal information from the literature.

'Personal information' is defined by the Information Privacy Act (Data Protection Act 1998) to mean: *"Information or an opinion, including information or an opinion forming part of a database, whether true or not, and whether recorded in a material form or not, about an individual whose identity is apparent, or can reasonably be ascertained, from the information or opinion."*

A decisive element in this definition is that personal information must be about an individual whose identity is apparent, or can reasonably be ascertained. Justice Underhill of the United States District Court (Connecticut) defines it as follows:

> *"Personal information, in the constitutional sense (due process), is information about an individual that, if widely known, would reasonably cause that individual embarrassment, discomfort, or concern."*

This law also focuses on the legal part of personal information to protect it. However, it does not give a comprehensive definition of personal information.

While the PIPEDA Act (University of Alberta, Health Law Institute, University of Victoria and School of Health Information Science 2005) protect personal information by including information in its definition such as the following:

- age, name, income, ethnic origin, religion or blood type;
- opinions, evaluation, comments, social status or disciplinary actions;
- credit records, employment history and medical records.

That helps PIPEDA's personal information definition to be one of the most comprehensive personal information definitions and the researcher would quote some of its clauses during the system design process.

In fact, any kind of information that is somehow related to a person can be regarded as personal information. This is described by (Jones, 2008), where he defines six (sometimes overlapping) classes of information based on their relationship to a person or proprietor:

- Information that is controlled or owned by person;
- Information that is about a person or proprietor;
- Information that is directed to a person or proprietor;
- Information that is sent, posted or provided by a person or proprietor;
- Information that has been already experienced by a person or proprietor;
- Information that is relevant or useful to a person or proprietor.

Therefore, personal information is any information or opinion about an identifiable person. Personal information is divided into:

- Name, such as full name, maiden name, mother‘s maiden name, or alias
- Personal identification number, such as social security number (SSN), passport number, driver‘s license number, taxpayer identification number, or financial account or credit card number
- Address information, such as street address or email address
- Personal characteristics, including photographic image (especially of face or other identifying characteristic), fingerprints, handwriting, or other biometric data (e.g., retina scan, voice signature, facial geometry. This could include:

    - Written records about a person
    - Photograph or image of a person
    - Fingerprints or DNA samples that identify a person
    - Information about a person that is not written down, but which is in the possession or control of the agency.

As long as information is being communicated to someone who can identify the person it is about, the information may meet the definition of ‘personal information’. The more details that are given about a person, and the wider the audience, the more likely it will be that it will amount to ‘personal information’.

As seen in Figure 1, the personal information definition could cover a huge amount of different information about a specific person such as their name, address, passport number, date of birth, phone number or bank account. Each piece of information could cause endless trouble for their proprietor if it has been disclosed to unauthorised people.

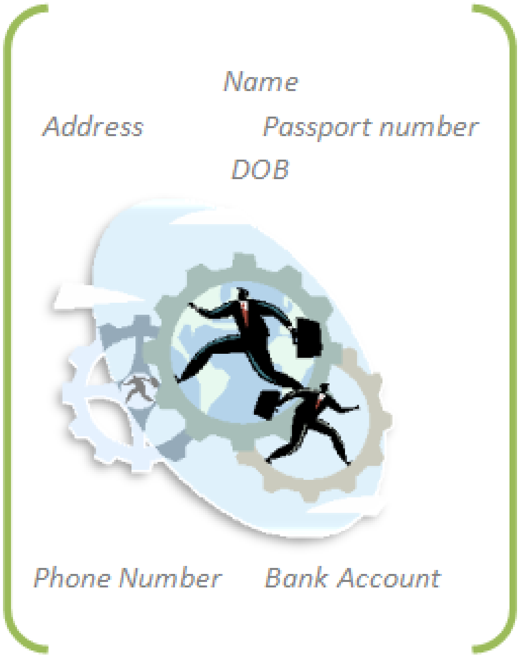


Figure 1: Personal information is a key in our daily life

A set of PI definitions has been analysed to shape the personal information definition used in this thesis. To defeat identity theft, developing a robust data access management method is needed. Therefore the researcher needs to follow on Justice Underhill of the United States District Court PI definition and the clauses of HIPAA and PIPEDA and classes founded in (Jones 2008) mentioned above while shaping the system design. The researcher will use some clauses of HIPAA and PIPEDA to construct the semantic layer and its rules. Clauses of these PI definitions will be used later in designing the system ontology and access requirements.

In this thesis the personal information is defined as “Any information that is of importance to a person and which the person is interested in keeping track and privacy of and its malicious disclosure could harm that person”, which is adopted from the definition given in (Larsen, 2005).

In next section a discussion of why it is needed to protect personal information, its impact on information systems, and the most significant legalisations to protect privacy.

In order to be able to appreciate the extent of the problem, one should consider what Personal Information comprises. Personal Information, or what is also frequently termed Personally Identifiable Information (PII), is used to refer to information that can be used to uniquely identify a specific person or can be used with other sources to uniquely identify a single individual. The PPIP Act and HRIP Act (Privacy NSW Privacy Management Plan, 2006) define 'Personal Information' as "*information or an opinion (including information or an opinion forming part of a database and whether or not recorded in a material form) about an individual whose identity is apparent or can reasonably be ascertained from the information or opinion*."

1.2.2 The Need for Protection

The information stored in service organisation systems about individuals and their usage of these services, should ideally be used for service delivery and billing purposes. Access to this information should be restricted to preserve the privacy of the individuals. Special care should be taken to keep this information from other organisations with information about those individuals. As such, disclosure of this information may disadvantage those individuals at best, or worse still, cause concrete damage and harm (In May 2006, an employee of the US Department of Veteran Affairs took a laptop home without authorisation from the department. The laptop and the sensitive personal data of 26.5 million people who were discharged from the US military since 1975 it contained, were stolen during a burglary at the employee's home. Included in the data were veterans' names, Social Security numbers and dates of birth.). Most governments recognise the individual’s right to protection against loss of privacy and insider trading and fraud and other forms of criminal activity against a person and their estate. Most countries have enshrined this in the form of data protection and privacy protection legislation laws (see Chapter 2 for examples of such legalisations and laws). These laws govern what types of Personal Information can be held by organisations as well as restrictions about safe-keeping, disclosure and ways in which this information can be used.

This legislation, as well as any self-imposed standards of corporate governance and professional conduct, requires organisations to put safeguards in place which ensure the safe keeping and appropriate use of personal information in their information systems, databases and on their respective servers and networks. This is usually done through a combination of policies and technologies. The organisation’s compliance is ensured by information officers, IT managers and systems programmers who strive to ensure that the information is used in accordance with legislation and professional practice. In addition, they ensure that the systems impose appropriate restrictions on data access and proliferation and are secure enough to provide for safe-keeping. The implementation of these restrictions can be an onerous task if the access requirements are complex and the information concerned is highly sensitive. The available tools, methods and systems to implement this, such as prevalent access control approaches, are only partially equipped to solve this problem (see Chapter 3) and more sophisticated approaches are required to comply with the appropriate restrictions. This problem is what this thesis aims to address.

The work in this thesis could be applied to infinite number of domains. But healthcare has been chosen because of the vital and clear importance of managing the access to the sensitive information in this domain while keeping it available to authorised people.

1.3 The Specific Case of Healthcare

From ancient civilisations the importance of privacy has been recognised as essential to patient-physician relationships as stated in the Hippocratic Oath: "*What I may see or hear in the course of the treatment or even outside of the treatment in regard to the life of men, which on no account one must spread abroad, I will keep to myself, holding such things shameful to be spoken about*" (Agrawal *et al.* 2002). since that time governments and organisations have sought for ways to protect personal information in the healthcare sector. Therefore, healthcare is one of the areas in which these issues are most prevalent. The digitalisation of this information and its availability through the international and local networks makes issues such as privacy and security of sensitive information much harder to control. As there is movement towards interoperable electronic health records (where digital copies of the medical information can be interchanged between different authorised peers such as the hospital and the insurance company), there will be both new challenges and new opportunities in protecting the privacy and security of health information.

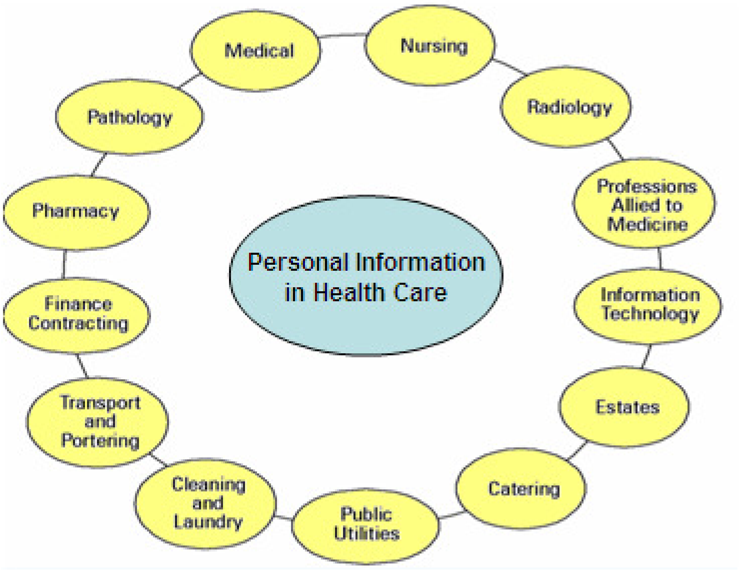


Figure 2: Personal information in health care

Figure 2 shows the main professionals involved in the healthcare process. All those professionals are necessary to make the most accurate diagnoses and provide the best treatment and health service. Patient information may be shared with others, such as: insurance companies; pharmacies; researchers; and employers, for many reasons, for example: paying the bill for a patient; delivering medicines and making quality research. Health records of patients are at the very centre of service delivery and a considerable number of professionals will need to access and contribute to these records over the lifetime of a patient. In healthcare, Personal Information is a collection of records that need different levels of protection and this depends on the context such as appointments, referrals, surgery, etc. Dealing with these medical records is a very critical issue, as they contain sensitive information about the patient and could end or destroy one's life if misused.

1.3.1 Example of Healthcare in Practice

This section introduces a typical scenario for healthcare provision in a hospital environment. Privacy is an underlying governing principle of the patient – physician relationship for effective delivery of healthcare. Patients are required to share information with their physicians to facilitate correct diagnosis and determination of treatment, especially to avoid adverse drug interactions. However patients may refuse to divulge important information in cases of health problems such as psychiatric behaviour and HIV, as their disclosure may lead to social stigma and discrimination (Applebaum, 2002). Over time, a patient's medical record accumulates significant personal information including: identification; history of medical diagnosis; digital renderings of medical images; treatment received; medication history; dietary habits; sexual preference; genetic information; psychological profiles; employment history; income; and physicians' subjective assessments of personality and mental state among others (Mercuri, 2004).

Figure 4 shows a typical information flow in the healthcare system. Patient health records can serve a range of purposes apart from diagnosis and treatment provision. For example, information can be used to improve efficiency within the healthcare system, drive public policy development and administration at state and federal level, and in the conduct of research to advance medical science (Hodge, 2003). A patient's medical records are also shared with other organisations such as medical insurance, to handle payment of services rendered by physicians. Healthcare providers may use records to manage their operations, to assess service quality, and to identify quality improvement opportunities. Furthermore, providers may share health information with other healthcare organisations as they collaborate to provide patient support and with governments for statistical purposes.

The scenario shown in Figure 3 is based on the activities carried out by the International Clinic (IC) in Kuwait (see Chapter3).

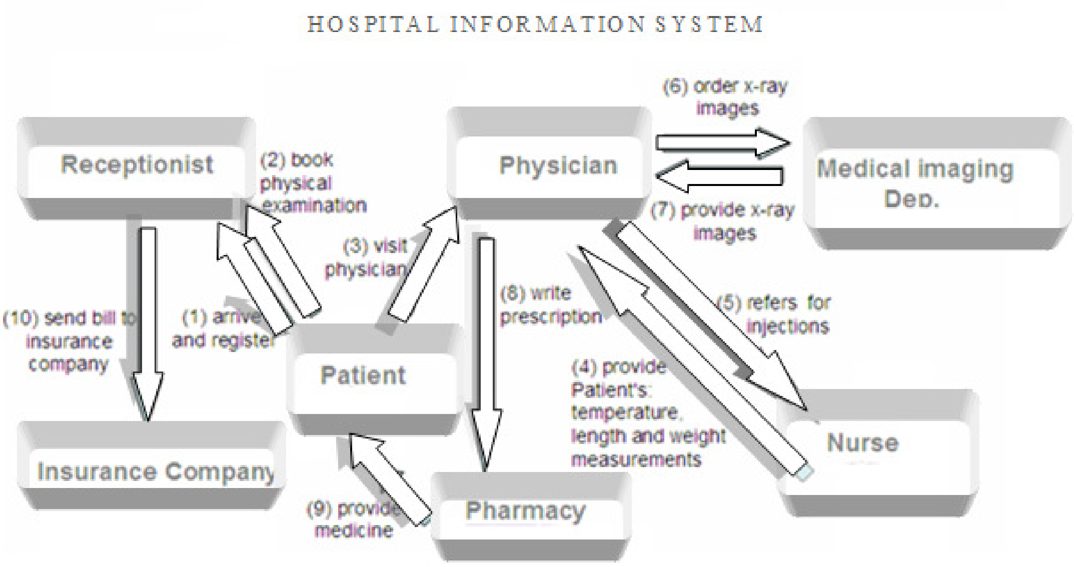


Figure 3: Healthcare provision in the International Clinic, Kuwait

As shown in Figure 3, the process in hospitals is complicated and therefore needs a reliable system. One patient record could, for example, be transformed between different parties such as: the receptionist who would collect his information first to register him in this hospital. He would then transfer his information to the physician, who could also transfer him to other parties such as the laboratory to take some X-rays or for the nurse to give him some injections. Then he would return back again to the receptionist to book another appointment. The hospital could also send some of his information to the insurance company.

Considering that in each situation, there are a number of different sensitive information records which are either added to or updated, it can be seen how crucial the problem in hand is. Also, each user should be authorised to access only a portion of the personal information that is related to their role in a specific situation. For example in the case of a doctor, receptionist and nurse: the doctor can process (edit) the medical record of the patient, he can then create (write) a prescription but can't register a new patient because this is the allowed function for the receptionist. The receptionist can (collect) information from the patient to register him. But the doctor can see some of the attributes of the registration file such as (age, gender, etc...) but he can't process (edit) this information.

The nurse can also collect information from the doctor such as the patient temperature and weight. But the nurse can't view the medical record written by the doctor nor can she edit it. Also, she can't write a prescription to the patient as will be explained in Chapter 3 (See Chapter 3 for the full set of scenarios).

The typical scenario above illustrates that medical information systems are a good example of the complexity of privacy design issues and personal information management, and how controlling access to these systems becomes a vital issue.

The way that policies are defined in today's information systems is highly inflexible (Al-Fedaghi *et al.,* 2005). There is a lack of flexible, composable constructs for expressing policies. Any modification in the policy architecture is very hard to incorporate without affecting the rest of the components. Consequently there is a need to represent policies using constructs and in a manner such that performing policy analysis and propagating changes should be comparatively easy.

The next section provides an overview of privacy legalisations needed for privacy protection. The outcomes and expected results will be highlighted to draw out the introductory research question and hypothesis (this hypothesis will be discussed in detail in Chapter 4).

1.4 The Need to Data Protection

1.4.1 Existing Legislation on Privacy Protection

The issues with personal information in the age of the Internet have led to a large amount of legislation in different countries and for illustration purposes the researcher will give examples of the legalisation set in Canada, the USA and the UK to protect personal information:

Canada

The most outstanding law to protect personal information privacy in Canada is The Personal Information Protection and Electronic Documents Act (PIPEDA). PIPEDA was enacted to establish national rules for personal information protection in the private sector and establishes, as law, the Canadian Standards Association's Model Code for the Protection of Personal Information, which encompasses the following principles: accountability; identifying purposes; consent; limiting collection; limiting use, disclosure, and retention; accuracy; safeguards; openness; individual access; and challenging compliance (University of Alberta, Health Law Institute, University of Victoria and School of Health Information Science, 2005).

PIPEDA has been phased into effect over three years: 2001, 2002 and 2004. PIPEDA defines personal information to mean identifiable information about an individual and personal health information is defined from (University of Alberta, Health Law Institute, University of Victoria and School of Health Information Science 2005) as follows:

(a) Information concerning the physical or mental health of the individual;
(b) Information concerning any health service provided to the individual;

(c) information concerning the donation by the individual of any body part or any bodily substance of the individual or information derived from the testing or examination of a body part or bodily substance of the individual;

(d) Information that is collected in the course of providing health services to the individual; or

(e) Information that is collected incidentally to the provision of health services to the individual.

Taking account of the above principles and consulting experiences from the International Health clinic in Kuwait, the researcher has investigated Personal Health Information Ontology as will be shown in Chapter 6 and Chapter 7.

United States

A constitutional right to privacy is inferred from the Fourth Amendment; and specific federal privacy statutes. It dates back to 1890, when Samuel Warren and Louis Brandeis published their seminal work (Warren and Brandeis, 1890), The Right to Privacy, recognising a “right to be let alone,” Privacy was enforceable through legal protection from “injurious disclosures as to private matters.”

In a growing number of jurisdictions, the term personal information is defined by local statute, typically within the context of an attempt by the legislative assembly to protect individuals from careless storage or release of information about them.

In Security Industry and Financial (Legal directory, 2012), Justice Underhill of the United States District Court (Connecticut) wrote:

> *"Personal information, in the constitutional sense (due process), is information about an individual that, if widely known, would reasonably cause that individual embarrassment, discomfort, or concern."*

In State v Reid (Legal directory, 2012), Justice Weissbard of the appeal division of the Superior Court of New Jersey adopted these words:

> *"Informational privacy has been variously defined as shorthand for the ability to control the acquisition or release of information about oneself ... or an individual's claim to control the terms under which personal information is acquired, disclosed, and used.*
>
> *"In general, informational privacy encompasses any information that is identifiable to an individual. This includes both assigned information, such as a name, address, or social security number, and generated information, such as financial or credit card records, medical records, and phone logs...."*

Data privacy is not highly legislated or regulated in the U.S. In the United States, access to private data contained in for example third-party credit reports may be sought when seeking employment or medical care, or making automobile, housing, or other purchases on credit terms. Although partial regulations exist, there is no all-encompassing law regulating the acquisition, storage, or use of personal data in the U.S. In general terms, whoever can be troubled to key in the data is deemed to own the right to store and use it, even if the data was collected without permission. Examples of US laws to protect privacy are: the Health Insurance Portability and Accountability Act of 1996, the Children's Online Privacy Protection Act of 1998, and the Fair and Accurate Credit Transactions Act of 2003 (US State Privacy Laws, 2010).

Although Personal information exposure as a result of a private-entity data breach does not infringe upon constitutional rights, the constitutional right to privacy influences the overall approach to legal protections of privacy in the United States. As discussed above, the evolution of the right to privacy in the United States does not incorporate personal information, and the federal privacy laws so far enacted only address specific types of data and are often not applicable to exposures of personal information.

United Kingdom

The Data Protection Act (1998) is a United Kingdom Act of Parliament which defines UK law on the processing of data on identifiable living people. It is the main piece of legislation that governs the protection of personal data in the UK. Although the Act itself does not mention privacy, it was enacted to bring UK law into line with the EU data protection directive of 1995 (Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995, 1995) which required Member States to protect people's fundamental rights and freedoms and in particular their right to privacy with respect to the processing of personal data. In practice it provides a way for individuals to control information about themselves. Most of the Act does not apply to domestic use (Data Protection Act, 1998) for example keeping a personal address book. Anyone holding personal data for other purposes is legally obliged to comply with this Act, subject to some exemptions. The Act defines eight data protection principles. It also requires companies and individuals to keep personal information to them.

1.4.2 Problems Facing Privacy Legalisations and Rules

As discussed in the previous section, laws to protect privacy of personal information in large countries such as the USA and Canada lacks the completeness and the tools to enforce them.

The PIPEDA law is one of the most important laws that have been set to preserve the privacy of personal information especially in the health care domain. It clauses look at the problem from different perspectives unlike the laws that have been discussed in the US section, and each one focuses on a specific part of personal information and tries to protect it. For example the Health Insurance Portability and Accountability Act is concerned with preserving the personal health information while the Children's Online Privacy Protection Act preserves the right of children and their parents information to be protected while using the internet.

Current laws do not help consumers who are trying to limit the collection, use, dissemination, and misuse of their Personal Information. Victims of privacy violations have no cause of action unless they can show direct loss as a result of unauthorised use of their personal information, while privacy violation notification laws only indirectly encourage encryption of data. Privacy laws are also not well-suited to personal information. Better privacy policies can lead to more visitor awareness of personal information. Better awareness of these personal information-handling practices can lead to visitors being more careful before submitting personal information to unauthorised people or organisations that may not protect it adequately or who may sell it on the open market. This type of privacy protective behaviour could give organisations more of an incentive to protect personal information in order to maintain business that would be lost under their current privacy regimes.

In conclusion, government organisations and companies must design, implement, and maintain adequate security systems to protect personal information. Based on the continuous reports of privacy violation, companies have yet to be properly motivated to implement such systems. Eventually, governments must pass legislation that would at least require comprehensive internal data protection procedures and systems, coupled with substantial fines for failing to implement and maintain such procedures and systems. This would not only continue the privacy violation notification requirements already in place in most countries, but also mandate adequate privacy preserving systems, and include the fines necessary to give organisations and companies the proper incentive to put those programs in place.

In the next section, problem of saving privacy in healthcare is presented as this is the domain that has been chosen in this thesis to be applied in the proposed principle of data access. This is because this domain is rich in sensitive data that should be protected.

As previously discussed, there is a vital need to find a means of protection to Personal information especially in sensitive domains such as healthcare. Traditional non-technical methods do not provide a solution to the privacy violation problem in the age of digitalisation and semantics (See Chapter 2).
Information systems are a collection of integrated applications that manage the work of the enterprise databases and control the flow of information from and into the enterprise.

Hacking into databases can give someone access to sensitive data and to its unintended disclosure. This encourages the need for increased protection at source – i.e. at the database level using technical approaches that enforce data protection policies while taking care of the context (situation-user combination). Privacy rules cannot be set without ensuring their application in the face of a flow of thousands of users who want to access different information at specific times, and at the same time save the privacy, accuracy and correctness of the retrieved information. This needs a system to be developed that could overcome all these problems while remaining reliable.

1.5 Focus of the Investigation

Current research on access control largely tends toward a theoretical approach (Al-Fedaghi, 2007, Al-Fedaghi, 2006, Agrawal *et al.,* 2002). There are a vast number of digital rights access management, access control and data protection approaches that have been proposed. Nevertheless some of them such as “Chain” method which is suggested in (Al-Fedaghi, 2007) have never been put into real applications, this is because the lack of design and implementation specification in that reference.

There are a vast number of papers presenting varied access control methods. While some of them use healthcare as a motivating example, some are based on empirical studies that support the selection of model properties (i.e. (Komlenovic et al., 2011)) or explain in more detail why the models are suitable for a healthcare setting.

Research on access control may be viewed on a scale from theoretical through implementation to problem focused. Research to date leans toward the former while little has been done on the latter. Motivated by this fact, this PhD project has taken a practical approach to access control in healthcare.

Chapter 4 goes through the details of the research question, hypothesis and methodology. But here a short introduction about the research question is needed. The main objective of this thesis is to develop a reliable method that could overcome all of the outstanding problems in data access management. Therefore the main research question is:

> *“Would a data access management method that is based on semantics overcome the outstanding problems faced by other existing methods?”*

A practical methodology is then set and verified by set of experiments. In order to reach the objectives of this thesis the following process will be followed:

- Analysis of the available methods in the literature by surveys and makes comparison of their structures and how each attacks the problem of data access management;
- After doing the analysis, the advantages and shortcomings of each method have been clearly highlighted;
- Design the new data access method based on findings from the literature analysis;
- Collaboration with experts in the field from different organisations such as: IBM, University of Eindhoven, University of Madrid and University of Trento;
- Implementing the system in three central parts (i.e. ontology, database and semantic layers);
- Integration of the three parts into one cohesive system;
- Evaluation of the experimental results.

The main objectives of our work can therefore be summarised as follows:

- First, an enhanced access control model is defined;

- Second, the defined access control model is integrated with a data handling model and ontology allowing users to define restrictions on the management of their sensitive data used by the receiving parties. For this purpose, the researcher focuses on the development of an architecture implementing a privacy-aware access control system that integrates access control and data handling policies.

In this section, the focus of investigation has been summarised in the above points and the next section will have an overview of the dissertation to explain how the goals of the work have been achieved.

The problem presented in this chapter raises the question of the availability of an approach that has a simple design and can improve the following two criteria in data access management:

- Simplification of the data access management configuration
- Increasing the precision of the retrieved data.

The research question, contribution and hypothesis are presented in detail with the methodology of the thesis in Chapter 4.

To find key tools to implement a reliable data access management, a systematic literature review will be undertaken in the next chapter to provide guidance to researchers, decision-makers and others who are involved in the planning and implementation of integrated e-health systems. The researcher will focus on the different solutions that have been suggested in the literature in order to know how to achieve the above requirements and overcome these problems. An overview of each method will be presented in addition to a comparison of the advantages and disadvantages of each.

1.6 Overview of the Dissertation

The previous sections have drawn the main features that will shape this thesis. The details of the topics highlighted above will be discussed in the rest of the thesis as described below.

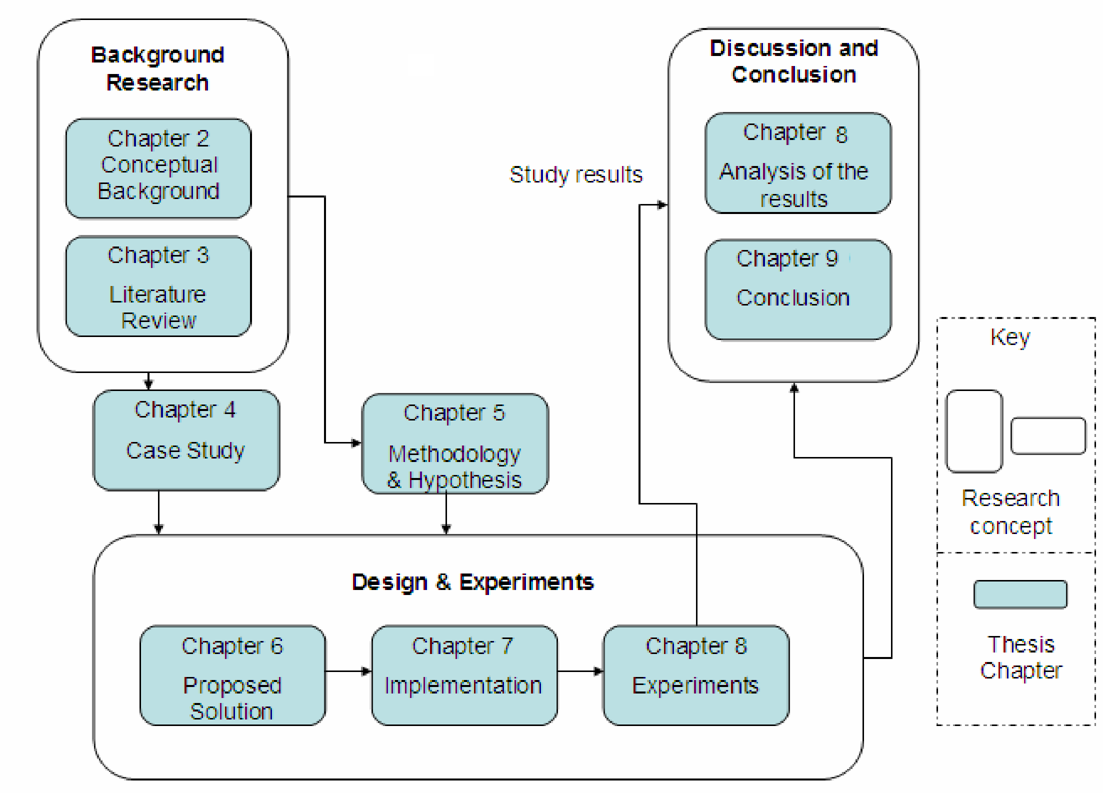

Background Research
Chapter 2 Conceptual Background
Chapter 3 Literature Review
Chapter 4 Case Study
Chapter 5 Methodology & Hypothesis
Study results
Discussion and Conclusion
Chapter 8 Analysis of the results
Chapter 9 Conclusion
Key
Research concept
Thesis Chapter
Design & Experiments
Chapter 6 Proposed Solution
Chapter 7 Implementation
Chapter 8 Experiments

Figure 4: Overview on Thesis Chapters

An overview of each chapter separately is given below:

- **Chapter 2: Literature Review**

  This chapter provides the reader with a comprehensive background and describes the main suggested solutions to privacy violation. Legislation and legal requirements, Access control mechanisms and semantic technologies are all presented, discussed and compared in this chapter.

- **Chapter 3: International Clinic Kuwait Case Study**

  This chapter provides a general background to the case study “International Clinic”, general presentations of Patient and Information Flows in the International Clinic, Record Storage and Access Requirements and Current and future systems and systems needs of the hospital.

- **Chapter 4: Hypothesis and Methodology**

  In this chapter the research question, hypothesis, objectives and proposed methodology are discussed in detail.

- **Chapter 5: Proposed Solution**

  This chapter presents the chain method implementation, the design of the ontology and the overall design of the Chain-ontology based system.

- **Chapter 6: Implementation**

  This chapter discusses the system implementation, and how the integration between the chain and the semantics has taken place.

- **Chapter 7: Experiments and Analysis of the Results**

  Experiments that have taken place by expert database administrators on scientific scenarios are analysed. Results conducted from the experiments on the semantic chain-ontology based system are also discussed.

- **Chapter 8: Conclusions**

  An overall discussion of the thesis objectives and outcomes is provided. In addition, this chapter will include suggestions for future work.

Chapter 2
Literature Review

From the previous chapter it has become clear that there is a crucial problem in data access management. The problem is how to save the privacy of personal information while using a reliable system. And this problem becomes worse when it exists in the healthcare field, where a lot of patients' information is to be entered and processed in one information system.

From the discussion in the previous chapter, the researcher will start to draw an outline of the system that is needed to overcome this problem. The main feature of this system is to have relatively simple design while keeping the capability of retrieving the correct and exact information to the authorised system. This should be done within a suitable time limit in order for the system to be reliable. And all of this should be done under the umbrella of a secure system that could preserve the privacy of the information.

The relevant literature should be examined to find answers to the question raised in Chapter 1: “Does a reliable data access management method exist in the literature?” and in order to answer this question the researcher will identify the working, advantages and disadvantages of each method.

So in order for a researcher to have better vision for the problem of privacy preserving, one needs to look at a long list of privacy preserving attempts. One needs to look first for the legalisations that have been proposed in order to save the privacy of personal information. A good definition of personal information should be set in order for the researcher to know what should be protected. Then the researcher should consider the database and non-database oriented methods. After looking at the database oriented systems and methods, the researcher may seek supporting context for these methods.

So this chapter is organised as follows: first, personal information and privacy specifications are presented with their definitions from different perspectives. Then, the researcher will start presenting the existing privacy languages set to preserve privacy and personal information. Next, the different database approaches are outlined to solve the data privacy violation problem. Finally the semantic approach is presented as a complementary component in any modern database privacy preserving system.

2.1 Personal Information and Privacy specifications

Information of a personal nature can in some instances allow identification of an individual. This includes information such as a person's name, address, financial information, marital status or billing details.

The Privacy and Personal Information Protection (PPIP) Act and Health Records and Information Privacy (HRIP) Act (New South Wales Consolidated Acts, 2012) define 'personal information' as "information or an opinion (including information or an opinion forming part of a database and whether or not recorded in a material form) about an individual whose identity is apparent or can reasonably be ascertained from the information or opinion”.

In the next two subsections the researcher will discuss in detail two methods suggested by the literature to preserve the personal information discussed above. The first discusses privacy specifications and the other discusses a language that describes privacy preferences.

3.1.1 Privacy specifications

As the World Wide Web became a genuine medium in which one can buy products and get services, commercial websites tried to collect more information about the people who purchased their products. Some companies used controversial ways such as tracker cookies to ascertain the users' demographic information and buying habits. This information is used to provide specifically targeted advertisements or what is known also as “Adware”. Users who think this violates their privacy would sometimes turn off HTTP cookies or use proxy servers to keep their personal information secure. P3P is designed to give users a more precise control of the kind of information that they allow to release. According to the World Wide Web Consortium (W3C) the main goal of the Platform for Privacy Preferences Project (P3P) “is to increase user trust and confidence in the Web through technical empowerment”.

P3P (w3C website, 2012) is a machine-readable language that helps to express a website’s data management practices. P3P manages information through privacy policies. When a website uses P3P, the developers set up a set of policies that allows them to state their intended uses of personal information that may be gathered from their site visitors (Ashley *et al.*, 2002). When a user decides to use P3P, they set their own set of policies and state what personal information they will allow to be seen by the sites that they visit. Then when a user visits a site, P3P will compare what personal information the user is willing to release, and what information the server wants to get – if the two are not equivalent, P3P will inform the user and ask whether he/she is willing to proceed to the site, and risk giving up more personal information. As an example, a user may store in the browser preferences that information about their browsing habits should not be collected. If the policy of a Website states that a cookie is used for this purpose, the browser automatically rejects the cookie. The main content of a privacy policy is the following:

The information the server stores:

- Which kind of information is collected (identifying or not);
- Which particular information is collected (IP address, email address, name, etc.);

Use of the collected information:

- How this information is used (for regular navigation, tracking, personalisation, telemarketing, etc.);
- Who will receive this information (only the current company, third party, etc.);

Permanence and visibility:

- How long information is stored;
- Whether and how the user can access the stored information (read-only, option, opt out).

P3P allows browsers to understand their privacy policies in a simplified and organised manner rather than searching throughout the entire website. By setting your own privacy settings at a certain level, P3P will automatically block any cookies that you might not want on your computer. Additionally, the W3C explains that P3P will allow browsers to transfer user data to services, ultimately promoting an online sharing community.

The Electronic Privacy Information Centre (EPIC) has been critical of P3P and believes P3P makes it too difficult for users to protect their privacy (EPIC website, 2011). In 2002, it assessed P3P, and referred to the technology as a "Pretty Poor Policy" (EPIC website, 2011). According to EPIC, some P3P software is too complex and difficult for the average person to understand, and many Internet users are unfamiliar with how to use the default P3P software on their computers or how to install additional P3P software. Another concern is that both websites and Internet users are not obligated to use P3P. P3P has been known to undermine public confidence by collecting enormous amounts of information that can be used against its user. Moreover, the EPIC website claims that P3P's protocol would become burdensome for the browser and not as beneficial or efficient as it was intended to be.

2.1.2 Privacy Languages

IBM was able to see the limitation of P3P. Michael Kaply from IBM is reported saying the following when the Mozilla Foundation was considering the removal of P3P support from their browser-line: "*We (IBM) wrote the original P3P implementation and then Netscape proceeded to write their own. So both our companies wasted immense amounts of time that everyone thought was a crappy proposal to begin with. Remove",* and decided to build a technology that would fill that deficiency, and thus was the Enterprise Privacy Authorisation Language (EPAL) (Ashley *et al.,* 2002) project created in 2002. EPAL is mainly a business-to-business (B2B) technology that helps streamline information flow during business interactions. It helps ensure that information is protected and used in accordance with the responsible organisation's privacy policies. IBM introduced EPAL as a formal language that provides enterprises with a way to automate and enforce privacy policies across IT applications and systems. The language allows organisations to specify their privacy practices in a way that they, and other organisations with which they interact, can read and use. EPAL policies, unlike P3P policies, are enforceable, as they are written and structured in a similar fashion to access control policies that one may find in the security domain. EPAL stores policies, as well as log and audit access to data as a means to document policy enforcement. The policies are enforced by an enforcement engine that parses the files, assuring the information collection, use and storage that occurs within the organisation, and amongst the organisation and its partners, complies with the EPAL specified privacy practices.

EPAL policies contain meta-information that does not exclusively address information access and usage, a property shared also by P3P. The meta-information includes the policy ID and description (as in the example of Figure 5 where the description in Rules 1 and 2), information about the issuing organisation, and modification dates and document revision numbers. Organisations define a vocabulary specific to their needs using EPAL, and the only resulting condition is that every agent that wants to use the policy to govern their interactions must agree upon and understand the vocabulary being used. EPAL rules specify the policies regarding a specific information access.

At first, the EPAL vocabulary defines several elements which can be used in EPAL Policy. As shown in Figure 5, it has sets of user category, data category, purpose, action, and obligation. It serves as the definition of internal privacy policy. EPAL Policy would be setup according to some specific EPAL Vocabulary with additional information.

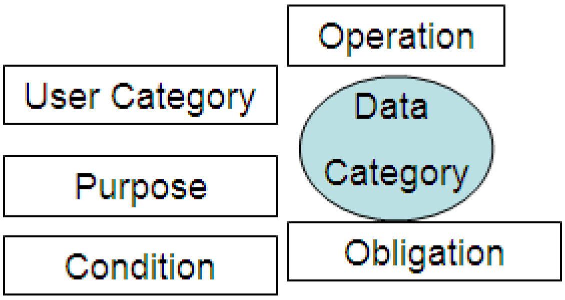


Figure 5: Example on the work of EPAL

Let’s take an example of a doctor who wants to write a medical note for his patient during the appointment. In this case the Privacy statement would be: “Medical Note can be used for writing the doctor’s medical note if the patient has an appointment with him”
If translated into EPAL it would be:

“

EPAL RULE <ALLOW
User-category = “Medical Staff”
Data-category=“Medical Note”
Purpose=“Writing the medical note”
Operation=“write”
Condition=“/Appointment=True&&the patient has appointment with the doctor”>
“

From the above example, many requirements are needed to be carefully written in order to write a privacy policy in EPAL such as user category, data category, purpose, operation and condition. Therefore a special expert database administrator is needed in order to create a data access management system that uses EPAL to write its privacy policy. Such a database administrator cannot be easily found especially in the developing countries such as Kuwait. In addition hiring them would add costs to the system costs, and this wouldn't be very appealing to the project owners.

On the technical side, it will be clear later on in this chapter that there are other methods such as the Chain method combine many of the above requirements such as the purpose and the condition in one goal. This makes the implementation much easier and cost effective.

Accordingly, EPAL can't stand as a proposed solution according to the requirements presented in this section because it doesn't have a simple design nor is it reliable for large sensitive systems.

2.2 Database Oriented Solutions

Before going into available database oriented solutions, it is necessary to go through the theoretical definition of Access Management and the basic design of it that has been developed by Lampson. And then the researcher will present the most significant Access Management methods such as DAC, MAC, RBAC, XACML and the Hippocratic database.

Chong (2004) defines Identity and Access Management (I&AM) as follows:

"Identity and access management refers to the processes, technologies and policies for managing digital identities and controlling how identities can be used to access resources."

From the above definition it can be noted that:

- I&AM is not just about technology, but rather, is comprised of three indispensable elements: policies, processes and technologies. Policies refer to the constraints and standards that need to be followed in order to comply with regulations and business best practices; processes describe the sequences of steps that lead to the completion of business tasks or functions; technologies are the automated tools that help accomplish business goals more efficiently and accurately while meeting the constraints and guidelines specified in the policies.

- The relationships between elements of I&AM can be represented by the triangle illustrated in Figure 6. Of significant interest is the fact that there is a feedback loop that links all three elements together. The lengths of the edges represent the proportions of the elements relative to one another in a given I&AM system. Varying the proportion of one element will ultimately vary the proportion of one or more other elements in order to maintain the shape of a triangle with a sweet spot (shown as an intersection in the triangle).

- The triangle analogy is perfect for describing the relationships and interactions of policies, processes and technologies in a healthy I&AM system as well. Every organisation is different and the right mix of technologies, policies and processes for one company may not necessarily be the right balance for a different company. Therefore, each organisation needs to find its own balance represented by the uniqueness of its triangle.

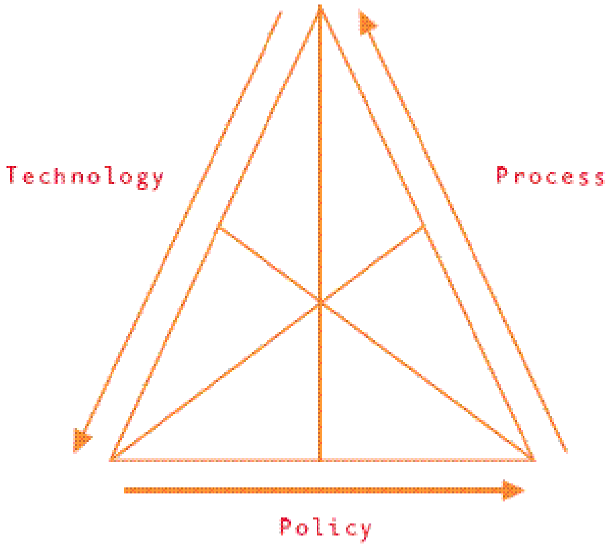


Figure 6: Essential elements of an identity and access management system (Chong 2004).

All the information represented in the previous section is usually stored in databases. Databases traditionally have access control mechanisms associated with them. So in order to discuss any of the data base oriented solutions, it should be emphasised that all of them should have the essential elements of the triangle above. And in order to know how each subject and object acts on the process and polices of the previous triangle, the Lampson's model should be explained.

In Computer Science, an Access Control Matrix or Access Matrix is an abstract, formal security model of protection state in computer systems that characterise the rights of each subject with respect to every object in the system. It was first introduced by (Lampson, 1971).

In his model, Lampson defines security to be the prevention and detection of unauthorised actions on information. He splits them into two important cases:

- An attacker has access to the raw bits representing the information;

- In this case cryptographic techniques are needed;
- There is a software layer between the attacker and the information and thus there is a need for access control techniques.

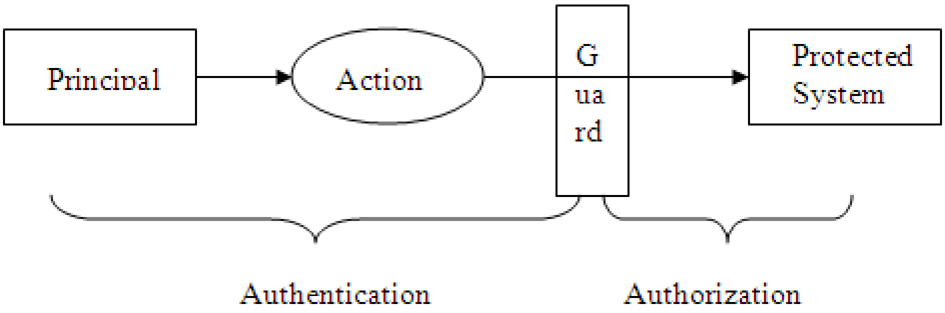


Figure 7: General Access Control Model (Lampson's model (Lampson and Butler, 1971)).

As shown in the figure above:

- Actions are written as procedures
- Behaviour of the guard is specified by:
  - Declaration of state variables
  - Implementations of the action procedures

In short all the database oriented solutions below would be based on the above principles of Lampson's model. All of them would have two areas: Authentication and Authorisation. Inside the Authentication area they would have the principal and action, but the content of the principal and action would differ according to the design of different methods.

In the next section the researcher will start to look at the different Database Oriented Solutions available in the literature, and highlight their infrastructure and their advantages and disadvantages in solving the problem at hand.

2.3 Discretionary Access Control (DAC)

Discretionary Access Control (DAC) is a type of access control in which users have complete control over all the programs they own and execute, and also determines the permissions other users have over those files and programs. Because DAC requires permissions to be assigned to those who need access, DAC is commonly called described as a "need-to-know" access model.

ACLs and owner/group/other access control mechanisms are by far the most common mechanisms for implementing DAC policies. Other mechanisms, even though not designed with DAC in mind, may have the capabilities to implement a DAC policy. This represents a problem for practical access control systems, because there are numerous access control policies that have aspects of discretionary access control, but are not purely discretionary (Osborn and Hulme, 2000).

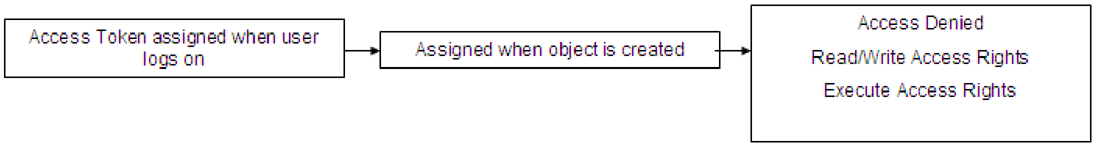

Access Token assigned when user logs on
Assigned when object is created
Access Denied
Read/Write Access Rights
Execute Access Rights

Figure 8: Infrastructure of DAC.

As shown in the figure above, DAC leaves a certain amount of access control to the discretion of the object's owner or anyone else that is authorised to control the object's access. For example, it is generally used to limit a user's access to a file; it is the owner of the file who controls other users' access to the file. Only those users specified by the owner may have some combination of read, write, execute, and other permissions to the file. DAC policy tends to be very flexible and is widely used in the commercial and government sectors. However, DAC is known to be inherently weak for two reasons. First, granting read access is transitive; for example, when Ann grants Bob read access to a file, nothing stops Bob from copying the contents of Ann's file to an object that Bob controls. Bob may now grant any other user access to the copy of Ann's file without Ann's knowledge. Second, DAC policy is vulnerable to Trojan horse attacks. Because programs inherit the identity of the invoking user, Bob may, for example, write a program for Ann that, on the surface, performs some useful function, while at the same time destroys the contents of Ann's files. When investigating the problem, the audit files would indicate that Ann destroyed her own files. Thus, formally, the drawbacks of DAC are as follows:

- Information can be copied from one object to another; therefore, there is no real assurance on the flow of information in a system.
- No restrictions apply to the usage of information when the user has received it.
- The privileges for accessing objects are decided by the owner of the object, rather than through a system-wide policy that reflects the organisation's security requirements.

In addition, if there is no restriction to the usage of information once the user receives it. This could cause horrible consequences, which means that nurses for example could re-write the dosage of the injections given to a certain patient because once she receives this information she could perform any action on it. And the last drawback is self-criticising in the case of a hospital.

From the above, it can be seen that DAC does give us a good starting point for policies in data access management but it still can't be used to solve the problem presented in previous chapters.

2.4 Non-Discretionary Access Control

In computer security, discretionary access control (DAC) is a type of access control defined by the Trusted Computer System Evaluation Criteria (TSEC) (United States Department of Defence, 1985) "as a means of restricting access to objects based on the identity of subjects and/or groups to which they belong. The controls are discretionary in the sense that a subject with certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject (unless restrained by mandatory access control)".

In general, all access control policies other than DAC are grouped in the category of nondiscretionary access control (NDAC). As the name implies, policies in this category, unlike DAC, have rules that are not established at the discretion of the user. Non-discretionary policies establish controls that cannot be changed by users, but only through administrative action. So if one turns back to Figure 8, there would be an additional layer of administration between the user and the rights. Separation of duty (SOD) policy can be used to enforce constraints on the assignment of users to roles or tasks. An example of such a static constraint is the requirement that two roles be mutually exclusive; if one role requests expenditures and another approves them, the organisation may prohibit the same user from being assigned to both roles. So, membership in one role may prevent the user from being a member of one or more other roles, depending on the SOD rules, such as Work Flow and Role-Based Access Control. Another example is a history-based SOD policy that regulates, for example, whether the same subject (role) can access the same object a certain number of times. A typical example of NDAC is Mandatory Access Control (MAC). An example of MAC occurs in military security, where an individual data owner does not decide who has a Top-Secret clearance, nor can the owner change the classification of an object from Top Secret to Secret. MAC is the most mentioned NDAC policy (Loscocco *et al.*, 1998).

MAC takes a hierarchical approach to controlling access to resources. Under a MAC enforced environment access to all resource objects (such as data files) is controlled by settings defined by the system administrator. As such, all access to resource objects is strictly controlled by the operating system based on system administrator configured settings. It is not possible under MAC enforcement for users to change the access control of a resource.

MAC begins with *security labels* assigned to all resource objects on the system. These security labels contain two pieces of information - a classification (top secret, confidential, etc.) and a category (which is essentially an indication of the management level, department or project to which the object is available).

Similarly, each user account on the system also has classification and category properties from the same set of properties applied to the resource objects. When a user attempts to access a resource under MAC, the operating system checks the user's classification and categories and compares them to the properties of the object's security label. If the user's credentials match the MAC security label properties of the object access is allowed. It is important to note that *both* the classification and categories must match. A user with top secret classification, for example, cannot access a resource if they are not also a member of one of the required categories for that object.

MAC is by far the most secure access control environment but does not come without a price. Firstly, MAC requires a considerable amount of planning before it can be effectively implemented. Once implemented it also imposes a high system management overhead due to the need to constantly update object and account labels to accommodate new data, new users and changes in the categorisation and classification of existing users.

The need for a MAC mechanism arises when the security policy of a system dictates that:

1. Protection decisions must not be decided by the object owner.
2. The system must enforce the protection

Multilevel security models such as the (Bell-La Padula Confidentiality) and Biba integrity models are used to formally specify this kind of MAC policy. However, information can pass through a covert channel in MAC, where information of a higher security class is deduced by inference such as assembling and intelligently combining information of a lower security class.

These policies for access control are not particularly well suited to the requirements of government and industry organisations that process unclassified but sensitive information. In these environments, security objectives often support higher-level organisational policies which are derived from existing laws, ethics, regulations, or generally accepted practices. Such environments usually require the ability to control actions of individuals beyond just an individual's ability to access information according to how that information is labelled based on its sensitivity.

Most organisations nowadays do not use MAC in their applications, as it is hard to design and apply to real systems. They usually use another type of NDAC which is Role based Access Control that will be explained in detail in the next section.

2.5 Role-Based Access Control (RBAC)

Access is the ability to do something with a computer resource (e.g., use, change, or view). Access control is the means by which the ability is explicitly enabled or restricted in some way (usually through physical and system-based controls). Computer based access controls can prescribe not only who or what process may have access to a specific system resource, but also the type of access that is permitted. These controls may be implemented in the computer system or in external devices.

The concept RBAC has been used with multi-user computer systems and multi-application online systems since the late 1960s and early 1970s. However, RBAC has rapidly emerged in the 1990s as a promising technology for managing and enforcing security in large-scale enterprise-wide systems, largely because of the lack of enhancement in the traditional Mandatory Access Control (MAC) and Discretionary Access Control (DAC) used in many computer systems and networks. Thus, RBAC is an attractive alternative to traditional MAC and DAC policies.

With role-based access control, access decisions are based on the roles that individual users have as part of an organisation. Users take on assigned roles (such as doctor, nurse, teller, manager). The process of defining roles should be based on a thorough analysis of how an organisation operates and should include input from a wide spectrum of users in an organisation.

Access rights are grouped by role name, and the use of resources is restricted to individuals authorised to assume the associated role. For example, within a hospital system the role of doctor can include operations to perform diagnosis, prescribe medication, and order laboratory tests; and the role of researcher can be limited to gathering anonymous clinical information for studies.

The use of roles to control access can be an effective means for developing and enforcing enterprise-specific security policies, and for streamlining the security management process.

Below, the features and the infrastructure of RBAC are explained.

2.5.1 Users and Roles

As previously explained, in RBAC, access decisions are based on the roles that individual users have as part of an organisation.

When a user is associated with a role, the user can be given no more privilege than is necessary to perform the job. This concept of least privilege requires identifying the user's job functions, determining the minimum set of privileges required to perform that function, and restricting the user to a domain with those privileges and nothing more. In less precisely controlled systems, this is often difficult or costly to achieve. Someone assigned to a job category may be allowed more privileges than needed because it is difficult to tailor access based on various attributes or constraints. Since many of the responsibilities overlap between job categories, maximum privilege for each job category could cause unlawful access.

Under RBAC, roles can have overlapping responsibilities and privileges; that is, users belonging to different roles may need to perform common operations. Some general operations may be performed by all employees. In this situation, it would be inefficient and administratively cumbersome to specify repeatedly these general operations for each role that gets created. Role hierarchies can be established to provide for the natural structure of an enterprise. A role hierarchy defines roles that have unique attributes and that may contain other roles; that is, one role may implicitly include the operations that are associated with another role.

In the healthcare situation, a role “Specialist” could contain the roles of” Doctor” and “Intern”. This means that members of the role Specialist are implicitly associated with the operations associated with the roles” Doctor” and “Intern” without the “administrator” having to explicitly list the “Doctor” and “Intern” operations. Moreover, the roles “Cardiologist” and “Rheumatologist” could each contain the Specialist role.

Role hierarchies are a natural way of organising roles to reflect authority, responsibility, and competency: the role in which the user is gaining membership is not mutually exclusive with another role for which the user already possesses membership. These operations and roles can be subject to organisational policies or constraints. When operations overlap, hierarchies of roles can be established. Instead of instituting costly auditing to monitor access, organisations can put constraints on access through RBAC. For example, it may seem sufficient to allow physicians to have access to all patient data records if their access is monitored carefully. With RBAC, constraints can be placed on physician access so that only those records that are associated with a particular physician can be accessed.

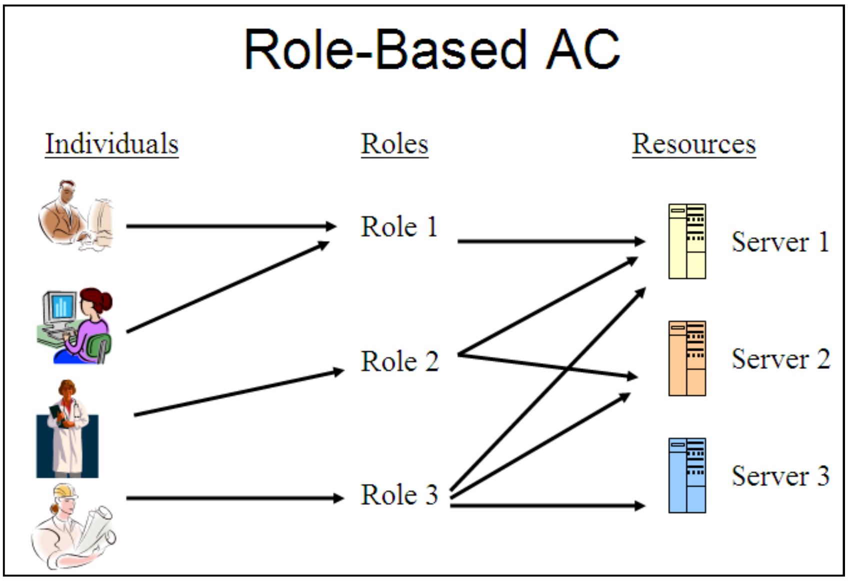


Figure 9: RBAC Model.

Figure 9 shows the mechanism of RBAC which depends mainly on the role of the user as discussed earlier.

2.5.2 Roles and Operations

Organisations can establish the rules for the association of operations with roles. For example, a healthcare provider may decide that the role of clinician must be constrained to post only the results of certain tests but not to distribute them where routing and human errors could violate a patient's right to privacy. Operations can also be specified in a manner that can be used in the demonstration and enforcement of laws or regulations. For example, a pharmacist can be provided with operations to dispense, but not to prescribe a medication.

An operation represents a unit of control that can be referenced by an individual role, subject to regulatory constraints within the RBAC framework. An operation can be used to capture complex security-relevant details or constraints that cannot be determined by a simple mode of access.

For example, there are differences between the access needs of a teller and an accounting supervisor in a bank. An enterprise defines a teller role as being able to perform a savings deposit operation. This requires read and write access to specific fields within a savings file. An enterprise may also define an accounting supervisor role that is allowed to perform correction operations. These operations require read and write access to the same fields of a savings file as the teller. However, the accounting supervisor may not be allowed to initiate deposits or withdrawals but only perform corrections after the fact. Likewise, the teller is not allowed to perform any corrections once the transaction has been completed. The difference between these two roles is the operations that are executed by the different roles and the values that are written to the transaction log file.

The RBAC framework provides administrators with the capability to regulate who can perform what actions, when, from where, in what order, and in some cases under what relational circumstances.

Only those operations that need to be performed by members of a role are granted to the role. Granting of user membership to roles can be limited. Some roles can only be occupied by a certain number of employees at any given period of time. The role of manager, for example, can be granted to only one employee at a time. Although an employee other than the manager may act in that role, only one person may assume the responsibilities of a manager at any given time. A user can become a new member of a role as long as the number of members allowed for the role is not exceeded.

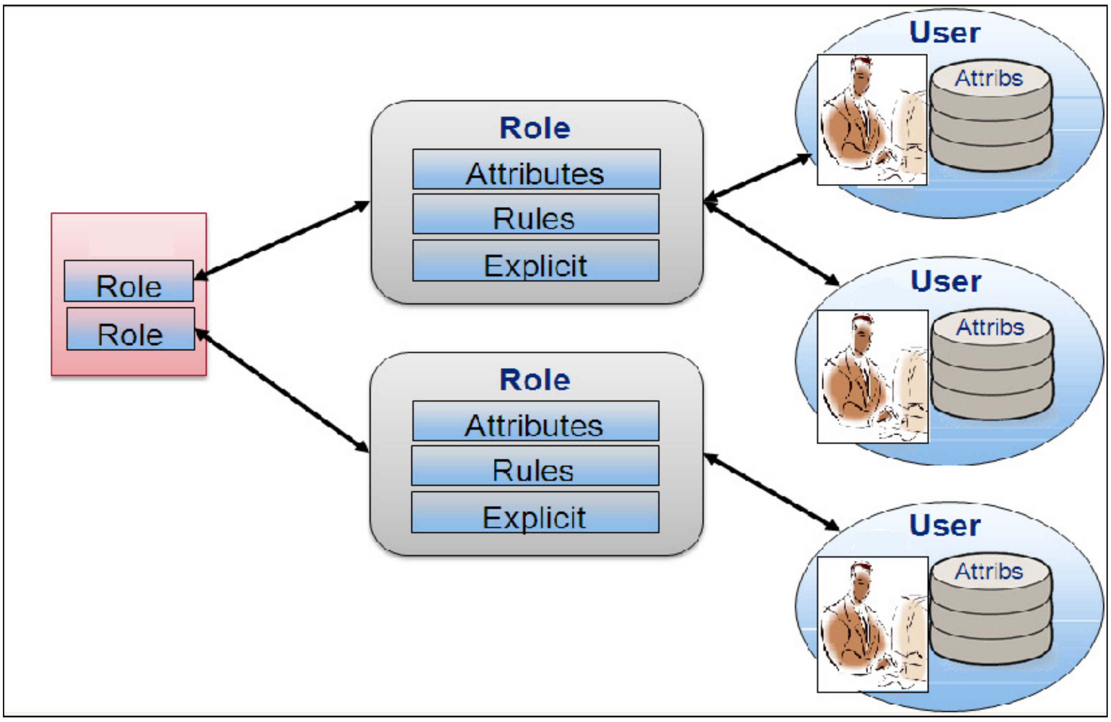


Figure 10: RBAC Roles and Users.

2.5.3 Advantages of RBAC

A properly administered RBAC system enables users to carry out a broad range of authorised operations, and provides flexibility and breadth of application. System administrators can control access at a level of abstraction that is natural to the way that enterprises typically conduct business. This is achieved by statically and dynamically regulating users' actions through the establishment and definition of roles, role hierarchies, relationships, and constraints. Thus, once an RBAC framework is established for an organisation, the principal administrative actions are the granting and revoking of users into and out of roles. This is in contrast to the more conventional and less intuitive process of attempting to administer lower-level access control mechanisms directly (e.g., access control lists [ACLs], capabilities, or type enforcement entities) on an object-by-object basis.

Further, it is possible to associate the concept of an RBAC operation with the concept of "method" in Object Technology. This association leads to approaches where Object Technology can be used in applications and operating systems to implement an RBAC operation.

For distributed systems, RBAC administrator responsibilities can be divided among central and local protection domains; that is, central protection policies can be defined at an enterprise level while leaving protection issues that are of local concern at the organisational unit level. For example, within a distributed healthcare system, operations that are associated with healthcare providers may be centrally specified and pertain to all hospitals and clinics, but the granting and revoking of memberships into specific roles may be specified by administrators at local sites.

2.5.4 RBAC versions and Status of Current RBAC Activities

RBAC is a technical means for controlling access to computer resources. While still largely in the demonstration and prototype stages of development, RBAC appears to be a promising method for controlling what information computer users can utilise the programs that they can run, and the modifications that they can make. Only a few off-the-shelf systems that implement RBAC are commercially available; however, organisations may want to start investigating RBAC for future application in their multi-user systems. RBAC is appropriate for consideration in systems that process unclassified but sensitive information, as well as those that process classified information.

Several organisations are experimenting with the inclusion of provisions for RBAC in open consensus specifications. RBAC is an integral part of the security models for Secure European System for Applications in a Multi-vendor Environment (SESAME) distributed system and the database language SQL3.

RBAC has performed well as data access management method that fit for different privacy policies in different organisations. Nevertheless, some RBAC models have been considered to be inefficient for several reasons:

First, differentiating roles in different contexts often proved to be difficult. This has resulted in large quantities of role definitions in some cases producing more roles than users.

Second, RBAC remains somewhat coarse grained while modern requirements are increasingly fine grained.

Finally, while the initial RBAC model was based on permissions only, the need to explicitly specify denial of access became unavoidable.

So in short the drawbacks of RBAC are:

- The method is more categorical as you can specify some roles and stick to them but it's not context sensitive.
- The lack of knowledge and staff expertise in the area of RBAC increases the uncertainty of technical feasibility and developing successfully.
- This would increase time, effort and funding needed for implementation and design.

These factors have resulted in multiple variations of the RBAC model; Administration RBAC (ARBAC) involves control over components such as roles, users, and permissions (Sandhu et al., 1998).

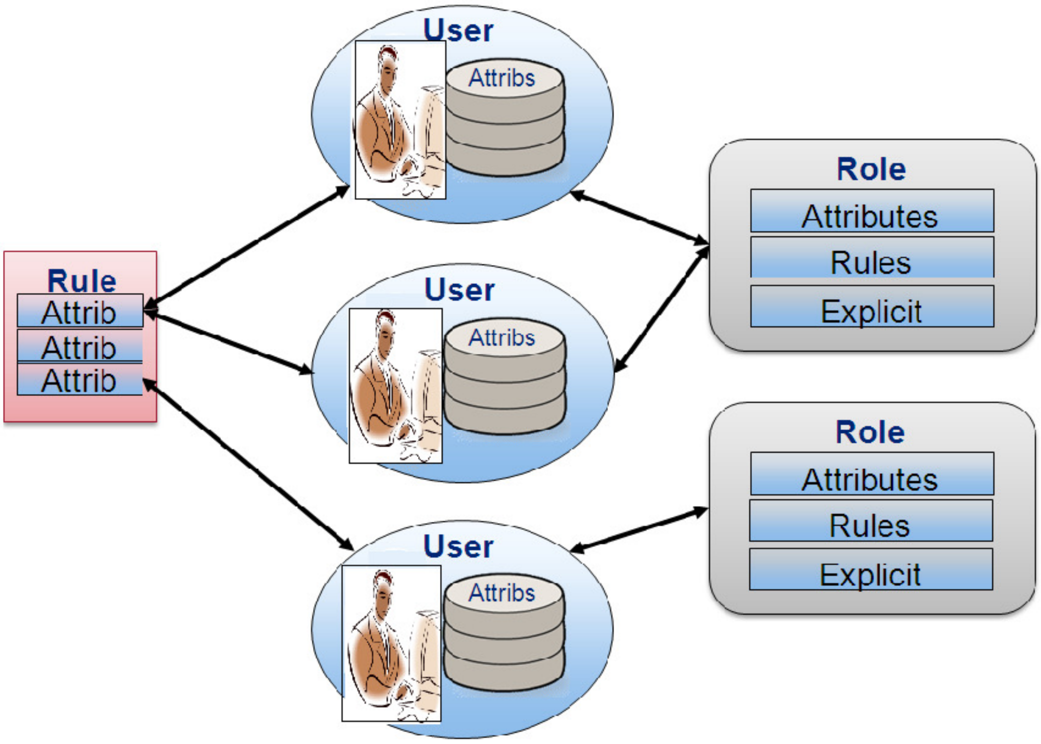


Figure 11: ARBAC Roles and Users.

In the ARBAC, the roles are not directly involved with access control rules – except perhaps that they may show up as an attribute of the user and be used in the rules' truth evaluations. However the roles are very useful in the administration of massive sets of users. They are also very useful in the attestation, auditing and other security and identity processes around entitlement management.

The RBAC versions include creations and deletion of roles, creation and deletion of permissions, assignment of permissions to roles and their removal, creation and deletion of users, assignment of use to roles and their removal. Moreover, it also includes definition and maintenance of the role hierarchy, definition and maintenance of constraints; all of these in turn are for administrative roles and permissions. It has three components or sub-models called user-role assignment (URA97), permission-role assignment (PRA97) and role-role assignment (RRA97) (Sandhu *et al.*, 1998), (Sandhu *et al.*, 1997).

Moffet (1998) expanded the NIST RBAC model to make it more suitable for complex systems. He presented an RBAC model, with three types of hierarchies; is-a, activity, and supervision. For example, in the healthcare system, one can create a role called healthcare provider who has all the responsibilities common to nurses, physicians, and lab technicians. By giving a set of permissions to healthcare providers, the nurses, physicians, and lab technicians also inherit the same set of permissions. Moffett called this type of hierarchy is-a; and the relationship can be read as: a physician/nurse/lab technician is-a healthcare provider. Activity hierarchy connects the roles that are needed to perform a task. For example, only a physician who is responsible for a patient can give a prescription to him/her. The supervision hierarchy connects senior roles to junior ones, for instance, nurse to a head nurse.

Covington (Covington *et al*., 2001) included another type of permission to the RBAC that is based on the environment. This type of permission is not needed in systems such as healthcare where the healthcare providers have access to patients' medical information anywhere and anytime there is a need for it. Crook (2003) defined roles and categorised them as follows: functional role, seniority role, and contextual role. Access is defined as a relation among users, roles, operations, and assets. If a user has certain role(s), he can do specific operations on one or more assets. A contextual role is connected to a context type where it is connected to an asset (Crook *et al*., 2003). Fig. 12 shows an example where a doctor (role) has read and write (operation) access to a patient's medical record (asset), provided that the doctor is responsible (role) for that patient (context).

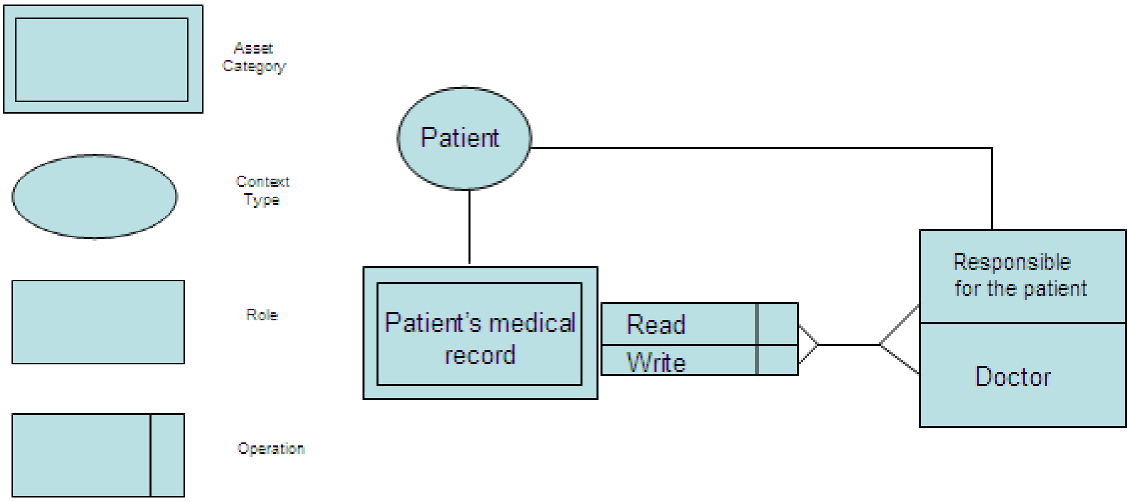


Figure 12: The doctor who is responsible for the patient has read and write access to a patient's medical record.

Using Crook's RBAC technique, one has to consider that the purpose of an individual in requesting access is not included in Crook's role based access control. Users may need to have access to variant information for variant purposes. As a result, the type of access they get should change depending on their purpose. For example, a doctor may need access to a patient's information. His purpose can be to give a prescription to the patient or to complete the patient's profile. In the first case, the system can give read access to the physician. However, in the second case, the physician should also be able to add or change the patient’s profile as well. Therefore, access control should be able to give different types of accesses in the two cases. Crook’s models are not able to take the purpose of the data into considerations

2.6 Task Based Authorisation Control TBAC

Task Based Access Control (TBAC) is well suited for distributed computing and information processing activities with multiple points of access, control, and decision making such as those found in workflow and distributed processes and transaction management systems. TBAC differs from traditional access controls and security models in many respects (Thomas and Sandhu, 1997). Instead of having a system-centric view of security, TBAC approaches security modelling and enforcement at the application and enterprise level, which makes it more desirable in real world enterprises.

In 2000, Sandhu *et al*. pioneered NIST (National Institute of Standards and Technology) RBAC models (W3 website, 2001). Like other RBAC models, permissions are given to the roles rather than the users, where roles are defined to be mutually exclusive. Sandhu *et al* introduced two types of hierarchies: a role hierarchy and an activity hierarchy. In the role hierarchy, senior roles inherit all permissions of junior roles, whereas in the activity hierarchy senior roles inherit only partial permissions of junior roles.

There are some non-canonical (or non-"standard") access control models (besides the well-known MAC, DAC and RBAC) that are simply not well defined. Anyone can define or redefine them as they want, as long as the model makes sense. In most cases TBAC is aggregated back up into roles.

That is, the access is granted based on a task but the access check then compares this task to roles that contain that task, and users that are part of one of those roles. In other words, tasks can be seen as "sub-roles" - or if it is easier to understand, roles become role-containers, and the tasks are the real roles.

Clearly, this is a *huge* improvement on straight RBAC, since it gives you some granularity and dynamics to play with, but it is still a form of (extended) RBAC. It allows many dynamic information processing activities with multiple points of access, control, and decision making. An active security system takes into account the impact of context as it emerges with progressing tasks and distinguishes task-based and context-based permission activation from permission assignment.

TBAC differs from traditional access controls and security models in many respects. Instead of having a system-centric view of security, TBAC approaches security modelling and enforcement at the application and enterprise level. Secondly TBAC lays the foundation for a new breed of what is called “active” security models. By active security models, the researcher means models that approach security modelling and enforcement from the perspective of activities or tasks, and as such, provide the abstractions and mechanisms for the active runtime management of security as tasks progress to completion. In an active approach to security management, permissions are constantly monitored and activated and deactivated in accordance with emerging context associated with progressing tasks (such as in workflows). Such a task-based approach to security represents a radical departure from classical passive security models such as those based on one or more variations of the subject-object view of security and access control. In a subject object view of security, a subject is given access to objects in a system based on some permission (rights) the subject possesses. However, such a subject-object view typically divorces access mediation from the larger context (such as the current state of tasks) in which a subject performs an operation on an object. The most obvious application of TBAC is in secure workflow management.

TBAC enables the granting, usage tracking, and revoking of permissions to be automated and co-ordinated with the progression of the various tasks. Without active authorisation management, permissions will in most cases be “turned on” too early or too late and will probably remain “on” long after the workflow tasks have terminated. This opens up vulnerabilities in systems. Any attempt to minimise such vulnerabilities will require a security administrator to keep track of the progress of the tasks for all enacted workflow instances; an error prone and impossible task! Thus what is needed is an approach where access control permissions are granted and revoked according to the validity of authorisations and one where this can be done without manual security administration. The authorisations themselves are of course processed strictly according to some application logic and policy. In the remaining sections of this project the researcher will describe how TBAC ideas can be used to accomplish this.

TBAC focuses on security modelling and enforcement at the application and enterprise level. In the TBAC paradigm of access control, permissions are associated with contextual information about on-going activities when evaluating an access request. Permissions are checked-in and checked-out from protection states in a just-in-time fashion based on activities or tasks and synchronised with the processing of authorisations in progressing tasks. Thus, TBAC dynamically manages permissions as authorisations by progress to completion and minimises the vulnerabilities in a system.

There are basically two broad objectives guiding the research efforts in TBAC. The first is to model from an enterprise perspective, various authorisation policies that are relevant to organisational tasks and workflows, and a set of user friendly envisioned tools to help a security officer model and specify policies. The second objective is to seek ways in which these modelled policies can be automatically enforced at runtime when the corresponding tasks are invoked. Preliminary ideas for TBAC that recognised the need for active security were presented in (Thomas and Sandhu 1997). More recently, a workflow authorisation model (WAM) was presented in (Qin and Atluri 2003). WAM has the same general motivation as TBAC in that it tries to provide some notions of active security and just-in-time permissions.

TBAC keeps track of the usage and consumption of permissions, thereby preventing the abuse of permissions through unnecessary and malicious operations. However, in TBAC as in all the modified Access Control family, there are no contexts in relation to activities, tasks, or workflow progress and it only keeps track of usage and validity of permissions. This is insufficient for collaborative systems that require a much broader definition of context. More fine grained components need to be defined to support dynamic environments motivated by TBAC. TBAC can be used effectively in an application or enterprise, but for most collaborative environments, TBAC is used within other access control models.

In conclusion, TBAC would have the same problem that RBAC has in complex systems such as large healthcare systems. In that the privacy extension will add to the complexity of the system. And the context issue needs to be well addressed in TBAC before it could be applied in real application. And in (Omran *et al.* 2010 and Omran *et al.* 2012), the researchers have presented a study that has real numbers for comparing the required number of tables, SQL statements and policies to show how the complexity of the design using RBAC and TBAC as compared with our method.

2.7 Comparisons between DAC, MAC, RBAC and TBAC

So, although the Access Control family presented above provides a reasonable solution to the access management problem, none of them has answered the question from Chapter 1 which is the lack of “Context sensitivity”. As all of them are static and can’t be flexible with different situations. This doesn’t give reliability in real case applications such as a hospital, where you have different upcoming cases and some with emergency demands.

In conclusion, it can be stated that earlier RBAC models have some limitations; for instance, they do not include purpose and therefore, cannot distinguish between scenarios where the healthcare providers need information for different purposes. This problem has been addressed in RBAC models with privacy extension. However even those with a privacy extension still lack simplicity of the design as the privacy obligation adds to the complexity of the RBAC design and the work still needs developments. As they (Thomas and Sandhu, 1997) have admitted in their paper that they have not included obligation and retention and/ or a more complete privacy requirements model. Below is a table that compares DAC, MAC, RBAC and TBAC according to three main criteria:

- Access right Permission given by database administrator.
- Context sensitivity.
- Need for expert database administrator.

These three criteria are vital for any system to be durable in preserving the privacy. In addition, the context sensitivity adds flexibility to the system in order to face the changing situations. The need for an expert database administrator, however, adds to the cost of any project.

Method	**DAC**	**MAC**	**RBAC**	**TBAC**
Access right Permission given by	Users	Administrator	Administrator	Administrator
Context sensitivity	Not aware	Not aware	Not aware	Not aware
Need for expert database administrator	No need	In need	In need	In need

Table 1: Comparison between DAC, MAC, RBAC and TBAC

In the next section, an overview of a new method for data access management that works at the data level will be discussed. This thesis provides a new model for access management, ChBAC, and a comparison of these methods with ChBAC is provided in section 7.5

2.8 Hippocratic Database

In a paper titled “Hippocratic Databases", (Agrawal *et al*. 2002) outlined the concept of integrating the right to privacy within database management systems. Their proposed database system was inspired by the medical Hippocratic Oath, hence the term “Hippocratic Database". A founding tenet of a Hippocratic Database system is that it should be responsible for the privacy of the data it manages. The ten principles of Hippocratic database systems have been defined as follows (Zurich IBM website):

1. **Purpose specification:**

For personal information stored in the database, the purpose for which the information has been collected should be associated with that information.

2. **Consent:**

The purpose associated with personal information should have the consent of the donor of the personal information.

3. **Limited collection:**

The personal information collected should be limited to the minimum necessary for accomplishing the specified purposes.

4. **Limited use:**

The database should run only those queries that are consistent with the purposes for which the information has been collected.

5. **Limited disclosure:**

Personal information stored in the database should not be communicated outside the database for purposes other than those for which there is consent from the donor of the information.

6. **Limited retention:**

Personal information should be retained only as long as necessary for the fulfilment of the purposes for which it has been collected.

7. **Accuracy:**

Personal information stored in the database should be accurate and up-to-date.

8. **Safety:**

Personal information should be protected by security safeguards against theft and other misappropriations.

9. **Openness:**

A donor should be able to access all information about him or her stored in the database.

10. Compliance:

A donor should be able to verify compliance with the above principles. Similarly, the database should be able to address a challenge concerning compliance.

The initial concept of the Hippocratic database might well have been inspired by the Hippocratic Oath, but the outlined principles are, also, deeply rooted in the idea of “Fair Information Practices". These practices are themselves based on the privacy principles outlined by Organisation for Economic Co-operation and Development (OECD) in 1980. The Hippocratic designers further outlined a straw man design along with a set of use cases against which Hippocratic databases could be tested.

2.8.1 The Hippocratic Database Architecture

In a summary of current database systems, (Agrawal *et al.*, 2002) considers two properties fundamental for a database system: the ability to manage persistent data; and the ability to access large amounts of data efficiently. In addition to these two properties they further postulate that certain capabilities are universal to database management systems:

High level language support for data structure definition, data access and data manipulation, concurrency control in the form of transaction management, controls to ensure authorised data access and data validity and also a means to recover from system failure with minimal loss of existing data.

In defining the concept of Hippocratic Databases, the designers were very clear on two points. Firstly, a Hippocratic database will need all of the capabilities available in current database systems. Secondly, in the interests of privacy preservation, efficiency, while still important, may not be the central focus. Instead, ensuring that data is used for the purpose for which it was collected will be the overriding concern. The straw man architecture outlined by the Hippocratic database designers, serves not as a blueprint, but rather as a road-map for future development on the path to the realisation of a fully functional Hippocratic database system.

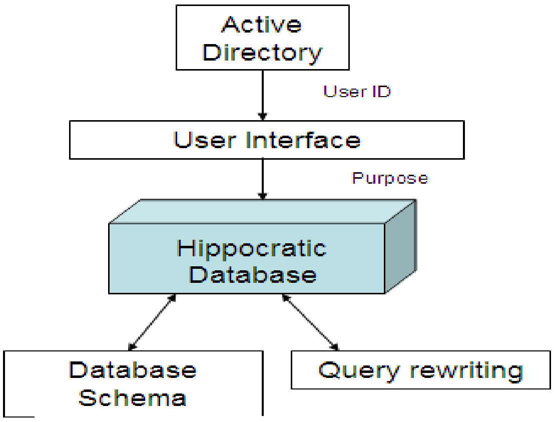


Figure 13: Hippocratic Database model.

Thus the main components of the Hippocratic database architecture are:

1- Privacy Metadata

Privacy metadata tables are the means by which the purposes of data collection are defined. Each piece of collected information must be associated with the purpose(s) for which it is collected. Additionally, the following needs to be described and defined by the metadata:

- The external-recipients: with whom may this information be shared, the retention-period: the duration of time that the collected information is to be stored, and
- The authorised-users: the set of users and/or applications that may access the information.

Creating the metadata tables can be made easier by the privacy metadata creator. Its task would be to automatically generate the required metadata tables using the organisation's privacy policy as its data source.

2- Data Collection

Prior to a user releasing information, the Privacy Constraint Validator will verify that the organisation's privacy policy is in line with the user's privacy preferences. An audit trail of a user's acceptance of the privacy policy must be maintained to address any future challenges regarding compliance. Once the user's acceptance has been obtained, data can be inserted into the database. Along with each stored attribute, the purposes to which the user has agreed to must also be stored. In order to address the principle of accuracy, the Data Accuracy Analyser should perform data accuracy checks. This may take place prior to or after data insertion.

3- Queries

An audit trail of all queries must be maintained to address compliance challenges, as well as to enable external privacy audits. There are essentially three phases that take place in the fulfilment of a Hippocratic database query

4- Retention

The Data Retention Manager is responsible for deleting all information whose retention period has expired.

5- Additional Features

The Data Collection Analyser will examine all queries for all purposes to determine any data collected but not used. Thus other words ensuring adherence to the principle of limited collection.

It also collects any data held for longer than required, thus supporting the principle of limited retention whether persons have unneeded authorisations for queries with a given purpose. This will play a vital role in ensuring the principles of limited use and limited disclosure.

2.8.2 Challenges to Hippocratic Databases

During the course of designing the straw man architecture the designers identified some problems and challenges regarding the 10 principles mentioned above that the Hippocratic database is based on. This subsection presents a summary of their findings:

1- A Policy and Preference Language

The specification of policies lies at the very heart of Hippocratic databases. The Hippocratic database designers believe that P3P form a solid base for the expression of privacy policies. However, since P3P was geared towards the Web and Web shopping, they recommend building on the work of P3P to provide greater support for the richer environments in which they envisage Hippocratic databases operating. The efforts of (Karjoth *et al.* 2003) are cited by the designers as they work towards this end.

2- Large numbers of purposes

Despite the intuitive appeal of Hippocratic Databases there are problems with administering large numbers of purposes correctly and with automatically determining concretely the purpose of any given access request (Al-Fedaghi, 2007).

3- Limited Disclosure

Allowing users the ability to dynamically choose the external recipients of their private information poses challenges for limiting disclosure. The Hippocratic designers show identity theft as one such problem. They propose that public-private key cryptography offers a possible solution, but concede that deploying this solution poses its own challenges.

4- Limited Retention

Adhering to the principle of limited retention seems simple enough. On the face of it, it would appear that information should be deleted when it is no longer required. However, data is not only stored in the data table, but in the database logs and past checkpoints. Deleting data from these logs and checkpoints, without affecting recovery will be a challenge.

5- The management of attributes and users purposes

The management of attributes and users' purposes is a complicated issue (Masacci *et al.*, 2005).

6- Safety of Information

Controlling the access to tables can primarily be controlled by the database system. However, the storage media on which the tables are stored may be vulnerable. For example, someone with super user authority may not have permission to access a table, but may gain access to database files using the operating system. While encryption of database files may help, the performance implications it entails will need serious consideration.

7- Openness

Even the principle of openness, which on the face of it appears straightforward, has its own challenges. Users should be able to determine if a database has information stored about them. However, in allowing this determination, the database should not know who issued the query, if they in fact hold no information about the querying user. Additionally, a user whose information is not stored, and who initiates a query for information, should learn nothing beyond the fact that no information is stored.

8- Compliance

Generating audit trails of every access to personal information and making this available to users can be a powerful means to protect privacy. Doing this without paying a large performance penalty is a challenge. A potential solution may be the use of a trusted intermediary. Rather than sending the logs to each individual user, they may be sent to the intermediary. Users can then access log information on demand from the intermediary.

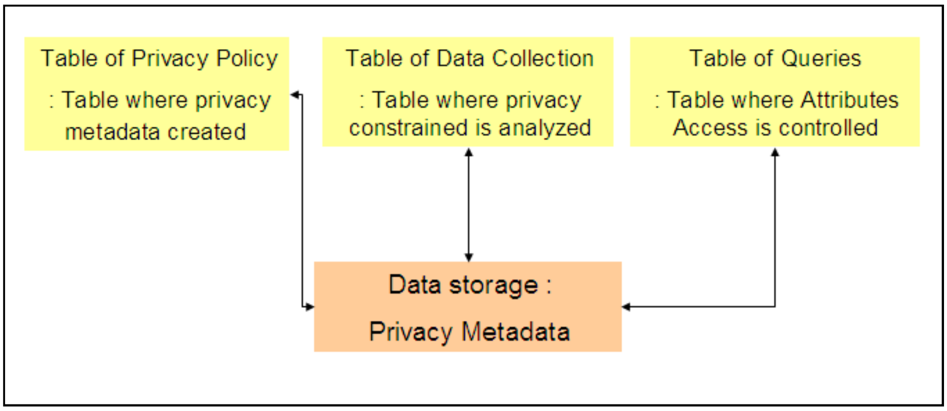


Table of Privacy Policy
: Table where privacy metadata created
Table of Data Collection
: Table where privacy constrained is analyzed
Table of Queries
: Table where Attributes Access is controlled
Data storage :
Privacy Metadata

Figure 14: The infrastructure of the Hippocratic database.

In the next section, a summary that shows the drawbacks and advantages of the Hippocratic database will be given.

2.8.3 Summary of the Drawbacks and Advantages of the Hippocratic Database

There were attempts in the literature to use the Hippocratic database in designing real applications such as (Masacci *et al.,* 2005) and (Omran *et al.,* 2008). But these attempts have stumbled against problems of the Hippocratic database, Such as:

- The numerous number of purposes that can be used as a reason for accessing the database in the Hippocratic model (Al-Fedaghi, 2007), and the problem of mapping these purposes with the authenticated group of users (Omran *et al.,* 2008) and (Masacci *et al.,* 2005).
- Hippocratic Database is the first method to implement privacy at the data level. This requires data access constraints to be defined at the data level, and mapped to attributes in the original database tables. Database implementation and adoption subsequently have added difficulty.

Figure 15 shows how complex the hierarchy and number of purposes would be in the healthcare case. The figure shows only a portion of the possible purposes for selected users. In this figure which is similar to the findings in (Omran *et al*., 2008), the researcher tried to show how complicated it is to for the non-expert database administrator to design his system using principles of Hippocratic database. Using the example of a hospital, he needs to have a good background in the detailed processes of the hospital and then translate them into purposes and hierarchies of users.

The complexity of limiting the huge number of purposes available for a specific domain is highlighted in (Omran et al., 2008).

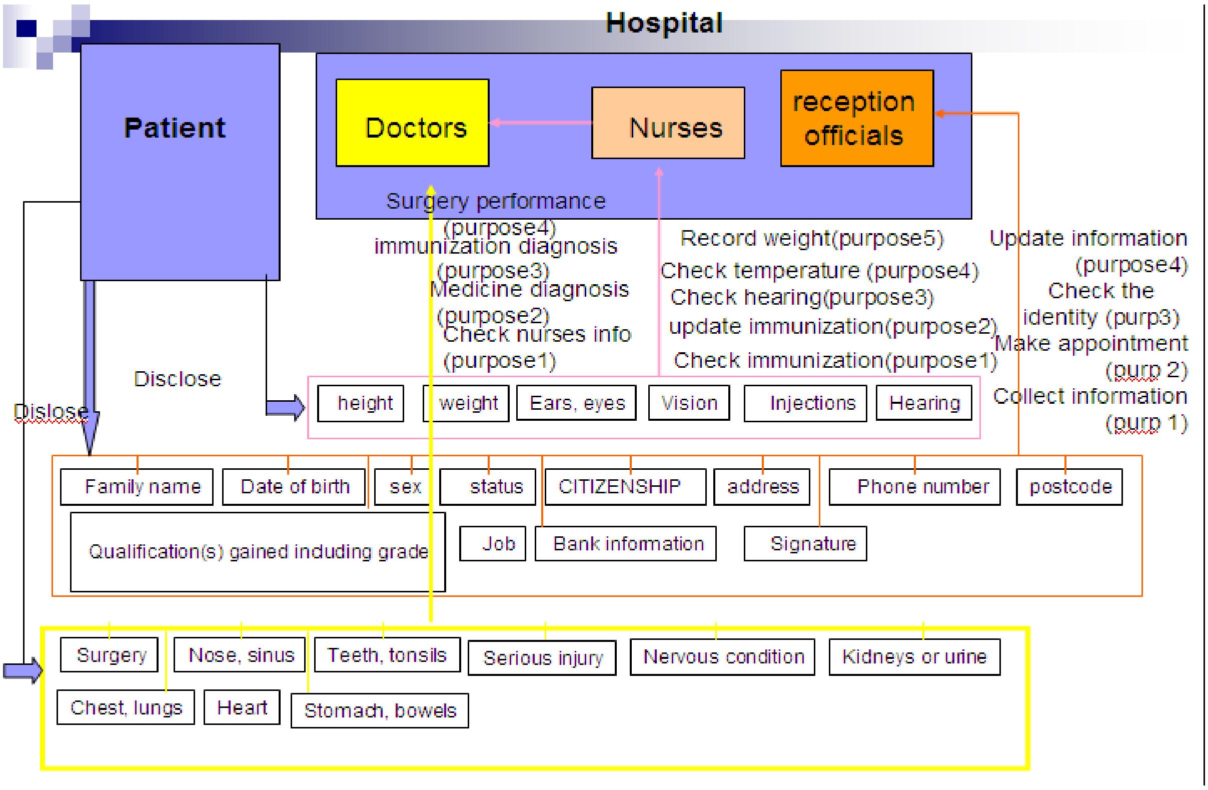


Figure 15: Prototype for Hippocratic database in healthcare application.

In the next section, a data access management that suggests lest parameter for data access management will be presented.

2.9 Chain Method

Chain-Based Access Control is based on the notion of a *chain of acts* (Al-Fedaghi, 2007). In his paper, Al-Fedaghi presented the idea by changing the principle of data access control from *purposes* to *chains of limited acts*. He stated that the management of attributes and users' purposes is a complex issue. To simplify the mapping process, he depends on the idea that users are assigned roles and access purpose permissions are granted to roles associated with tasks or functionalities, not directly to individual users.

The idea of the Chain method has been derived from Al-Fadaghi's personal information theory which has been presented in a long series of publications such as: (Al-Fedaghi *et al.,* 2005), (Al-Fedaghi, 2005), (Al-Fedaghi, 2006a) and (Al-Fedaghi, 2006b). The next sub section will give a brief summary of his personal information theory.

2.9.1 Al-Fedaghi Personal information Theory

Personal Information Theory (PIT) is based on an ethical foundation (Al-Fedaghi, 2006a). Potential abuse of personal/private information raises many ethical, legal, and economic issues. One aspect of (PIT) is Personal Information Ethics (PIE), which is based on the thesis that personal information itself has an intrinsic moral value. Recognition of the intrinsic ethical value of personal information does not imply prohibiting acting upon the information. Rather, it means that, while others may have a right to utilise personal information for legitimate needs and purposes; it should not be done in such a way that devalues personal information as an object of respect (Al-Fedaghi, 2006b). The human-centred significance aspect of personal information derives from its value to a human being as something that hides his/her secrets, feelings, embarrassing facts, etc., and something that gives him/her a sense of identity, security, and, of course, privacy (Al-Fedaghi, 2006a). The notion of security in this context means that personal information would be protected from malicious users while it moves through the seven acts in the PI flow model (collecting, creating, processing, storing, disclosing, using and mining). For example, the typical countermeasure against attacks in the processing act involves enforcing access permissions policies. When malicious users gain access to personal data, the database system is responsible for protecting the personal information.

Personal information privacy involves acts in reference to personal information. For example, creating, collecting, processing, and disclosing as reflected in the PI flow model, are examples of these acts. Al-Fedaghi grasps the difference between privacy and security in the context of personal information from the Health Insurance Portability and Accountability Act (HIPAA), which may be said to be a comprehensive venture in the direction of privacy. He uses of the following HIPAA clause:

"Security refers to the specific measures and efforts taken to protect privacy and to ensure the integrity of personal information. Security is the ability to prevent unauthorised breaches of privacy, such as might occur if data are lost or destroyed by accident, stolen by intent, or sent to the wrong person in error."

After a deep analysis he gives his definition of personal information which is: **"**Personal information is any linguistic expression that has referent(s) of type individual, assuming that p(X) is a sentence such that X is the set of its *referents and V is the verb used in this sentence"*. According to this definition there are two types of personal information:

(1) p(X) is atomic personal information if X ∩ V is the singleton set {X}. That is, atomic personal information is an assertion that has a single human referent.

(2) p(X) is compound personal information if |X ∩ V| is greater than 1. That is, compound personal information is an expression that has more than one human referent.

A single piece of atomic personal information may have many possessors; where its proprietor may or may not be among them. A possessor refers to any entity that knows, stores, or owns the information. Any compound personal statement is privacy-reducible to a set of atomic personal statements. Personal information privacy involves acts on personal information in the context of creating, collecting, processing, and disclosing this type of information.

At this point, all the basic requirements needed to form the chain method to be suggested as a candidate for managing data access have been covered as the next subsection illustrates.

2.9.2 The infrastructure of the Chain method

Unlike RBAC, Chain doesn’t need to have long, complicated policies for each group of roles (Al-Fedaghi, 2007). Instead, a set of 7 limited acts (Creating, Processing, Disclosing, Storing, Collecting, Using and Mining as shown in Figure 17), are distributed amongst the different group of roles. These acts are the policy and purpose of why this group of roles is accessing the database and at the same time it includes the action that the user can apply to the database.

As shown in Figure 16, data usage can be divided into four phases, namely creation, collection, processing and disclosure of personal information. Each phase can be associated with a number of allowed acts. Personal information can be created by proprietors (i.e., the data subject), by non-proprietors (i.e., any data recipient different from the data subject), or can be deduced from existing information (e.g., using data mining). Created information can be either used (e.g., for decision making), stored, or disclosed. In addition, information can enter into the processing and disclosing phases. The processing of personal information involves storing, using, and mining personal information. The disclosure phase involves releasing personal information to other actors.

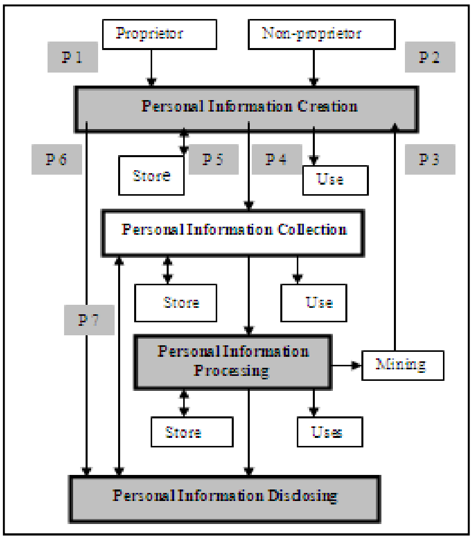

Proprietor
Non-proprietor
P 1
P 2
Personal Information Creation
P 6
Store
P 5
P 4
Use
P 3
Personal Information Collection
Store
Use
P 7
Personal Information Processing
Mining
Store
Uses
Personal Information Disclosing

Figure 16: Personal Information flow model (based on (Al-Fedaghi, 2007)).

Al-Fedaghi argues that each role can be translated to a chain of acts on personal information, such as in Figure 16. He further proposes that any piece of personal information only requires a limited set of acts that can be operated on it. He claimed that those limited acts could be used to design a more robust data access control mechanism that could safeguard personal information privacy. So, instead of huge policy tables, there is only a manageable set of limited acts.

Using the PI flow model, the researcher can build a system that involves a proprietor on one side and others (other persons, agencies, companies, etc.) who perform different types of activities in the PI transformations among the four phases of flow of personal information. Al-Fedaghi refers to any of these as PI agents. PI agents may include any one who participates in activities over PI. The proprietor is not accepted as an agent with respect to his/her own PI (Al-Fedaghi, 2007). In order to bring the principles of the Chain method that Al-Fedaghi has developed, this researcher has applied it to the healthcare domain (which is the domain of our interest in this thesis).

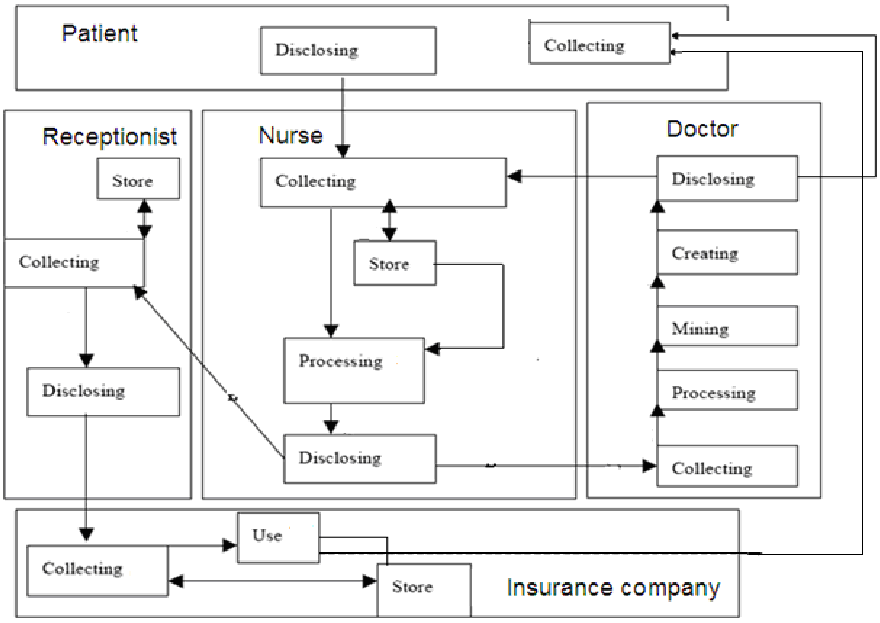


Figure 17: Architecture of Information Flow (based on Figure 4 in (Al-Fedaghi, 2007).

Figure 17 represents the personal information flow model of a typical healthcare scenario. Here, the proprietor of personal information is the patient, whereas the non-proprietors are doctors, nurses, receptionists and insurance companies. Every actor involved in the data processing is represented along with the acts that he can perform. For instance, nurses can collect, store, process and disclose patient information. The arrows between acts represent the allowed chains of acts. For instance, the information disclosed by the patient can be collected by the nurse, who in turn can either store it or process it. If the nurse stores them, she can either collect new information or process it. In (Omran *et al.,* 2010a) the researchers have drawn the basic lines of the specifications of Chain method construction.

2.9.3 The Chain Method in Real Applications

The chain method has not been implemented before the work on this project. Therefore, when the researcher started the real work on it, many practical problems appeared and the researcher started to fill in the gaps resulting from Al-Fedaghi's work. And in order to do so, she published a series of publications such as: (Omran *et al.,* 2008), (Omran *et al.,* 2009c), (Omran *et al.,* 2010) and (Omran *et al.,* 2012).

While working on the Chain, the researcher has realised the simplicity of its design compared with the other methods such as: RBAC, TBAC, Hippocratic and XACML (Omran *et al.,* 2010) and (Omran *et al.,* 2012). The results of this study will be fully presented in Chapters 6, 7 and 8.

It has been shown that if the acts of the Chains are written compared with the policies of RBAC-both in XACML, one gets much straight forward and less commands in favour of the Chain as will be shown in Chapter 8.

2.9.4 Summary

From the above, it becomes clear that the Chain gives us the first key for a solution to the problem because:

- Chains have been proposed to simplify the complexity of Hippocratic databases (Omran *et al.,* 2010)
- The chain improves upon the RBAC with its simple design (Omran *et al.* 2011)
- In Al-Fedaghi (2007) a personal information flow model is proposed that specifies permitted acts on personal information.
- Chains of these acts can be used to control acting on personal information instead of purposes used in privacy access control.

But the main outstanding problems for the Chain method are:

- It suffers from the same problem as RBAC in not being able to determine the context.

- The Chain of acts method has not been applied in real applications because of difficulty in defining the data for the chains of acts.

These problems were the motivation that has inspired us to go and look for our second key solution which is “Semantics” in order to be able to specify the 7 acts of Al-Fedaghi for the domain and get use of the simple design of the Chain model.

2.10 XACML and SAML

When the network was contained within a single building or campus, the problem was relatively simple and generally handled by software that was hooked into the operating system. But today's networks involve interconnected segments distributed across the country and around the globe, and many of these are also joined to the public Internet.

Markup Language (SAML) defines how identity and access information is exchanged and lets organisations convey security information to one another without having to change their own internal security architectures. However SAML can only communicate information. How to use that information is where XACML comes in. This is a language, which uses the same definitions of subjects and actions as SAML, and offers a vocabulary for expressing the rules needed to define an organisation's security policies and make authorisation decisions. XACML has two basic components.

The first is an access-control policy language that lets developers specify the rules about who can do what and when. The other is a request/response language that presents requests for access and describes the answers to those queries.

XACML is an OASIS standard that describes both a policy language and an access control decision request/response language (both written in XML) (OASIS 2005). The policy language is used to describe general access control requirements, and has standard extension points for defining new functions, data types, combining logic, etc. The request/response language lets one form a query to ask whether or not a given action should be allowed, and interpret the result. The response always includes an answer about whether the request should be allowed using one of four values: Permit, Deny, Indeterminate (an error occurred or some required value was missing, so a decision cannot be made) or Not Applicable (the request can't be answered by this service).

The typical setup is that someone wants to take some action on a resource. They will make a request to whatever actually protects that resource (like a file system or a web server), which is called a Policy Enforcement Point (PEP). The PEP will form a request based on the requester's attributes, the resource in question, the action, and other information pertaining to the request. The PEP will then send this request to a Policy Decision Point (PDP), which will look at the request and some policy that applies to the request, and come up with an answer about whether access should be granted. That answer is returned to the PEP, which can then allow or deny access to the requester. It should be noted that the PEP and PDP might both be contained within a single application, or might be distributed across several servers. In addition to providing request/response and policy languages, XACML also provides the other pieces of this relationship, namely finding a policy that applies to a given request and evaluating the request against that policy to come up with a yes or no answer.

There are many existing proprietary and application-specific languages for doing this kind of thing but XACML has several points in its favour (OASIS 2003):

- It is standard. By using a standard language, a user doesn't need to change his own system each time, and he doesn't need to think about all the tricky issues involved in designing a new language. Plus, as XACML becomes more widely deployed, it will be easier to interoperate with other applications using the same standard language.

- It is generic. This means that rather than trying to provide access control for a particular environment or a specific kind of resource, it can be used in any environment. One policy can be written which can then be used by many different kinds of applications, and when one common language is used, policy management becomes much easier.

- It is distributed. This means that a policy can be written which in turn refers to other policies kept in arbitrary locations. The result is that rather than having to manage a single monolithic policy, different people or groups can manage sub-pieces of policies as appropriate, and XACML knows how to correctly combine the results from these different policies into one decision.

- It is powerful. While there are many ways the base language can be extended, many environments will not need to do so. The standard language already supports a wide variety of data types, functions, and rules about combining the results of different policies. In addition to this, there are already standard groups working on extensions and profiles that will hook XACML into other standards like SAML and LDAP, which will increase the number of ways that XACML can be used.

To give you a better idea of how all these aspects fit together, what follows is a discussion of XACML policy, which will demonstrate many of the standard features of the language. Note that XACML is a rich language, so only some of its features are shown here. You should look at the specification for more information on all of the features.

2.10.1 Top-Level Constructs: Policy and PolicySet

At the root of all XACML policies is a Policy or a PolicySet. A PolicySet is a container that can hold other Policies or PolicySets, as well as references to policies found in remote locations. A Policy represents a single access control policy, expressed through a set of Rules. Each XACML policy document contains exactly one Policy or PolicySet root XML tag.

Because a Policy or PolicySet may contain multiple policies or rules, each of which may evaluate according to different access control decisions, XACML needs some way of reconciling the decisions each makes. This is done through a collection of combining algorithms. Each algorithm represents a different way of combining multiple decisions into a single decision. There are Policy Combining Algorithms (used by PolicySet) and Rule Combining Algorithms (used by Policy). An example of these is the “Deny” Overrides Algorithm, which says that no matter what, if any evaluation returns “Deny”, or no evaluation permits, then the final result is also Deny. These combining algorithms are used to build up increasingly complex policies, and while there are seven standard algorithms, you can build your own to suit your needs.

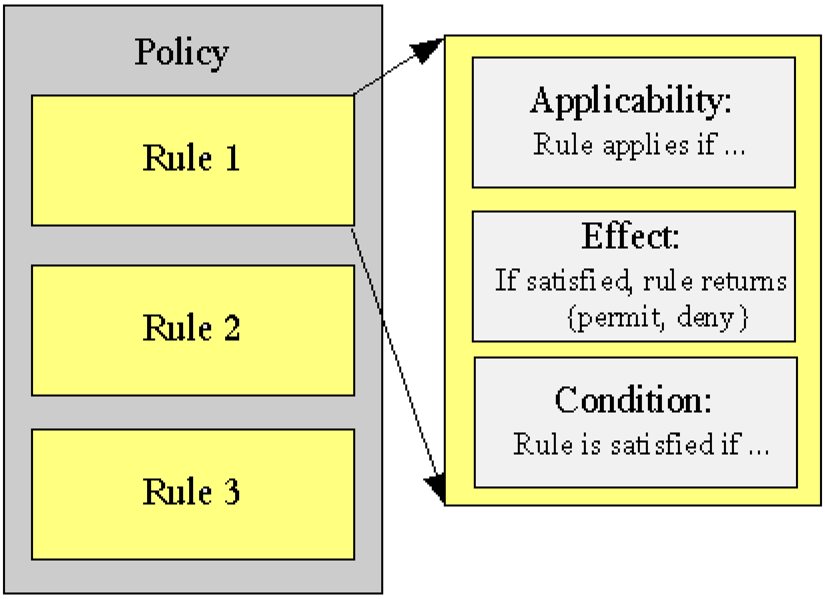


Figure 18: XACML Policy model.

2.10.2 Targets and Rules

Part of what XACML Policy Decision Point (PDP) - Point (which evaluates and issues authorisation decisions) needs to do is to find a policy that applies to a given request. To do this, XACML provides another feature called a target. A Target is basically a set of simplified conditions for the Subject, Resource and Action (a Subject element is the entity requesting access, an Action element defines the type of access requested on the Resource (such as a file system or a web server)) that must be met for a PolicySet, Policy or Rule to apply to a given request. These use Boolean functions to compare values found in a request with those included in the Target. If all the conditions of a Target are met, then its associated PolicySet, Policy, or Rule applies to the request. In addition to being a way to check applicability, Target information also provides a way to index policies, which is useful if you need to store many policies and then quickly sift through them to find which ones apply. For instance, a Policy might contain a Target that only applies to requests on a specific service. When a request to access that service arrives, the PDP will know where to look for policies that might apply to this request because the policies are indexed based on their Target constraints. Note that a Target may also specify that it applies to any request.

Once a Policy has been found and verified to apply to a request, its Rules are evaluated. A policy can have any number of Rules which contain the core logic of an XACML policy. The heart of most Rules is a Condition, which is a Boolean function. If the Condition evaluates to true, then the Rule's Effect (a value of Permit or Deny that is associated with successful evaluation of the Rule) is returned. Evaluation of a Condition can also result in an error (Indeterminate) or discovery that the Condition doesn't apply to the request

(Not Applicable). A Condition can be quite complex, built from an arbitrary nesting of non-Boolean functions and attributes.

2.10.3 Attributes, Attribute Values, and Functions

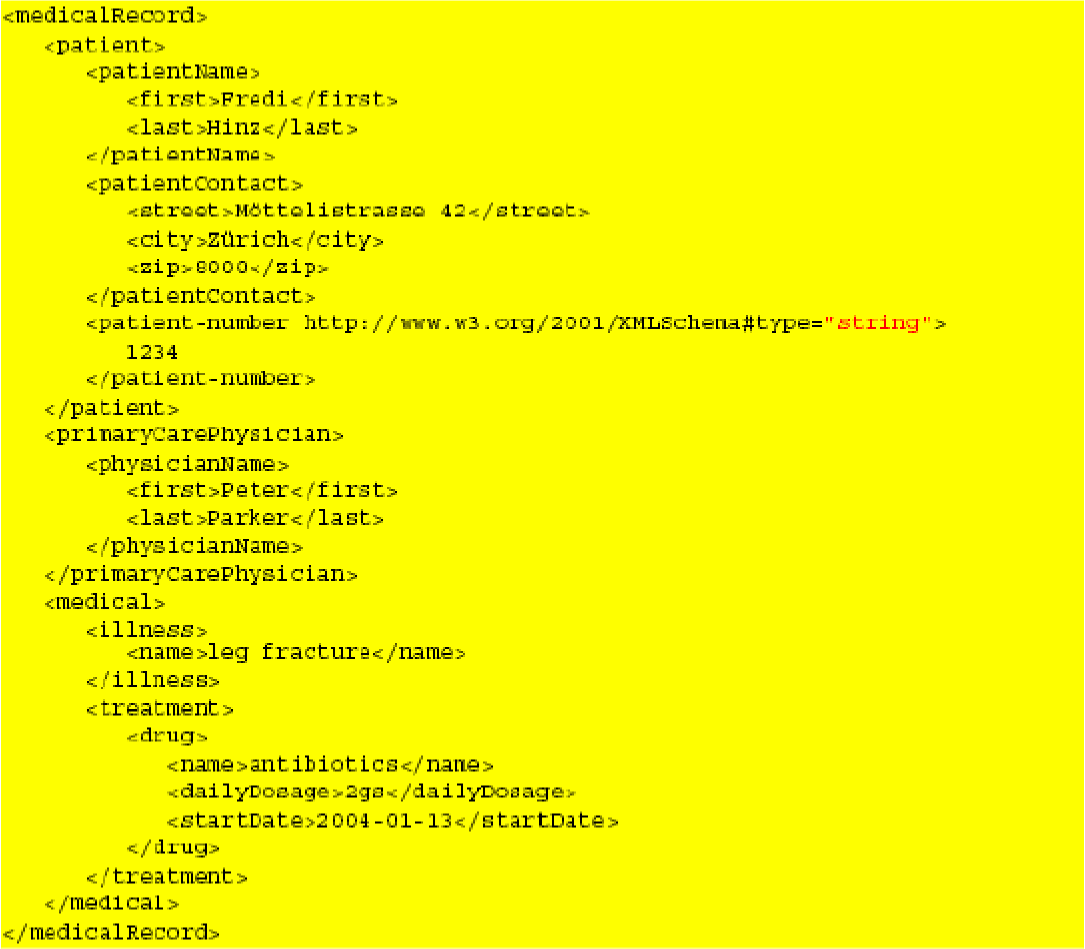
The currency that XACML (OASIS 2003) deals in is attributes. Attributes are named values of known types that may include an issuer identifier or an issue date and time. Specifically, attributes are characteristics of the Subject, Resource, Action, or Environment in which the access request is made. A user's name, their security clearance, the file they want to access, and the time of day are all attribute values. When a request is sent from a PEP to a PDP, that request is formed almost exclusively of attributes, and they will be compared to attribute values in a policy to make the access decisions.

A Policy gets attribute values from a request or from some other source through two mechanisms: the Attribute Designator and the Attribute Selector. An Attribute Designator lets the policy specify an attribute with a given name and type, and optionally an issuer as well, and then the PDP will look for that value in the request, or elsewhere if no matching values can be found in the request. There are four kinds of designators, one for each of the types of attributes in a request: Subject, Resource, Action, and Environment. Because Subject attributes can be broken into different categories, Subject Attribute Designators can also specify a category to look in. Attribute Selectors allow a policy to look for attribute values through an XPath query. A data type and an XPath expression are provided, and these can be used to resolve some set of values either in the request document or elsewhere.

Both the Attribute Designator and the Attribute Selector can return multiple values (since there might be multiple matches in a request or elsewhere), so XACML provides a special attribute type called a Bag. Bags are unordered collections that allow duplicates, and are always what designators and selectors return, even if only one value was matched. In the case that no matches were made, an empty bag is returned, although a designator or selector may set a flag that causes an error instead in this case.

Once some Bag of attribute values has been retrieved, they need to be compared in some way to expected values to make access decisions. This is done through a powerful system of functions. Functions can work on any combination of attribute values, and can return any kind of attribute value supported in the system. Functions can also be nested, so you can have functions that operate on the output of other functions, and this hierarchy can be arbitrarily complex. Custom functions can be written to provide an even richer language for expressing access conditions.

One thing to note when building these hierarchies of functions is that most functions are defined as working on specific data types (like strings or integers) while designators and selectors always return Bags of values. To handle this, XACML defines a collection of standard functions of the form [type]-one-and-only, which accept a bag of values of the specified type and return the single value if there is exactly one item in the bag, or an error if there are zero or multiple values in the bag. This is one of the most common functions that one will see in a Condition. Type-one-and-only functions are not needed in Targets, however, since the PDP automatically applies the matching function to each element of a Bag.



```
<medicalRecord>
   <patient>
      <patientName>
         <first>Fredi</first>
         <last>Hinz</last>
      </patientName>
      <patientContact>
         <street>Möttelistrasse 42</street>
         <city>Zürich</city>
         <zip>8000</zip>
      </patientContact>
      <patient-number http://www.w3.org/2001/XMLSchema#type="string">
         1234
      </patient-number>
   </patient>
   <primaryCarePhysician>
      <physicianName>
         <first>Peter</first>
         <last>Parker</last>
      </physicianName>
   </primaryCarePhysician>
   <medical>
      <illness>
         <name>leg fracture</name>
      </illness>
      <treatment>
         <drug>
            <name>antibiotics</name>
            <dailyDosage>2gs</dailyDosage>
            <startDate>2004-01-13</startDate>
         </drug>
      </treatment>
   </medical>
</medicalRecord>
```

Figure 19: An example to show how the XACML represents a policy that states: Physician can see any medical record (OASIS 2003).

There was an attempt to join the RBAC with XACML as in (Crampton, 2003) and (Crampton, 2004). The aims were to:

- Obtain a closer correspondence between XACML policies and the RBAC model
- Provide a more natural way of defining:
  - Role hierarchies
  - Permissions

  – Permission-role assignment
- Support the idea of complex permissions

But the main problem was that there is no mechanism for associating subjects with roles and it was designed for centralised systems with a known user population and is hardly suitable for web services.

2.10.4 Drawbacks of XACML

1) Readability of target elements

In XACML targets, normally, it is possible to have several matches on a target element (subject, resource and action). One problematic issue with targets is that they are hard to read because there is no explicit indication about whether the multiple matching specifications are linked via conjunction or disjunction operators.

The second issue with XACML targets is the limitation in expressive power due to the fact that subjects, resources and actions are related implicitly by a conjunction operator. This prevents writing logical expressions where various combinations of subjects, resources and actions could be specified in a single policy or rule. For example two different resources r1 and r2 that would be associated respectively with action a1 for r1 and a2 for r2 would require two separate policies or rules to be specified using an ordinary XACML target. If, on the other hand, the target could be written using a single logical expression mixing subjects, resources and actions using disjunction operators, it would require only one policy.

2) Disadvantages of the rule target/condition conjunctive Model

A third issue with XACML targets is similar to the second but with the additional problem of a rule target's relation with the rule's condition part. The condition part applies to the entire target either at the rule, policy or policy set level and thus here again multiple policies or rules must be specified with individual combinations of conditions for the target elements. This has the immediate result of forcing the administrator to write separate policies for each combination of targets and conditions, while the same could have been represented in a single rule or policy if the logic of target elements could be related to the condition using a single logical expression.

Thus, in order to alleviate the above problems, it is strongly recommended to use single logical expressions that allow the combination of target elements and conditions. It will be shown that this has the direct result of reducing the number of policies or rules, thus making the administration of policies considerably easier.

In the XACML RBAC profile, role inheritance is represented using the *Policy SetIdReference* and *PolicyIdReference* language elements. While inheritance is an important reusability technique, it has the undesirable side effect of dispersing information. This dispersal is by definition unavoidable. *There is a need for a XACML rule inheritance mechanism as* XACML rules currently do not have such an inheritance mechanism, even in version 3.0.

Decentralised access control, for example, requires sophisticated techniques for conflict detection and for managing rules across multiple applications with different rule formats. XACML is an OASIS standard whose interoperability qualities help in solving the latter problem. XACML has its own limitations, however. In particular, although it has the expressive power to specify very complex conditions like those needed in the ABAC (Attribute Based Access Control) model, users tend to avoid using its full power because of its verbosity.

2.10.5 Discussion

In general XACML provides promising method to preserve privacy while taking care of the context (but the context sensitivity in this method is scattered through the system with a set of complicated rules). Therefore the main disadvantages of XACML can be outlined as its being complex and hard to understand (policies and rules) (OASIS, 2003).

When targeting a complex system such as large hospitals and healthcare institutions, the design and application of XACML systems becomes even more difficult. Taking care of all these targets and attributes (in XML format) in an ocean of information flow would be confusing. Therefore, the XACML in its original situation and without real developments and enhancement would not be a complete solution to the problem in hand.

The usage of semantics in XACML has paved the way for us to include semantics in our solution. In a way, that does not repeat the same errors and problems that are in XACML. For example, the purpose of accessing the database and the action that would be performed can be combined in one goal, in order to make use of the principles found in the chain method to access the database, can be controlled with semantics as in XACML. But even the semantics that will be used, it would be based on durable personal information ontology and not scattered throughout the system as in the case of XACML. That's why it is important to have a closer look at on the semantics in the next section.

2.11 Ontology and Semantics

The history of artificial intelligence shows that knowledge is critical for intelligent systems. In many cases, better knowledge can be more important for solving a task than better algorithms. To have truly intelligent systems, knowledge needs to be captured, processed, reused, and communicated. Ontologies support all these tasks (Obitko, 2007).

The development of ontologies - explicit formal specifications of the terms in the domain and relations among them (Gruber, 1993) - has been moving from the realm of Artificial Intelligence laboratories to the desktops of domain experts. Ontologies are starting to become common on the World-Wide Web. The WWW Consortium (W3C) is building up the Resource Description Framework, a language for encoding knowledge on Web pages to make it understandable to electronic agents searching for information. Many disciplines now develop standardised ontologies that domain experts can use to share and annotate information in their fields. Medicine, for example, has produced large, standardised, structured vocabularies such as SNOMED (Price and Spackman, 2000) and the semantic network of the Unified Medical Language System (Humphreys and Lindberg, 1993). Nevertheless, none of these medical ontologies describe the process of medical care. At this a point that the researcher has stopped at and started from to develop a complete Medical care ontology that contains all the medical processes in addition to the medical personal information.

“Ontology” in general, defines a common vocabulary for people who need to share information in a domain. It includes machine-interpretable definitions of basic concepts in the domain and relations among them. The term "ontology" can be defined as an explicit specification of conceptualisation. Ontologies capture the structure of the domain, i.e. conceptualisation. This includes the model of the domain with possible restrictions. The conceptualisation describes knowledge about the domain, not about the particular state of affairs in the domain. In other words, the conceptualisation is not changing, or changes very rarely. The Ontology is then the specification of this conceptualisation - which is specified by using a particular modelling language and particular terms. Formal specification is required in order to be able to process ontologies and operate on ontologies automatically.

The Artificial Intelligence literature contains many definitions of an ontology; many of these contradict one another. For the purposes of this dissertation an **ontology** is a formal explicit description of concepts in a domain of discourse (**classes,** sometimes called **concepts**), properties of each concept describing various features and attributes of the concept (**slots,** sometimes called **roles** or **properties**), and restrictions on slots (**facets,** sometimes called **role restrictions**). An ontology together with a set of individual **instances** of classes constitutes a **knowledge base**. In reality, there is a fine line where the ontology ends and the knowledge base begins (Natalya *et al.,* 2003).

Some of the reasons to investigate and develop ontologies are:

- To share common understanding of the structure of information among people or software agents;
- To enable reuse of domain knowledge;
- To make domain assumptions explicit;
- To separate domain knowledge from the operational knowledge;
- To analyse domain knowledge.

An Ontology describes a domain, while a knowledge base (based on an ontology) describes a particular state of affairs. Each knowledge based system or agent has its own knowledge base, and only what can be expressed using an ontology can be stored and used in the knowledge base. When an agent wants to communicate to another agent, he uses the constructs from some ontology. In order to understand in communication, ontologies must be shared between agents (Obitko, 2007).

2.11.1 Specifications of Conceptualisations

The second definition of ontology given above, "explicit specification of conceptualisation", comes from Thomas Gruber (Gruber, 1993). The exact meaning depends on the understanding of the terms "specification" and "conceptualisation". Explicit specification of conceptualisation means that the ontology is a description (like a formal specification of a program) of the concepts and relationships that can exist for an agent or a community of agents. This definition is consistent with the usage of ontology as set of concept definitions, but is more general.

A conceptualisation can be defined as an intentional semantic structure that encodes implicit knowledge constraining the structure of a piece of a domain. An ontology is a (partial) specification of this structure, i.e., it is usually a logical theory that expresses the conceptualisation explicitly in some language. A conceptualisation is language independent, while an ontology is language dependent. Notice that an ontology does not have to express all the possible constraints - the level of detail in conceptualisation depends on the requirements of the intended application and expressing conceptualisation in an ontology in addition depends on the used ontology language.

In this sense, an ontology is important for the purpose of enabling knowledge sharing and reuse. An ontology is in this context a specification used for making ontological commitments. Practically, an ontological commitment is an agreement to use a vocabulary (i.e., ask queries and make assertions) in a way that is consistent (but not complete) with respect to the theory specified by an ontology. Agents then commit to ontologies and ontologies are designed so that the knowledge can be shared among these agents.

The representation of a body of knowledge (knowledge base) is based on the specification of conceptualisation. A conceptualisation is an abstract, simplified view of the world that should be represented for some purpose. For knowledge-based systems, what "exists" is what can be represented. When the knowledge of a domain is represented in a declarative formalism, the set of objects that can be represented is called the universe of discourse. This set of objects and the describable relationships among them are reflected in the representational vocabulary with which a knowledge-based program represents knowledge. Thus, in the context of AI, the ontology of a program can be described by defining a set of representational terms. In such an ontology, definitions associate the names of entities in the universe of discourse (e.g. classes, relations, functions, or other objects) with descriptions of what the names mean, and formal axioms that constrain the interpretation and well-formed use of these terms. Formally it can be said that an ontology is a statement of a logical theory.

The backbone of an ontology is often a taxonomy. A taxonomy is a classification of things in a hierarchical form. It is usually a tree or a lattice that express the subsumption relation, where, A subsumes B means that everything that is in A is also in B. An example is classification of living organisms. The taxonomy usually restricts the intended usage of classes - where classes are subsets of the set of all possible individuals in the domain. A taxonomy of properties can be defined as well.

However, ontologies need not be limited to taxonomic hierarchies of classes and need not be limited to definitions that only introduce terminology and do not add any knowledge about the world. To specify a conceptualisation, axioms that constrain the possible interpretations for the defined terms may be also needed. Pragmatically, an ontology defines the vocabulary with which queries and assertions are exchanged among agents. The ontological commitment is then a guarantee of consistency for communications.

2.11.2 Semantics and semantic models

Semantics definition as appeared in (Richmond and Thomason 1996) is the study of the meaning of linguistic expressions. The language can be a natural language, such as English or Navajo, or an artificial language, like a computer programming language. Meaning in natural languages is mainly studied by linguists. In fact, semantics is one of the main branches of contemporary linguistics. Theoretical computer scientists and logicians think about artificial languages. In some areas of computer science, these divisions are crossed. In machine translation, for instance, computer scientists may want to relate natural language texts to abstract representations of their meanings; to do this, they have to design artificial languages for representing meanings.

Semantic network (also called concept network) is a graph, where vertices represent concepts and where edges represent relations between concepts. Semantic network at the level of ontology expresses vocabulary that is helpful especially for human, but that still can be usable for machine processing. The relations between concepts that are used in semantic networks are as follows:

Synonym - concept A expresses the same thing as concept B
Antonym - concept A expresses the opposite of concept B
Meronym, holonym - part-of and has-part relation between concepts
Hyponym, hypernym - inclusion of semantic range between concepts in both directions

The complexity of communication in information technology is characteristic for more than the last ten years. A number of technical solutions have been provided as technical standards to improve the communication facilities especially between processes. However, standards as SQL, CORBA, DCOM define a common syntax, only. There is no way to support semantic standards based on standard terminology.

This section gives a short introduction to the context sensitivity models, introduction to semantic model, the meaning of semantic interfaces and finally several reasons for the urgent need of semantic interfaces are shown.

According to (Doumen *et al.,* 2005), context sensitivity is needed for the following reasons:

- Current security static and intrusive

  Something you have such as (username) and (password) and there are numerous usernames / passwords but sometimes Identity is not relevant and it would be hard to audit it.

- Increasing use of context information

  In multilevel authentication systems such as healthcare, the context plays a vital role in giving authentication rights.

- Use of context to enhance / replace existing security

  Context is used to make it more flexible and less intrusive.

2.11.3 Context-based security model

The Context-based security model emerged recently as a new approach to cope with the new types of security problems introduced by the high mobility of pervasive systems and the heterogeneity of devices used in these types of environments. A Context- based security model treats context as a first-class principle both in the specification and enforcement of policies. It models and represents the contexts in which agents operate and to which policies are associated, defines what actions are permitted or forbidden on resources in specific contexts, defines the actions that must be performed on resources in specific contexts, and dynamically associates agents with contexts.

In the context based security model, contextual graphs help specifying context -based security policies in a pervasive environment and are used as a management tool that eases security administration for complex environments with many heterogeneous services and devices. The exploitation of context as a mechanism for grouping applicable policies (not as a limit to the applicability of already retrieved policies as in traditional access control solutions) simplifies access control management by increasing policy specification reuse and by simplifying policy update and revocation. It also includes fine-grained control, policy specifications, and policy enforcement characteristics. However, this model is relatively new and requires further testing within the collaborative systems domain.

When the authorised user enters a certain security context, the context will be associated with the corresponding action automatically. The obligated user cannot access the security context. Here context can be any useful information about the world, such as the user location, the characteristics of the underlying device, relationship with other users and many others.

2.11.4 Semantic model

A **semantic data model** in software engineering has various meanings (NIST, 2012):

1. It is a conceptual data model in which semantic information is included. This means that the model describes the meaning of its instances. Such a semantic data model is an abstraction that defines how the stored symbols (the instance data) relate to the real world (Computer Systems Laboratory of the National Institute of Standards and Technology (NIST, 2012).
2. It is a conceptual data model that includes the capability to express information that enables parties to the information exchange to interpret meaning (semantics) from the instances, without the need to know the meta-model. Such semantic models are fact oriented (as

opposed to object oriented). Facts are typically expressed by binary relations between data elements, whereas higher order relations are expressed as collections of binary relations. Typically binary relations have the form of triples: Object-RelationType-Object. For example: the Eiffel Tower <is located in> Paris.

Typically, the instance data of semantic data models explicitly includes the kinds of relationships between the various data elements, such as <is located in>. To interpret the meaning of the facts from the instances it is required that the meaning of the kinds of relations (relation types) is known. Therefore, semantic data models typically standardise such relation types. This means that the second kind of semantic data model enables the instances to express facts that include their own meaning. The second kind of semantic data model is usually meant to create semantic databases. The ability to include meaning in semantic databases facilitates building distributed databases that enable applications to interpret the meaning from the content. This implies that semantic databases can be integrated when they use the same (standard) relation types. This also implies that in general they have a wider applicability than relational or object oriented databases.

2.11.5 The Need for Semantic Interface

Communication involves always an interface, an agreement that allows exchanging information between objects. Only persons and processes are considered as communicating objects, nevertheless there exist many other communicating objects such as exchanging information between two agents (e.g. hospital and insurance company). With this restriction one can vary between three types of communication:

- "Person to Person"
- "Person to Process"
- "Process to Process" communication.

“Person to Person” communication has been highly standardised during human’s history by means of natural language. There are about a hundred local standards (languages) for local communications. During the last century English became more and more an international standard.

In contrast to natural language there is no general agreement on semantics in process languages. Programmers choose names for program variables or database attributes according to their taste. When reading an SQL statement or a C-program it is in general not possible to interpret the meaning of the syntax because of using very specific technical names (terms). This causes difficulties not only in “Person to Process” communication, but also in “Process to Process” communication.

So in short, a semantic interface is “an agreement on terms and meaning, a language within a group of communicating objects. In this case the language is not only defined by its syntax but also by its semantic. Without semantic interface communication is impossible”.

The following are main features for semantic interfaces:

- A semantic interface is a natural way to handle the increasing complexity of information structures.

- Semantic interfaces are one step to “Person to Process” communication via natural language.

- Semantic interfaces used in “Process to Process” communication will increase the flexibility of standard software. Metadata driven standard software becomes universal and fits into any environment.

- Metadata driven applications and standard software becomes (metadata) database independent. Modern technologies as object orientation (object-oriented databases) and standard interfaces (COM, CORBA, XML) are the technical background that makes it possible to build and to use semantic interfaces.

2.11.6 Semantics applications

Currently, describing the semantics of Web services is a very active research area. DAML-S (DAML-S, 2001) (later OWL-S) is a comprehensive effort defining an upper ontology for Web services. Service discovery through DAML-based languages is also addressed in the literature (Denker *et al.,* 2003) and (Paolucci *et al.,* 2001) where artificial intelligence techniques are used to discover services. In (Omelayenko *et al*., 2002), an RDF mapping meta-ontology, called RDF Translation (RDFT), is proposed which specifies a language for mapping XML DTDs to and from RDF Schemas for business integration tasks.

The P2P paradigm is used to improve semantic interoperability, in particular in revealing new possibilities on how semantic agreements can be achieved. It is argued that establishing local agreements is a less challenging task than establishing global agreements by means of globally agreed schemas or shared ontologies. Once such local agreements exist, through the semantic gossiping" process proposed, global agreements can be achieved in a P2P manner.

There were also attempts to make use of semantics in the healthcare domain such as in (Dogac, 2006). In that paper they tried to provide interoperability in the healthcare domain. They represent healthcare applications as semantically enriched Web services and they stated that "only very recently semantics and Web services started to appear in the medical domain". But they did not investigate a comprehensive of completed ontology that describes the domain and services, as their aim was identifying the need for service functionality and service message ontologies to semantically annotate Web services. It can be seen that work on semantics in general and in the medical domain specific still progresses. It attracts a huge number of researchers to work on it because of its importance. Therefore it has been decided to make use of semantics and to integrate it in the work described in this thesis as will be presented in Chapter 6.

2.12 Conclusion and Summary of Outstanding Problems with Traditional Methods

On the way to solving the problem of the personal information privacy violation problem, the researcher highlighted in this chapter the most significant attempts to protect privacy in the literature. And from the discussion raised at the end of each method was seen that there is no perfect method in the literature that preserves privacy while keeping simplicity, reliability and context sensitivity. The outstanding problems found in the different approaches could be summarised as follows:

- A need for expert programmers that could implement complex systems such as of those of the Hippocratic database and the RBAC to take care of the inherent complexity in managing personal information and its legitimate access and use and the consequent risks of errors in setting up data access correctly;

- The second problem concerns the way Data access management approaches work in run-time as they need to be sensitive to the context in which data and functionality is accessed and more dynamic to be more effective and precise in the application of permissions and restrictions.

In the health care domain, physicians and practitioners are concerned about serious threats to patient privacy due to information gathering methods, record accuracy and access, and unauthorised secondary use (Baume *et al.,* 2000).

There are many attempts to protect privacy in the literature. A research stream focuses on the development of machine readable privacy policies and mechanisms that assist end-users in understanding those policies. The most prominent privacy language is the Platform for Privacy Preferences Project (P3P) (W3C website, 2002). P3P is a W3C standard, which enables organisations to express their privacy practices in a standard format, using an XML-based policy specification language. Privacy policies can be retrieved automatically and interpreted easily by policy-checking agents on the user's behalf, such as Microsoft's Internet Explorer or Privacy Bird of CMU. These agents check the website privacy policies against user preferences and carry out the compensation actions when policies are in conflict with user preferences. A main limitation of P3P is the lack of on enforcement mechanism that guarantees that the privacy promises made by the organisation are fulfilled.

Protection of sensitive information is usually carried out by access control. Examples of access control models are RBAC (Sandhu, 1998) and TBAC (Thomas and Sandhu, 1997). The Role-Based Access Control (RBAC) is proposed to address the limitations of Mandatory Access Control (MAC) and Discretionary Access Control (DAC). In particular, RBAC simplifies the management of permissions by assigning privileges for operating on some resources to roles instead of assigning them to users (Sandhu, 1998), (Ferraiolo and Kuhn, 1992). Users are then assigned to roles depending on their current position, responsibility and job requirements within the organisation. However, access control does not provide the necessary constructs to address privacy issues (Ashley *et al.,* 2002).

When explicitly addressing privacy protection, access control mechanisms are often augmented with the concept of purpose (Al-Fedaghi, 2007), (Ashley *et al.,* 2002), (Byun and Li, 2008), (W3C web site, 2003). The Hippocratic Database (HDB) (Agrawal *et al.,* 2002) has been proposed as a system that enforces privacy policies by technical means. The idea underlying the HDB is based on the Hippocratic Oath which aims to protect patient privacy. A HDB stores privacy metadata which specify privacy policies and privacy authorisations. *Privacy policies* define the privacy practices of the organisation (e.g., which data are collected, their intended purpose, retention period, etc.). *Privacy authorisations* capture the access controls that support the privacy policies by specifying usage purposes and authorised users for each attribute in the database. These authorisations are determined by comparing privacy policies against user preferences during data collection. During query processing, the HDB checks whether data are requested for the legitimate purposes. If the purpose for which date are requested matches the intended purpose, then the requested data are returned.

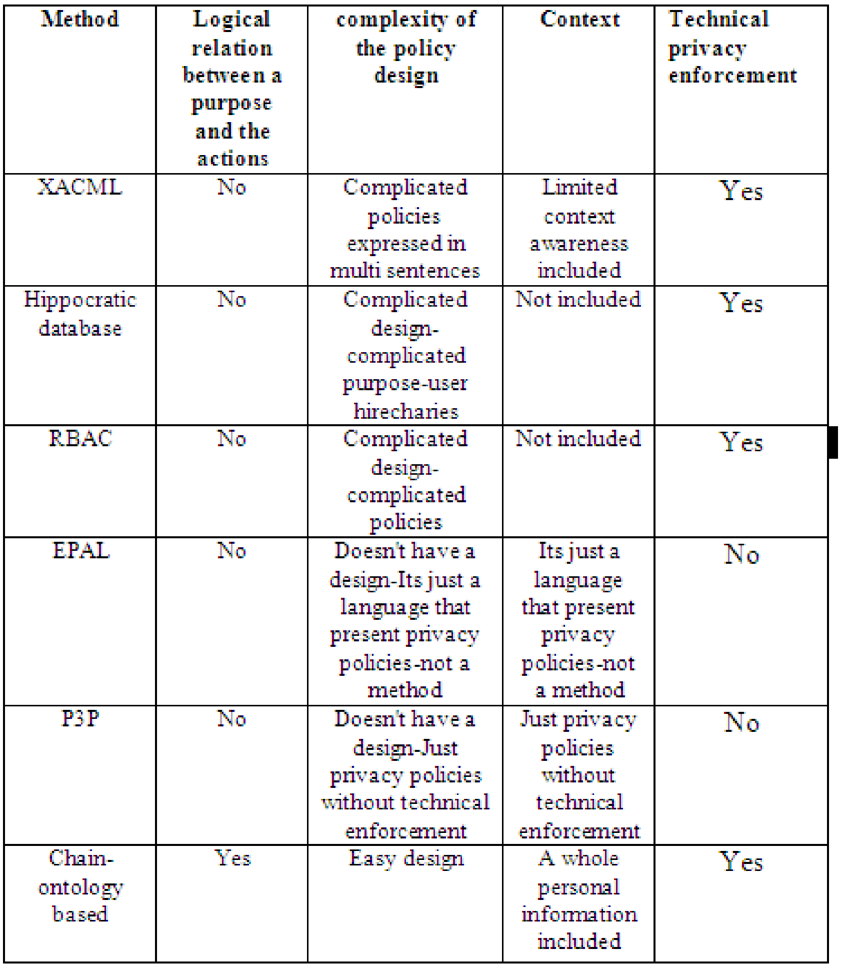
IBM developed the Enterprise Privacy Authorisation Language (EPAL) (W3C web site, 2003) to support organisations in keeping their privacy promises. EPAL provides enterprises with a way to formalise the exact privacy policy that shall be enforced within the enterprise. An EPAL policy consists of a vocabulary and a rule set. The vocabulary defines the scope of the policy. Rules are statements that specify which actions a user can or cannot perform on a certain object and for which purpose. When data are requested, the privacy management enforcement monitors ensure that only data accesses complying with the privacy policy are allowed.

A similar framework, eXtensible Access Control Markup Language (XACML) (OASIS, 2005), is proposed by OASIS. XACML is a structured language for expressing access control policies and a query-response protocol for the access of requests and decisions. To address privacy concerns, OASIS defines a profile of XACML for the specification of privacy policies (OASIS, 2005). In particular, the XACML's Privacy Profile defines standard variables to represent the purpose for which data was collected and the purpose for which data is requested, and shows how to create a constraint that requires these to be consistent. The advantage of using XACML is that the developer does not need to worry about multiple protocols to implement a security solution. Using one set of markup tags enables the developer to control security for a Web application. However, the problem that the database administrator could face is that the XACML profile needs to specify five main components to handle access decisions: Policy Enforcement Point (PEP), which is the interface of the whole environment to the outside world, Policy Administration Point (PAP), which is the policy repository, Policy Decision Point (PDP), which is the component where access request are evaluated against policies, Policy Information Point (PIP), which is the point where the necessary attributes for the policy evaluation are retrieved, and a context handler. Handling those five components may require a deep knowledge of the information system and application domain before the database administrator can start his work.

From the above it could be seen that the available methods have vital limitations. Thereby, there is a need for a new method that could overcome these problems.

Several methods and access control models have been proposed to protect sensitive information (e.g., Agrawal *et al*., 2002; Ashley *et al.,* 2002; Byun and Li, 2008; Sandhu, 1998. In this section the researcher compares those methods and models with the chain ontology-based method, especially, in terms of policy specification and management, as shown in Table 2.

Purpose is widely recognised as a fundamental concept for the specification of privacy policies (Guarda *et al.,* 2009). This has spurred several researchers to extend existing access control models by accommodating the concept of purpose into them. However, in most purpose-based frameworks like purpose-based access control, Hippocratic databases, EPAL, and the Privacy Profile of XACML, there is no logical relation between a purpose and the actions that are allowed for achieving that purpose. Consequently, one has to specify separately the actions that a user can perform on personal information and the purpose for which actions can be performed. Thereby, this approach can be error prone, leading to unauthorised access to personal information.



Method	Logical relation between a purpose and the actions	complexity of the policy design	Context	Technical privacy enforcement
XACML	No	Complicated policies expressed in multi sentences	Limited context awareness included	Yes
Hippocratic database	No	Complicated design-complicated purpose-user hirecharies	Not included	Yes
RBAC	No	Complicated design-complicated policies	Not included	Yes
EPAL	No	Doesn't have a design-Its just a language that present privacy policies-not a method	Its just a language that present privacy policies-not a method	No
P3P	No	Doesn't have a design-Just privacy policies without technical enforcement	Just privacy policies without technical enforcement	No
Chain-ontology based	Yes	Easy design	A whole personal information included	Yes

Table 2: Comparison between the different methods.

In contrast, the Chain method is based on the idea that each purpose, i.e. conceptual task or function, can be translated into chains of acts on personal information (Al-Fedaghi, 2007). The implicit assumption is that any piece of personal information does not need more than a limited number of acts to be dealt with, such as creating, storing, processing and disclosing. The advantage of the Chain method is that when the policy administrator allows a user to perform a chain of acts, he defines at the same time the purpose and the actions that the user can perform.

In addition, chains of actions can be used to design more lightweight and durable databases that safeguard personal information privacy. For instance, they make it possible to replace the complex Hippocratic database design, which usually include a huge number of purposes with chains of limited acts. Our experience at the International Clinic in Kuwait indicates that the Hippocratic database design is complicated even for expert database administrators; this is what makes it difficult to apply Hippocratic database systems in real applications. A study that compares the application of RBAC and the Chain method to a real healthcare application in Kuwait is presented in (Omran *et al.*, 2010a). This study shows that the number of tables (required to apply the authentication and constraints on the same database) using RBAC is larger than the number of tables used when the Chain method is in place. In particular, the number of administration tables was 4 using RBAC and 1 for the Chain method, the number of tables for doctors' interface is 7 for RBAC and 3 in the Chain method, the number of tables for the nurses' interface was 4 for RBAC compared to 1 in the Chain method and number of conditions required for authorisation is 3 compared to 1in the Chain method**.**

Additional advantages are obtained by complementing the Chain method with ontologies. In particular, the use of ontologies makes it possible to give a precise semantics to the concepts characterising an application domain and therefore to connect permissions to personal information as well as to the particular context in which the permission should be granted. For instance, the time and place in which the access is requested can influence the access decision; in emergency situations a physician is allowed to access patient's medical records. Although the context is necessary to specify fine-grained policies which allow individuals more control on their personal data, it is not included in many frameworks like Hippocratic databases and purpose-based access control.

Therefore, a need for a new method that overcomes the problems of previous systems and makes use of the advantage is needed. The new method should take care of the semantics as a vital component. Finding such a method is challenge described in the following chapters.

Chapter 3

Case Study: International Clinic (IC)

3.1 Background

As a natural conclusion to the discussion in the previous chapters, a concrete case study is needed where one can investigate a sophisticated data access requirement in a reasonably complicated setting that reflects much of what is typical of modern healthcare, and thus a hospital is a good example.

Here patients are seen to by a variety of departments and different types of staff where distinctions have to be made with respect to who can access what in what circumstance.

Hundreds of patients come in and out every day and where one needs to arrange all their incoming and outgoing information while keeping it available and secure.

Health care organisations must address a growing number of data management regulations and corporate governance requirements. If highly sensitive patient information is breached or lost, the organisation faces serious legal and financial consequences.

Therefore, for the purpose of this investigation the researcher has chosen to work in collaboration with one of the most successful hospitals and large companies in Kuwait which is the “International Clinic” hospital. This hospital has a sufficiently large operation of all the standard departments and the services they deliver and with a requisite large number of staff including consultants, doctors, nurses and medical support staff as well as admin staff.

3.1.1 Summary of International Clinic

International Health Services (IHS), a Kuwaiti Closed Shareholding Company was founded on June 15th, 1992 with the aim of improving and adding value to the private health sector in Kuwait and the Gulf region. Soon afterwards, International Clinic (IC), a western oriented private health care multidisciplinary facility was inaugurated as the first subsidiary of IHS to provide high quality comprehensive health services to all the local and expatriate population living in Kuwait.

Since then, IC has grown and flourished into a healthcare facility of international standards offering a full range of health care services in almost all medical, surgical, paediatric and obstetrics/gynaecology specialities within the IC Premises.

In March, 2005, IC opened its 24-hour, 7-day per week care centre with its separate male and female sections, each with beds and a private room for the individual care required. The 24-hour Care Centre was opened to treat men, women, and children who require a medical evaluation, treatment, and appropriate support services at any time of the day or night; and, if needed, the option to remain in a safe and comfortable setting for an extended period for care, under the close attention and observation provided by IC's team of doctors and nurses.

Some departments in IC have the most sophisticated and latest technology equipment including the laser machines in their dermatology department and the latest, MultiSlice CT Scanner in our diagnostic imaging department.

Their Support Services include highly sophisticated state-of-the-art medical technology found in such departments as Diagnostic Imaging, Laboratory, Physiotherapy, Plastic Surgery and Dermatology and the Pharmacy. They also offer high quality Dental, Periodontics, Implantology, and Orthodontic service in their Dental Unit.

Their comprehensive services continue to touch and enrich the lives of residents and visitors living in Kuwait.

The collaboration with IC started when my local supervisor Dr Shereef Abu Almaati, the head of the science division in the American University in Kuwait, sent me there to meet the manager. This helped in obtaining information about the processes in the system, how the IT department work and most important how the data access management system works there. In the next sections the policy, the process, the services and the different departments of the hospital will be presented.

3.2 Transformation to Electronic Health Records

Moving from paper records to electronic health records (EHRs) has been a challenge for many Kuwaiti hospitals. Implementation of this innovative technology will assist in providing better patient care by allowing for and providing more accurate and available patient information. The purposes of the collaboration with the IC was to assess the status of implementation of EHRs in outstanding sample of Kuwaiti hospitals; the factors that are associated with EHR implementation; and have a closer look at the benefits of, barriers to, and risks of, EHR implementation.

The key factor driving electronic health record (EHR) implementation was to improve clinical processes or workflow efficiency. Lack of adequate funding and resources was the major barrier to EHR implementation in many hospitals in Kuwait. Often this was the reason for the delay in the transition from paper based to electronic based records.

The reasons behind the transition to the EHR according to discussions with the manager and staff at the hospital can be summarised by the following points:

- EHR implementation will improve patient care by having better linkage to all caregivers, and reducing the need for file space, supplies, and workers for retrieval and filing of medical records;
- Improving workflow would be the major benefit of implementing the system;
- Reducing medical errors;
- Reducing cost and treatment time.

- Increasing revenue

The barriers to Implementing EHRs and the reason behind the slow transformation to the electronic records can be summarised into:

- Lack of adequate funding and resources.
- Lack of knowledge of database administrators to build reliable EHRs.
- Lack of support from medical staff and the need for training courses for the new system.
- Implementation and interpretation of rules to preserve health information privacy such as the Health Insurance Portability and Accountability Act (The Health Insurance Portability and Accountability Act of 1996, 1996).

The insistence of the administration there to build their own HER system has advanced the process effectively. To this end, they hired expert staff, especially the database administrator who has more than 20 years' experience in implementing such systems especially for hospitals. In addition, effective training was given to the staff at the hospital on the new investigated system. The system was designed to comply with all the hospital policies some of which are listed below:

- All patients have the right to receive considerate and appropriate medical care for their problem without regard to considerations such as race, colour, religion, national origin, disability or the source of payment for their care.

- All patients have the right to know the name and professional status of the physician who is responsible for their care.

- All patients have the right to obtain from their physicians current and understandable information about their clinical condition, treatment choices and potential benefits, risks and expected outcomes.

- All patients have the right to make informed choices about their health care. They have the right to receive information from their physicians that is necessary to give informed consent prior to the start of a medical procedure and/or treatment. They may accept or refuse treatment for any or all of the care offered.

- All patients have the right to personal privacy, safety, security and confidentiality.

- All patients have the right to have private and confidential medical records except as otherwise provided by law or upon the patient's written authorisation.

- All patients have the right to be treated respectfully by others and to be addressed by their proper names without undue familiarity, to be listened to when they have a question, and receive an appropriate response.

- All patients have the right to receive courteous attention from all personnel when they request help, with the understanding that other patients may have similar or more urgent needs.

- All patients have the right to communicate with any person or persons of their choice, including but not limited to physicians, administrators, nurses at any reasonable hour.

- All patients have the right to know about the policies and procedures related to patient care.

- Patients are responsible to follow instructions and recommendations provided by their physicians.

- Patients are expected to provide complete and accurate information about their medical history.

- Patients are expected to treat staff members and other patients with respect and consideration.

- Patients are expected to be fully responsible for the consequence of refusing treatment.

- Patients are expected to fulfil the financial obligations ensuing from their medical treatment.

As can be seen from the policies above, this is an attempt to balance the rights of the patients and the rights of the hospital, and it tries to manage the relationship between them. The policies above must to be taken into consideration when managing the data access system.

3.3 Process and services in the IC

This section introduces a typical scenario for healthcare provision in a hospital environment. Healthcare provision demonstrates a considerable amount of complexity in terms of processes and actors for providing medical treatments to patients. To execute the treatment plan, patient medical information needs to be accessed and shared by a number of professionals: physicians examine patients and can consult colleagues for a second opinion, nurses treat patients according to the diagnosis made by physicians; pharmacists have to provide the medicines prescribed by physicians, etc.

Our scenario describes the activities carried out by a hospital in Kuwait, the International Clinic (IC). According to this analysis in the IC and discussions with its staff, the researcher has produced the list of scenarios as shown in Figure 20; the complete list is presented in Table 1.

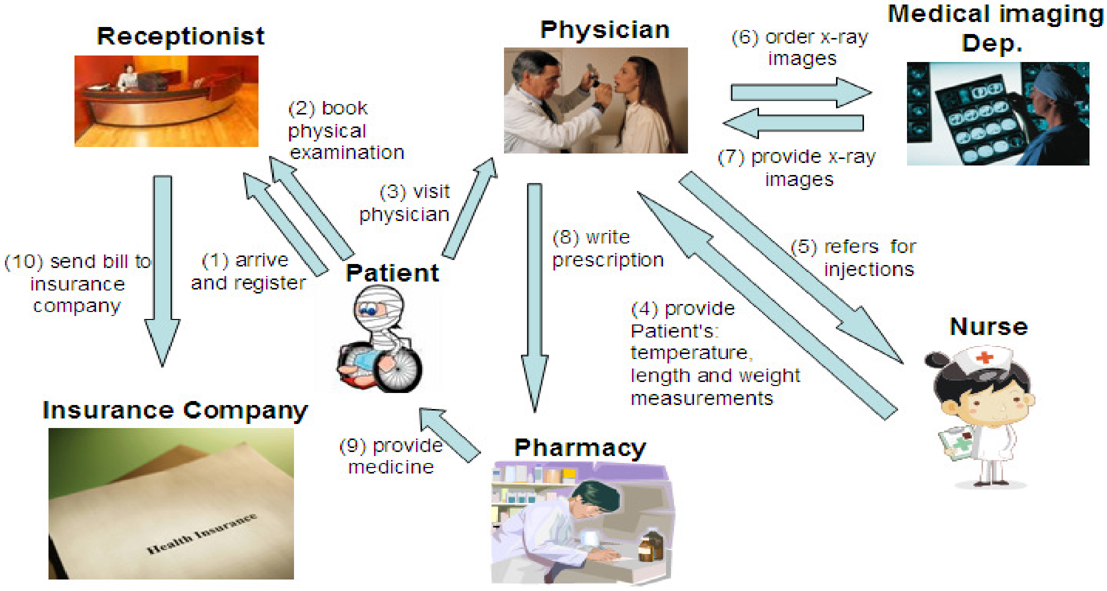


Figure 20: Healthcare provision in the International Clinic.

- Patient registration: When a patient visits the hospital for the first time, he needs to open a file at the reception desk. In order to register the patient with hospital, the receptionist asks him to provide his civil ID number and fill in a form with his personal information. In addition, other information may be required. For instance, if the patient has a health insurance, he has to present a valid insurance card upon registration; if he is an employee of a company that has credit billing facility with hospital, he is requested to show his company ID card and/or a referral letter from his company. When the necessary documentation has been presented, the receptionist creates a file for the patient in the hospital database system. Information about the patient (e.g., name, age, gender, disabilities, civil ID number, phone number and address) is entered into his file.

- **Appointment booking**: a patient can request an appointment by phone and/or in person. Appointments scheduled the same day and walk-in appointments are accommodated at the earliest possible time depending on the physician's schedule. The receptionist checks the visit schedule of doctors for free time slots. If a slot is available, the receptionist arranges the visit and enters the patient’s name and ID number as well as the time slot into the doctor’s schedule. The receptionist should not assign the same time slot of the same physician to two different patients. Therefore, the system should be designed in a way to simplify the booking process for the receptionist. Moreover, the system should retrieve the needed information for the booking (e.g., the name of the available physicians for a specific time slot) in a very short time so that patients do not need to wait for a long time at the receptionist’s desk or on the phone.

Figure 21: Esraa Omran (myself) looking at the paper based records at the IC.

- Doctor's visit: during the patient referral, the physician needs to review the notes he (or another physician) has made on the patient's health condition. He also needs to write his new notes as well as prescriptions for certain dosages of medicines. In addition, he may need to order X-Ray images and analyse images sent by the laboratory. He can also access information from the registration file or from the nursing database (e.g., temperature and weight, etc.).

Figure 22: Some Doctors from the IC.

- Nursing notes: Nurses should make information about patients available to the doctor in charge of providing medical care. In addition, nurses should access doctors' notes to gather information about a specific dosage of injections or drips. However, nurses should not be able to access the whole medical record and doctor's notes. They only need access to the part that is related to their job.

Figure 23: Some Nurses from the IC.

- Laboratory reports: Physicians can ask for specific laboratory tests from the laboratory using specific forms. The lab returns the results (e.g., blood test, radiology). Only the physician and the lab specialist have the right to access the lab report. Those images usually require a higher storing capacity; therefore, the greater the storing capacity is for the system, the better the retrieving and transferring of these images will be.

Figure 24: Radiology lab in the IC.

-Pharmacy: Physicians can write a prescription for the patient that specifies a dosage of a certain medicine from the pharmacy. The patient then goes to collect his medicine from the pharmacy. The pharmacist's information system should retrieve the information correctly without confusing the dosage or the name of the medicine or the patient.

-Insurance Coverage: If the patient has insurance, he should present his valid insurance card to the receptionists and sign a medical insurance payment declaration form. After the visit, he should sign a claim form and an invoice. Then the invoice with a special report is sent by the hospital to the insurance company. However, the disclosure of sensitive information (e.g., patient's health condition) to the insurance company could harm the patient's interest as the insurance company can take advantage of such information.

In order for the IC to provide excellent services, the IC has a reputable set of departments in a variety of specialisations such as:

- General Practitioner
- Cardiology
- Dermatology
- General Surgery
- Internal Medicine
- Obstetrics & Gynaecology
- Ophthalmology
- ENT
- Orthopaedics
- Paediatrics
- Paediatric surgery
- Plastic surgery
- Dental

A. General Practitioner Clinic

General Practitioners provide comprehensive primary healthcare services to patients of all ages. They provide high quality medical services to many acute and chronic medical conditions, and when needed, they refer you to the appropriate consultant.

B. Cardiology department

The Cardiology Clinic provides state of the art technology and compassionate care to patients with cardiovascular diseases.

C. Dermatology

The Dermatology Clinic at IC offers a full range of dermatological care for both common and rare problems of skin, hair, nails and mucous membranes.

D. Surgery department

Highly skilled surgeons address a large range of conditions that require surgical interventions. They provide comprehensive and conservative services to patients as needed and appropriate.

E. Internal Medicine

The Clinic provides diagnostic and therapeutic services for patients with medical disorders including major systems like cardiovascular, respiratory, and gastroenterology. In addition, this is the only haematology clinic in the private section in Kuwait that deals with blood disorders.

F. Obstetrics & Gynaecology

This clinic offers confidential counselling for and treatment of a broad range of important women's health issues and deals with the welfare of women of different age groups with special emphasis at the two ends of the spectrum: Adolescence and Menopause.

G. Ophthalmology

The IC ophthalmology Clinic deals with all problems of the eyes with special services for glaucoma, cataract, retinal eye problems and paediatric ophthalmology. The clinic is well equipped with retinal camera, perimeter and retinal laser Scanners.

H. E.N.T

The Otorhinolaryngology Clinic in the IC is a highly advanced well equipped unit for the management of diseases of the ear, nose and throat for adults and children. Surgical operations are done by our consultants in the clinic or in the hospitals using the latest techniques available such as lasers, radio waves and endoscopes.

I. Orthopaedics

Orthopaedics is a discipline concerned with preventing or correcting disorders of the body's basic framework, including the bones, joints, and muscles. An orthopaedic surgeon is a physician with many years training in the physical, medical, or surgical treatment and/or rehabilitation of the body's intricate mechanical system. Our Orthopaedic physicians may also have additional training in subspecialties such as sports medicine, joint replacement, knee, ankle, or shoulder reconstruction, foot problems and scoliosis or other disorders of the spine.

J. Physical Therapy & Rehabilitation service

IC provide services that help restore function, improve mobility, relieve pain, and prevent or limit permanent physical disabilities of patients suffering from injuries or disease. They restore, maintain, and promote overall fitness and health. Their patients include accident victims and individuals with disabling conditions such as low-back pain, arthritis, fractures, and sports injuries. Physical therapists also use electrical stimulation, hot packs or cold compresses, and ultrasound to relieve pain and reduce swelling. They may use traction or deep-tissue massage to relieve pain. Therapists also teach patients exercises to do at home to expedite their recovery.

K. Paediatrics

The IC Paediatricians provide comprehensive care for a wide range of childhood illnesses and injuries with an emphasis on health promotion and disease prevention. They play an important role in enhancing the physical, mental, and emotional growth and development of new-born babies as well as adolescents.

L. Paediatric Surgery

The IC skilled paediatric surgeons perform a broad range of surgical procedures and services and they address a large scope of conditions that require surgical interventions. They provide comprehensive and conservative services to patients as needed and appropriate. The Paediatric Surgery & Laparoscopy Clinic provides surgical treatment for patient between 0 and 14 years of age.

M. Plastic Surgery

The IC surgeons perform a wide range of procedures encompassing both Cosmetic and Reconstructive Surgery. The goal is to help the patients to look and feel better by utilising the most advanced plastic surgery techniques and providing them with their utmost personal care and attention.

N. Dental Clinic

The dental clinic deals with a wide variety of dental services covering general dentistry, oral surgery, dental implantology and conservative dentistry.

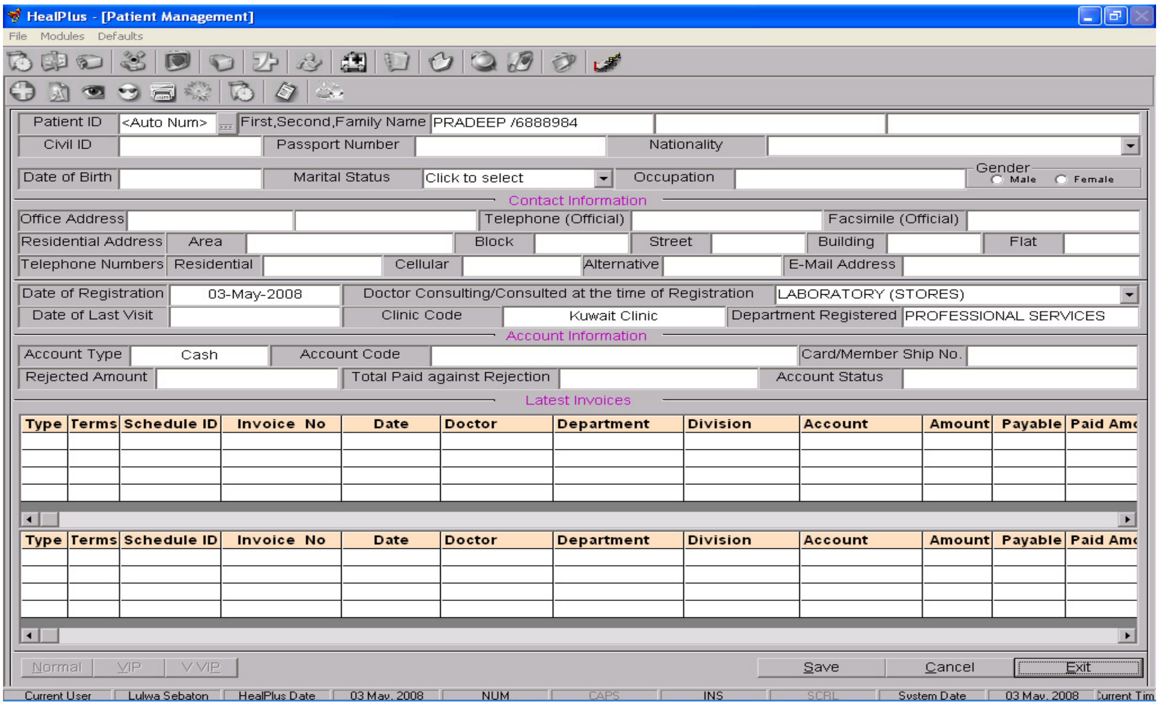
The IC provides all the above services to reach its vision which is to be the private healthcare facility recognised for setting the standards for excellence and responsiveness in the Gulf region. They aim at being desired and chosen by the masses for the provision of high quality healthcare services due to the hospital excellent service, modern facility and latest technology equipment, skilled and compassionate staff and very friendly home environment.

ICs unswerving mission entails provision of the highest quality comprehensive healthcare services in a caring, friendly, efficient and cost effective manner that represents value to their patients, while at the same time sustaining their needs and expectations for the wellbeing of the community.

The IC wants to reach their mission while keeping their values which is the commitment to maintaining the highest healthcare standards and levels of patient care. They value each patient and their right to receive professional, efficient, ethical, and quality service from all IC employees. They also value each person's (patient or visitor) time and need to be evaluated and treated in a safe, clean, well-equipped, and well-managed setting where patient satisfaction matters and continuous improvement is recognised as a priority.

The IC is committed to establishing goals and objectives that are consistent with the needs and expectations of consumers while keeping pace with the growing trends in healthcare technology and improvement. The IC is a dynamic organisation continuously striving to be the best at what they do and how they meet their customer and his family's healthcare needs. To that end, IC values the importance of remaining accessible at all times, amicable and professional, and able to provide the best of what the customer needs to live a safe and healthy life.

In order to handle these processes, the IT department in the hospital has developed a system called HealthPlus. The system has different interfaces for different users, such as that shown Figure 25 for the receptionist and the interface in Figure 26 for the doctor.



Patient ID <Auto Num> First,Second,Family Name PRADEEP /6888984
Civil ID Passport Number Nationality
Date of Birth Marital Status Click to select Occupation Gender ○ Male ○ Female

Contact Information

Office Address Telephone (Official) Facsimile (Official)
Residential Address Area Block Street Building Flat
Telephone Numbers Residential Cellular Alternative E-Mail Address

Date of Registration 03-May-2008 Doctor Consulting/Consulted at the time of Registration LABORATORY (STORES)
Date of Last Visit Clinic Code Kuwait Clinic Department Registered PROFESSIONAL SERVICES

Account Information

Account Type Cash Account Code Card/Member Ship No.
Rejected Amount Total Paid against Rejection Account Status

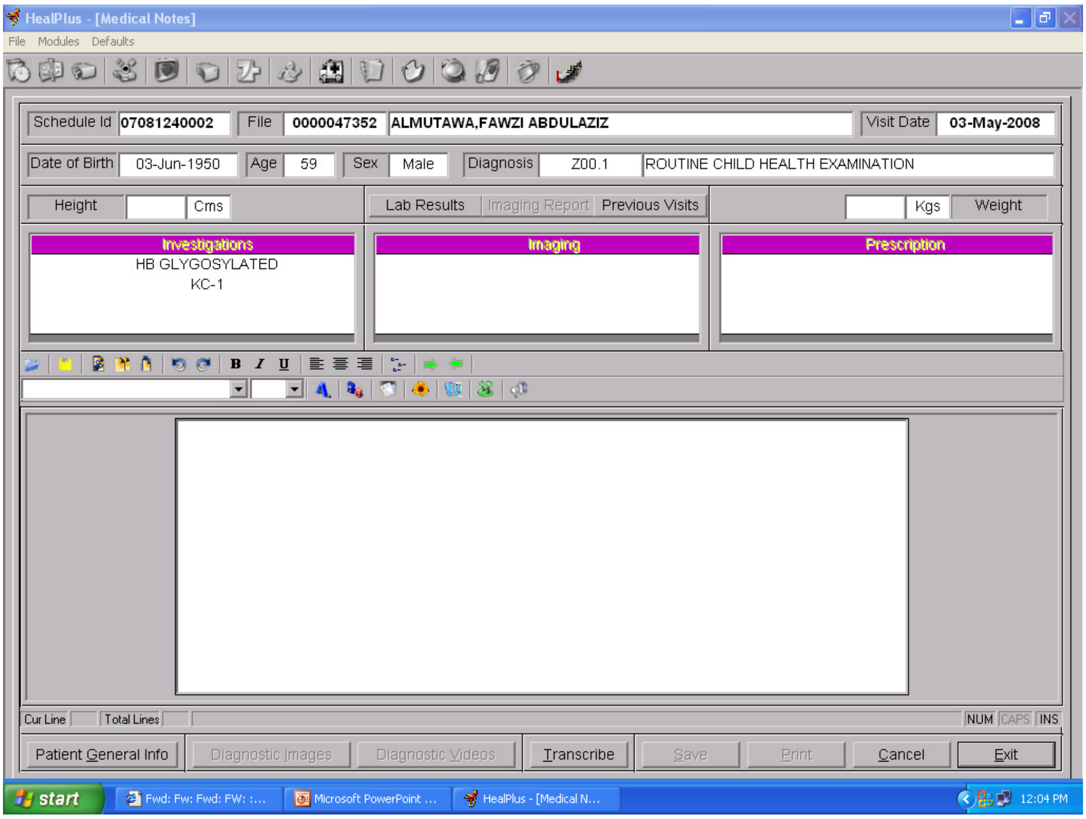
Latest Invoices

Type	Terms	Schedule ID	Invoice No	Date	Doctor	Department	Division	Account	Amount	Payable	Paid Amo

Type	Terms	Schedule ID	Invoice No	Date	Doctor	Department	Division	Account	Amount	Payable	Paid Amo

Current User Lulwa Sebaton HealPlus Date 03 May, 2008 NUM CAPS INS SCRL System Date 03 May, 2008 Current Tim

Figure 25: HealthPlus interface for receptionists.



Schedule Id 07081240002 File 0000047352 ALMUTAWA,FAWZI ABDULAZIZ Visit Date 03-May-2008
Date of Birth 03-Jun-1950 Age 59 Sex Male Diagnosis Z00.1 ROUTINE CHILD HEALTH EXAMINATION
Height Cms Kgs Weight

Investigations
HB GLYGOSYLATED
KC-1

Imaging

Prescription

Cur Line Total Lines NUM CAPS INS

Figure 26: HealthPlus interface for doctors.

The system works based on the role based access (RBAC) principles, where each set of users (Doctors, nurses, receptionists) have their own access policies and can view almost the same set of data. For example all doctors can search for the information of any patient and edit them regardless of the context. The system uses a Java GUI that is connected to a huge database, where function icons are enabled for each set of users so that they can process the authorised data.

3.3.1 Hospital scenarios

The goal of our discussions and observations in the hospital was to develop and provide flexible and adaptable coordination strategies that can be applied to real world hospital scenarios. This special hospital-scheduling scenario is typical for a computer-based simulation model, where different strategies and their consequences can be evaluated. The treatment of hospital patients requires complex coordination of many autonomous organisational entities. Special requirements to coordination emerge at the intersection between wards that have to organise patients' appointments, and functional units that provide medical services for treatment and examination. The capacities of these functional units are limited by their resources (medical devices, personnel, rooms). Coordination strategies can vary from simple queues to a precise anticipatory scheduling that also considers shifting of appointments. In particular, the researcher wants to evaluate the benefit of complex distributed coordination strategies using negotiation to meet all the actors' interests and goals. These interests may conflict, and can be formally expressed by appraisals for appointments and local restrictions that are valid for a certain entity. For example, a high load might be desirable by a functional unit, whereas patients prefer short waiting periods. To evaluate the effects of different strategies the researcher focuses on patient processes of selected standardised medical guidelines. These are used to create realistic demands in the scenario. In addition to these standardised guidelines and general medical knowledge there are parts of the scenario which are highly individual and typical for the specific considered hospital. This includes the structure of the organisation and the preferences and restrictions of the different actors. All this was taken into account in the development of the ontology that is presented in the next chapter. Thus, various scenarios can be described and the researcher can achieve a high grade of flexibility without having to change the ontology. Below is a table that contains all the possible scenarios that the researcher has observed and collected during more than two years of collaboration with the IC hospital:

Number	**Possible scenarios for group of users**	**Name of scenario**	**Description**	**Database entities**	**Data access privileges**
1.1	**Receptionist/ Administrator**	New patient registration	registering patients for first time, taking basic details	Patient record	Creation of patient record but read-edit only for demographic part
1.2		Booking appointments	patient booking for his next appointment	Patient record Appointments	read-edit access to demographic part and no access to other parts read-edit-creation and deletion access to appointment records
1.3		Visit for appointment	patient arriving to see doctor with existing appointment	Patient record Appointments	read-write access to demographic part and no access to other parts read-write-creation and deletion access to appointment records
1.4		Emergency case	patient arriving in emergency case	Patient record A&E Waiting List	read-write access to demographic part and no access to other parts read-write-creation access to appointment records
1.5		Billing	Preparing and managing bills with insurance company	Patient record Billing records	read-access to demographic part and no access to other parts

					read-edit-creation and deletion access to billing records
1.6		Managing Patients	Managing patients for actions required by healthcare professionals	Patient record Referral records Appointments All Waiting Lists	Read access to demographic part and pharmacy records, but no access to other parts read access to referral records/doctors' letters read-write-creation and deletion access to appointment records read-write-creation and deletion access to waiting lists records
2.1	**Doctors/ Consultants**	Routine Patient Consultation	Seeing patients who have appointment or are on lists to be seen	Patient record Referral records Appointments	Read-write access to full patient record read access to referral records/doctors' letters read-write access to appointment records
2.2		Outgoing referral	Patient to be referred to consultant/nurse or radiology	Patient record Referral records Appointments All Waiting Lists	Read-write access to full patient record Read-write access to referral records/doctors' letters Read-write access to appointment records Read-write access to waiting lists records
2.3		Incoming referral	Patient has been referred to doctor by other healthcare professionals	Patient record Referral records Appointments All Waiting Lists	Read access to demographic and prescriptions part of patient record read access to referral records/doctors' letters read access to appointment records read access to waiting lists records
2.4		Issuing Prescriptions	Prescription to be issued to patient	Patient record Prescription Record	Read-write access to full patient record Read-write access to prescriptions

2.5		Emergency case	Patient coming in emergency case without appointment	Patient record A&E Waiting List	Read-write access to full patient record Read-write access to waiting list
2.6		Waiting list consultation	Patient without appointment but not in emergency case	Patient record All Waiting Lists	Read-write access to full patient record read-write access to waiting lists records
3.1	**Nurse**	Nurse Consultation	Patient initiated service requests with appointment	Patient record Appointments	read-edit access to demographic and nursing part and no access to other parts read-edit-creation and deletion access to appointment records
3.2		Incoming Referral	Deal with patient according to the doctor's instructions	Patient record Referral records Appointments All Waiting Lists	read-edit access to demographic and nursing part and no access to other parts Read access to referral records/doctors' letters read-edit-creation and deletion access to appointment records read access to waiting lists records
3.3		Emergency Assessment	Patient coming in emergency case without appointment	Patient record A&E Waiting List	Read-write access to full patient record without clinical record read-write access to waiting lists records
4.1	**Manager and Senior Administrator**	Compliance auditing	A patient is complaining about sensitive information being disclosed. And he asked the hospital to know who is behind this disclosure.	Patient record Referral records Appointments All Waiting Lists Data Access Logs	read-edit access to demographic part and no access to other parts Read-edit access to referral records read-edit-creation and deletion access to appointment records read-write access to waiting lists records Read access to data access logs (where

					available)
4.2		Managing Healthcare	Managing patients and healthcare provision	Patient record Referral records Appointments All Waiting Lists User Accounts	Read access to demographic part but no access to other parts read access to referral records but not doctors' letters read-write-creation and deletion access to appointment records read-write-creation and deletion access to waiting lists records read-write-creation and deletion access to user accounts
5.1	**Insurance company**	Billing	Receiving bills for treatment of patient	Billing records	read access to billing records
6.1	**Radiology lab**	Radiology referral	Patient being referred to radiology by doctor	Referral records Appointments All Waiting Lists	Read-write access to referral records/doctors' letters Read-write access to appointment records Read-write access to waiting lists records
7.1	**Pharmacist**	Dispensing Prescriptions	Dispensing Medication according to Doctor's instructions	Patient record Prescription Record	Read access to basic details of name address and DOB Read-write access to prescriptions
8.1	**Laboratory**	Laboratory referral	Patient being referred to laboratory for blood analysis	Patient record Referral records	Read access to basic information (name, address, DOB) Read access to referral records/doctors' letters

Table3: Aggregated scenarios for health care.

Collecting these scenarios is of crucial importance for the ontology and database design in the next chapter. As shown in Chapter 2, no one to the researcher's knowledge has made this type of research although its importance has been recognised, as in (Becker *et al.,* 2002). In that paper, the authors have mentioned the importance of collecting such scenarios from a real hospital in order to build a reliable ontology for health information.

The next chapter illustrates how these scenarios are used in the design of this system.

3.4 Discussion

The above sections in this chapter have shown the importance of ethics and the hospital policy in the process of the daily work at the hospital and accordingly the data access system that manages the flow of the patient's sensitive information and control the application of the hospital ethics, rules and policies.

Table 3 shows that there are at least 21 different and sensitive situations in the hospital. Each one of them has different actors with different situations with different pieces of information. Some of them (such as scenario 1.4, 2.5 and 3.3) are very sensitive and need the right information at the right time or otherwise a human being could pay with his life as a result of any error or lateness. Some of them need a highly flexible system to carry the on-going changes (such as scenario 2.6). Openness is also crucial in cases such as scenario 2.4 (In order to issue an updated prescription). In all scenarios (from 1.1-8.1) accuracy and availability are extremely important for a reliable health care system.

The objective of the collaboration with the IC hospital was:

1. To study the existing hospital daily process and their information system.

2. To identify the shortcomings, if any, in the existing Hospital Information System to obey the requirements of the daily process (hospital work chain).
3. To suggest the necessary steps to improve the existing Hospital Information System and data access management.

The researcher has noticed through her observation in the hospital that staff in general and managers in particular would be able to access information if they had the competent skills to drive the system, as the system will retrieve all authorised information once the user has gained access with his username and password. This is reinforced by the fact that those people with better computer skills were able to use the system optimally to get the information they needed. Which means that they would need to search by themselves manually through the data they would have and would depends on their skill to extract the required information and sometimes by trial and error. This demonstrates vividly that in order for managers to retrieve information, computing skills are crucial.

These facts really highlight the importance of having a reliable data access management system to control the processes in the hospitals and manage the information flow, as doing this could highly affect the life of millions of people. This system should have the ability to retrieve exact information without excess and to be correct (without error) and in a short time.

In the next chapter, the hypothesis and the methodology needed for investigating such a system will be discussed.

Chapter 4 Hypothesis and Methodology

Albert Einstein said:
“If we knew what it was we were doing, it would not be called research, would it?”

The problem of the unavailability of reliable data access management has been raised in previous chapters. A chosen example has been given in Chapter 4 to show the dimensions of the problem in the healthcare domain.

As in healthcare, the processing of sensitive data is used in entering and retrieving the information to and from the system.

The overall aim of this thesis is to develop access control models that are “context-aware”, “more dynamic” and overall better suited to the needs of healthcare. However, through the literature review it became clear that the information needed to design such models is very difficult to find. As a result, detailed research has been carried to get this information which was scattered and incomplete. This investigation led to collaboration with the IBM company (these discussions led to a study agreement and a number of publications and a patent proposal for the developed system), Trento University, Eindhoven University (these discussions also led to number of publications) and Madrid University (these discussions also produced publications). These collaborations were essential to decide on the basics of the experiments designed and to get help in carrying them out.

As presented at the end of the previous chapter, each of the leading solutions from the literature such as Hippocratic (Agrawal *et al.,* 2002), Role Based Access Control (RBAC) (Sandhu, 1998) and the classical Chain (Al-Fedaghi, 2007) has their own problems such as the complexity of design and implementation processes, the need for expert staff for the design and implementation processes. Therefore these methods do not provide a reliable solution to the key problems of data access management discussed in previous chapters.

In addition to the problems described in the previous chapters, a further problem has been found:

- Finding experts to participate in large numbers to test the developed prototype system.

The classical chain method seems to provide a much simpler solution for those two problems with the idea of limited acts. This doesn't require so much special skills from the database designer in order to create complex hierarchy of authorised users and functions they can perform, such as that of the Hippocratic database or of RBAC. But the classical chain method has its own problem. It needs to have a way to decide for any given request, which chains to be allowed for which user. This decision should be based more on an assessment of the situation in which access requests are made. For this reason it would seem plausible that combining the chain based method with a domain model would provide a basis for deciding whether a given chain can be executed by a given user in the given context. In the following sections the problems found through research will be presented and how a solution was reached. The research question and hypothesis will also be highlighted.

So in order to provide a scientific focus to the investigation, the researcher starts by forming a research question and consequent hypothesis as well as deciding on the methodology required to assess the proposed approach via the hypothesis and develop a valid experiment.

Therefore, in this chapter, the researcher will describe the proposed Hypothesis and methodology for designing a reliable data access management system that preserves privacy while keeping the simplicity of design. This methodology is claimed to satisfy the requirements listed in section 4.2 and thus serves as a solution for the main problem presented in Chapters 1 and 2, i.e. The increasing amount of personal information that needs to be protected from unauthorised users, while relying on expert database designers to develop a database system through a complex process.

This chapter is organised as follows:

In section 4.1 the research question and hypothesis are presented. Section 4.2 describes the criteria for success for this project. Then, section 4.3 explains the contribution to knowledge. In section 4.4 outlines the methodology of the thesis. Finally, section 5.5 is an overall discussion of the chapter and draws its conclusions.

4.1 Research Question and Hypothesis

In order to appropriately protect sensitive personal information, such as in the healthcare domain, there is a need for workable solutions that can be handled more easily and thus reliably. Consequently, system programmers need to implement privacy rules in information systems and to improve the capabilities of information systems. The purpose of this is to very stringently apply data protection and privacy protection when information and functionality is accessed by users. It is time now to revisit the research question and turn it into a hypothesis for which the researcher can derive a meaningful set of criteria for success and a valid set of experiments.

The outstanding problem that has been concluded from Chapter 1 is “the need for an efficient method that can enforce privacy technically.” Consequently, and on the basis of the results of the literature review, the research question can be narrowed down to: whether a context sensitive approach to enable data access management of sensitive information, which is also less complex than the traditional access methods, where seven simple to apply acts replace complicate polices of RBAC and Hippocratic methods. In addition, semantics and ontology adds context sensitivity and accuracy to the process of data access management.

So the research question can be summarised as:

RQ: "Can a semantic approach based on simple limited acts of Chain method improve the data access management of personal information while preserving privacy? "

From the research question, hypotheses should be drawn to highlight the purpose of this dissertation. Accordingly the criteria of success and evaluation experiments need to be delivered from these hypotheses also.

The hypotheses, according to the research question, have to suggest semantics and Chains to be the basics of the proposed solution. This proposed solution also needs to overcome the shortcomings of the currently available methods in the literature in order to deliver better data access management.

Given the research question presented above the researcher now proposes the following hypotheses:

The improvement of the data access is defined as:

- *H1: To simplify the configuration of a data access management system for a given database through a reduction of complexity (complexity means number of required: tables, SQL statements and constraints).*
- *H2: To increase the precision (more focused results) with which the algorithm discerns between legitimate and illegitimate access. As precision will limit the unnecessary data disclosure by focusing only on the required one.*

In summary, there is a problem of correctly setting up and maintaining access privileges for a dynamic set and large numbers of users. In addition, there is a need for a context sensitive approach that can respond to changes in circumstances to adapt to what records users can have access to and at what level.

The researcher has proposed a solution to this problem using semantics and the Chain method. This system works through reducing complexity in applying data access policies during user administration and subsequently improving the accuracy with which access attempts are assessed.

The methodological principles will be applied to the healthcare domain to test its reliability in managing access to the sensitive data processed there.

4.2 Criteria for success

In this thesis, the main issue is to enable system programmers to successfully and easily apply relevant privacy protection policies. Reliably measuring subjective concepts such as ease is fraught with problems. Therefore there is a need to measure more indirect indicators to provide more readily measurable data to allow us to conclude that the design of privacy policy application has been simplified. According to our hypothesis, the new method should deliver better access management. Evaluating this progress in data access management is not a direct issue. Therefore the researcher proposes to summarise the criteria of success in the following points:

- Reducing the number of steps required to carry out essential tasks. The measurements of this factor are:
    - Reduction in the number of SQL statements required to build a specific data access scenario.

  - Reduction in the number of tables required for a specific data access scenario.
  - Reduction in the number of constraints required to apply a specific data access scenario.

- Measuring the time required to assess access requests and retrieve information. This is carried out by giving the database administrators and hospital staff a set of scenarios and measuring the time required to retrieve data for each scenario. The measuring process is done number of times and the average of time is taken.

- Evaluating the accuracy (consistency-exact result) and precision (more focused results) of the retrieved information (Teufel, 2006). This is done by comparing the retrieved data with the data that should be retrieved theoretically for each scenario.

In this thesis, accuracy is defined as the percentage of records correctly classified as returned or not returned. Precision is defined as the percentage of retrieved results that are relevant to the query (Han and Kamber, 2006).

If the developed prototype that is based on the investigated method improves upon the traditional methods in each of the above three criteria, the hypothesis will be accepted. These criteria have been chosen according to the discussions with IBM and Trento, Madrid and Eindhoven Universities. In addition to the observations, discussions and surveys have taken place the IC hospital.

4.3 Contribution to Knowledge

A unique aspect of the developed approach is that it focuses on developing the appropriate privacy preserving measures early in the design process while taking into account the different types of malicious users and the effect each type might have. As in Sawma (2002) malicious users can be grouped into three categories: Crackers, intruders and insiders. A cracker is someone who breaks into systems for nefarious purposes from, and he is a person from outside the system. An intruder is someone who gains access into systems by force, and he is a person from inside the system. Finally, an insider is a person who is in a position of power or has access to system confidential information. The insider could access and disclose information because he has authorisation to access information (more than required information to fulfil his job functions). In this thesis the researcher is developing a method that preserves the privacy from attacks of all of the above types.

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

1. Developing a new Chain method (ChBAC) based on the classical chain method.

2. Integrating semantics into the system by developing an ontology, and enabling this single ontology to be attached to the method, making modification much easier.

3. Reduction of complexity in setting up and maintaining access privileges.

4. Developing an original dynamic approach that responds to circumstance to grant access only to legitimate records

5. Provision of extensible personal information ontology and its use as a classification layer in database access management.

6. Identifying the major challenges that are to be addressed in the design of chain based solutions.

7. Translating a large set of typical hospital procedures into one of Al-Fadeghi's 7 acts.

The classical chain method that has been suggested by (Al-Fedaghi, 2007) has never been implemented nor tested in any hypothetical nor real enterprise. As mentioned before, the specifications given in (Al-Fedaghi, 2007) do not help alone in implementing the Chain method in reality without a semantic study that shows the definitions of each act. In addition, it has never been designed to solve any particular problem like for example the problem of managing access to personal information in healthcare without loss of privacy.

The second original contribution will be a dynamic data access management approach. This new approach decides on access attempts not statically as does the role based access method but dynamically, based in the situation in which a request is made.

Better privacy protection tools are required to manage personal information and determine what personal information really needs to be or can be collected. Most importantly methodologies and guidelines for implementing and integrating them into information systems are required. This is because system developers and operators have had little guidance on how to implement and comply with privacy guidelines and rules. Further, there have been few analytical or systematic attempts to understand the relationship between privacy and technology. Therefore, Information Systems need a comprehensive systems-wide approach to information privacy. This is the aim of this thesis. That is, to develop a new database management access method that integrates the Chain idea with a newly developed personal information ontology to answer the question of privacy preserving through a dynamic, durable and easy to implement method

4.4 Methodology

In this section, the researcher provides a detailed description of our proposed methodology for deriving and applying the Chain ontology based to the existing data access management principles. To verify the hypothesis that was reached earlier in this chapter, it is necessary to define a methodology that tests the semantic system for the main two criteria of success: reduction of design complexity and precision of the data retrieved.

In the following section, the choice of research methodology and the purpose behind this selection will be presented.

4.4.1 Choice of research methodology

In the previous section the research question and hypothesis have been presented together with the criteria for success. The hypothesis, that will be followed to improve the privacy protection methods, has been discussed. In some fields of study as is the case for this dissertation - it is not easy to measure each criterion independently. Nevertheless, it is preferable to measure each criterion separately to see its effect on the process and how it has improved the performance of the system.

Often one compares against a scientific control or traditional treatment that acts as baseline, since a reference point is needed to measure improvement against state of the art approaches. Thus the research methodology is based on a comparative study where the researcher compares the proposed approach against existing, widely used, approaches.

- **The problem**: Current implementation technologies are too complex to be implemented therefore the researcher will investigate newer technologies that can reduce complexity and thus enhance the overall system performance in terms of development and maintenance. Although the fundamental concepts of roles in the traditional methods (Hippocratic, RBAC and TBAC) are common knowledge, the capability to formalise model specifications needed to implement these models is beyond the knowledge base of existing staff in many software companies. The lack of knowledge and staff expertise in the area of RBAC increases the uncertainty of both the technical feasibility of developing successful RBAC-enabled products and the development cost and time-frame.

- **Type of problem:** An applied research problem where the domain is in privacy, security and database. The problem can be summarised as filtering authorised and unauthorised users and then classifying authorised users using an ontology into groups of users that can access certain groups of data. As new users access the system and old users no longer have the authority to use the system, the importance of having a reliable, flexible methodology that can deal with all the expected and unexpected problems becomes essential. Also, the system needs to be context sensitive to enable this flexibility.

- **The Solution:** A new methodology is required that incorporates the Chain method along with an ontology for modelling privacy policies in a database system. The Chain method provides a simple design solution with many fewer conditions, policies and hierarchies, while the ontology gives the dynamicity (the ability to update the database according to the given constraints from the ontology GUI) that is not available in current traditional database access management methods.

- **Type of experiment:** Comparative. In a comparative study, two (or more) cases, specimens or events are examined, often in the form of a table where a column is reserved for each case. On the basis of the aims of this study, a decision has been made as to which are the interesting aspects, properties or attributes that will have to be noted and recorded for each of the cases. The evaluated method will be compared with three other state of the art methods in the literature: RBAC, TBAC and Classical Chain method.

This thesis follows the experimental design approach. In an experimental design, the researcher actively tries to change the situation, circumstances, or experience of participants (manipulation), which may lead to a change in behaviour or outcomes for the participants of the study. The participants are ideally randomly assigned to different conditions, and variables of interest are measured. In the case of this thesis, the situations that were given to the users are the scenarios presented in Table3 in Chapter 3.

In a good experimental design, some things are of great importance. First of all, it is necessary to think of the best way to implement the variables that will be measured. Therefore, it is important to consider how the variable(s) will be measured, as well as which methods would be most appropriate to answer the research question. In addition, statistical analysis of the collected results has to be taken into account. Thus, the researcher should consider what the expectations of the study are as well as how to analyse this outcome. Finally, in an experimental design the researcher must think of the practical limitations including the availability of participants as well as how representative the participants are to the target population. It is important to consider each of these factors before beginning the experiment (Adèr *et al* 2008).

- **Who will carry out the experiments?:** As there are two types of experiments, two types of users will carry out the experiments:
  - For the simplicity of the design: five expert database administrators from the three largest institutions in Kuwait will go through this set of experiments.
  - A real hospital in Kuwait has provided us with the possibility of using their system design to compare with. In addition their employees will use our system for the usability comparison experiments.

- **Desired results from the experiments**: To demonstrate that our methodology can be applied in real situations and addresses privacy problems, is easier to design and implement than other existing systems, and can retrieve information quickly in real time.

The points presented in this section shape the experiments that will be conducted in order to evaluate the investigated method performance.

4.4.2 Details of the methodology and the proposed approach

The thesis methodology, depicted in Figure 27 can be divided into two functional phases. In phase 1, reliable data access management features are selected and a privacy-preserving–oriented design model is derived and verified. In phase 2, the derived model from phase 1 is instantiated to fit the healthcare sector.

The following UML figure shows possible relationships among Privacy violation attack, privacy feature, and access management method and attack enabler.

The following should be highlighted:

- One or more privacy violation attacks (whether dependent on or independent of other attacks) might be applicable to one privacy feature. This means the relationship is one feature to one or more attacks.
- One privacy violation attack can rely on one attack enabler to succeed.
- A reliable data access management method is required to disable an attack enabler.
- Residual vulnerabilities may remain after applying the data access management functions.
- In addition, residual vulnerabilities might enable one or more privacy violation attacks to sensitive data.

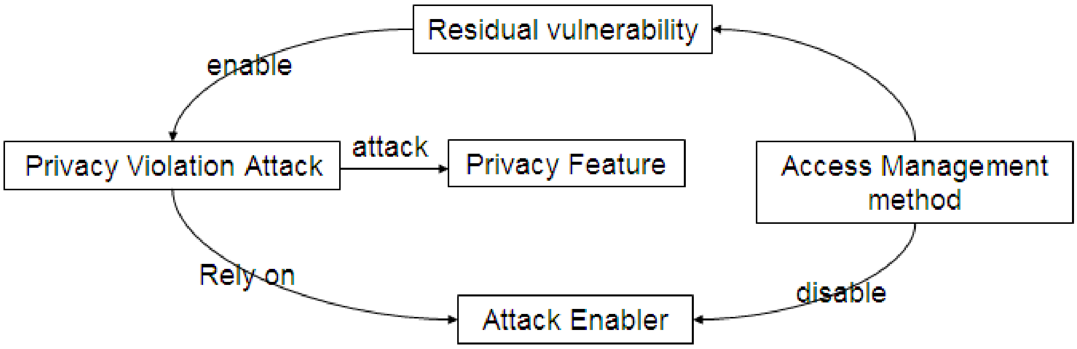


Figure 27: A UML class diagram of the relationships between Privacy violation attack, privacy feature, Access management method, and attack enabler.

The researcher has reviewed the literature in order to find the privacy features required for our project. The researcher selected four features of the NIST security services model (NIST, 2012) in order to focus on them and try to develop them which are widely seen as the common features required to ensure security. They are namely: authentication, authorisation, access-control enforcement, and transaction privacy. Types of intruders which can attack and affect these services such as Misfeasor, Masquerades and Clandestine users are explained in details in chapter 9 in Stalling (2011).

After drawing the main features of the method, the researcher began designing the system that will use this method. In order to do this, two years of observations and taking notes have taken place in a hospital in Kuwait. The researcher collected all the possible scenarios there and saw how they manage access to their huge database. The results of this phase led to the use of the chain method and semantics as a suggested solution from the literature, and the set of scenarios that has been presented in Table 3 in Chapter 3.

4.4.3 Overview of Proposed Experiments

In this thesis the issue is to enable system programmers to successfully and easily apply relevant privacy protection policies. Reliably measuring subjective uneasy criteria to measure such as ease is fraught with problems. Therefore there is a need to measure more indirect indicators (like for example the number of tables has indirect reason for its increase which is the number of required constraints) to provide more readily measurable data to allow us to conclude that the design of privacy policy application has been simplified.

The evaluation of the performance of the developed method will be through comparative experiments and analysis for the performance of the investigated system over traditional system using a set of standard scenarios. The criteria of evaluation for the experiments are:

1- **Design Simplicity** (the number of specifications required). This is related to the part of the hypothesis that is concerned with simplifying the configuration of a data access management system for a given database through a reduction of complexity. This will be evaluated in two levels:

-Database level: Comparison between the classical Chain method and the RBAC method. This set of experiments will evaluate the required number of: SQL commands, tables and constraints to construct each method for each specific scenario from Table 3. The respondents needed to write down SQL statements, create tables and constraints to apply the principles of each method for each specific scenario. The respondents need to be experts in database administration.

-Semantic level: Comparison between the Chain, RBAC and TBAC methods. These methods have been chosen for comparison as all of them work at the same level the application level. On the other hand, the Hippocratic database method for example works at the data level and would require redesign of the database. The experiments were to evaluate the number of OWL statements and constraints needed to develop each method in OWL. The experiments also measure the number of tables that need to be accessed in the determination of an access or disclosure decision. The respondents needed to write down OWL statements and constraints to apply the principles of each method for each specific scenario. The respondents needed to be experts in the OWL language.

The desired results from this set of experiments: To have fewer specifications and design parameters which simplifies the design and the implementation of the method.

2- **The Precision measurements.** This set of experiments is related to the part of the hypothesis that is concerned with increasing the precision (i.e. more focused results) with which the algorithm discerns legitimate from illegitimate access.

This set of experiments is not easy to measure and needed to be designed carefully to measure criteria that affect the precision (more focused results) of the data retrieved from each method such as: accuracy, context sensitivity and correctness. The methods chosen for evaluation for this set of experiments were the: Developed Chain method, the classical Chain method and RBAC. The respondents for this set needed to complete questionnaires that have scenarios from Table 3 to be fulfilled. The respondents in this case were staff from the Kuwaiti hospital.

The results expected from this set of experiments: To retrieve more focused data instead of having access data that are not required and could make harm in case of disclosure. In addition the more precision of the data, the easier will be the work of the users to accomplish their work.

3 -Time of retrieving information from the database:

Time is a vital factor in the healthcare process. Therefore there were two sets of experiments to evaluate the time required to retrieve information. The first sets of experiments need to evaluate time to retrieve information (this process was repeated and average of measures taken) for increasing the number of records.

The second set of experiments looked at the time to retrieve information when the classical Chain, developed Chain based ontology and RBAC methods are applied.

The desired results from this set of experiments: To retrieve information without notification of delay from the users.

4.5 Discussion and conclusion

This chapter stated the research questions, stated the hypothesis and the criteria of success that reflected whether the hypothesis and goal of the study were reached. The proposed experimental approach and how it can be evaluated has been given.

The main question the researcher is trying to answer is: ***Can a semantic approach based on simple limited acts of Chain method improve the data access management of personal information while preserving privacy?*** In order to solve this problem, the following process will be undertaken:

- *To simplify the configuration of a data access management system for a given database through a reduction of complexity.*
- *To increase the precision for data retrieved out of users queries.*

1- Collecting knowledge that forms a foundation for access control requirements in healthcare systems.

2- Creating improved access control models for healthcare systems based on real requirements

Criteria for success should be the reduction of complexity of applying data access control policies to the selected users and also the reduction of errors. The criteria of success for improvement of performance and precision are based on well-known measures from the NIST security services model and based on accuracy and precision measures through testing given scenarios that allow assessment of how close a given technique comes to the ideal solution. The crucial issue for a successful approach is to ensure that legitimate access is provided at all times, while avoiding inadvertent release of unnecessary data. In the next chapter, the design of the developed system will be presented based on the hypothesis given in this chapter.

In short, this thesis proposes the combination of chains and semantics to improve access management to personal information. This will be through implementing and evaluating the Chain method, reduction of design complexity and provision of an extensible PI ontology.

Chapter 5

Proposed Architecture for the Chain Ontology Base

In the previous chapters, the problem of preserving privacy in the course of data access management has been discussed. The problem has been analysed using a pertinent example from the healthcare sector. Because of the dynamic nature of this case in terms of incoming and out coming flows of sensitive information it is particularly difficult to preserve privacy in such a changing environment.

The next step is to design a solution, which is the purpose of this chapter. Based on the insights gained from the problem analysis, an effective and workable design has been developed. The researcher will also illustrate how lessons learned from the literature review have been incorporated into the design. Consequently, the present chapter is organised as follows:

- Section 5.1 revisits the key outcomes from the earlier problem analysis;
- Section 5.2 highlights the key system requirements;
- Section 5.3 presents the concept of the proposed solution and the logical architecture required to implement the proposed solution.
- Section 5.4 shows the system infrastructure.
- Section 5.5 analyses the ontology design.
- Section 5.6 gives an example of how the system works.

5.1 The Problem

Automated processing systems often store data that is required to be protected from unauthorised disclosure. For example, medical information is stored by automated processing systems at doctor's offices, hospitals, insurance companies, and various other facilities. Protecting this information from unauthorised or even inadvertent disclosure is becoming an increasing concern and in some cases is also the subject of industry regulation. For example, the Health Insurance Portability and Accountability Act (HIPAA) requires that individuals' health related information be protected from unauthorised disclosure.

One example of limiting access to resources assigns individuals to groups and access to particular data operations is granted or denied to all members of selected groups. However, identifying an accurate and complete set of groups for an organisation such that the group definitions correspond to different needs for information access and information, can be inaccurate and time intensive and subject to change over time.

In Chapter 2, the researcher has presented the problems affecting the effective management of data access in Chapter 3, the healthcare sector was chosen as a crucial example that shows real problems facing a sensitive domain such as the healthcare domain. The researcher has taken the International Clinic in Kuwait as a real example from the healthcare field. Because of the context in which the clinic retains and processes personal information, it is critical that they are able to maintain very tight control over how the information is used. This includes the individuals and users that have access to the data, the circumstances and contexts under which they are granted such access, and control over the specific actions they can take on this data. This includes for example collecting, modifying and disclosing to third parties. In this scenario individuals may be staff within the clinic, or external parties such as the insurance company. Therefore, the rules governing access must take into consideration:

1. The proprietor of the data (e.g. patient)
2. The individual processing the data (e.g. doctors, nurses.)
3. The context under which the data is being used (e.g. appointment, emergency.)

From the above three elements, the access control mechanism must define the specific actions which can be taken to handle the data in question.

The problem is twofold. The first aspect comes from the fact that the rules required to enforce the privacy policies are very complex. The complexity of these rules can make them difficult and time consuming to define, which is the case in all the traditional methods discussed in Chapter 3 such as RBAC, Hippocratic and XACML. As the system approaches this level of complexity, it becomes more and more difficult to provide an assurance of correctness. This increases the probability of errors in the definition of data entities, which could result in unintended side-effects and ultimately data protection breaches. Since access policies will probably need to be reviewed from time to time (in line with changes in policy or legislation), this problem will be constant and on-going.

Secondly, due to the sheer volume of data being controlled by the clinic, and the consequent fluidity of this data, the access control mechanism requires a high level of flexibility in order to manage the roles of users and constantly changing contexts which govern how the data can be accessed. In such reliable organisations of large size, it is not practicable to update and check the rules manually every time there is a change to the database. The large volume of data also requires a lot of consumption power to manage. A large set of complex rules applied to a large set of data will inevitably consume a lot of CPU resources at runtime.

In designing robust data access management, the following questions should be considered:

1. Who should be responsible for access policy?
2. What kind of access policy do you require?
3. What resources do you need to protect?
4. How do I plug in the access management solution?

This chapter describes a systematic approach to managing the complexity associated with software access management. In section 5.2, a look at the system requirements will be given. Next in Section 5.3, explanation of the system design and its different components will be presented.

5.2 System Requirements

In the literature review chapter, outstanding problems of traditional access management methods have been identified. These problems are:

- Complexity of system design;
- Lack of context sensitivity while performing data access management;
- Lack of precision while retrieving required data.

Therefore in developing new data access management systems, such concerns should be taken into consideration. For the first point, a decision has been made to use the Chain method (Al-Fedaghi, 2007) as it has the simplest design with the least number of parameters based on seven limited acts. While for the other two points, an ontology and semantics have been developed specifically for this project to overcome these two problems. In the following two sections, other system requirements will be highlighted.

5.2.1 Who should be responsible for access policy?

To implement an application access management solution, it must be ensured that access policies exist and are unambiguous. Although access controls will be enforced by technology, defining the required access policy is the responsibility of the business.

For this reason, an access policy related to the release of sensitive information and/or application features should be documented using business terminology. During the analysis of these business requirements, concise rules will be defined governing who has access to specific classes of business or personal information and under what circumstances (there may also be rules regarding who can access application features).

This analysis often requires a significant classification effort in three areas: 1) information; 2) application features; and 3) people. Many organisations already have an information protection group entrusted with ensuring that business policies are in place to ensure the protection of business and personal information. Such an organisation can play an important role in ensuring that access policy is consistent across business applications. If each application group does this classification independently, inconsistencies in policy may occur.

But can an internal organisation define access policy? Increasingly the answer is no. Legislation regarding confidentiality and privacy requires that individuals be allowed to define who (and under what circumstances) personal information is released. This adds new requirements for business applications in the area of access management. The users have become policy administrators with respect to access to personal information. While the application may restrict the policy choices, it must be able to dynamically change the policy in use.

5.2.2 What kind of access policy is needed?

An access policy can be very simple or very sophisticated. Once it has been determined that applications require access management features, they typically begin with a very simple access control policy based on user identity. There are many applications, however, that require more sophisticated access policies. To determine all requirements for access management solutions, one should determine the type of access policy should be determined. Access Policy can be classified as follows:

Policy Type	Question answered with regard to protected resource (information or application feature)	Example(s)
Identity-Based	Are you an individual that has been specifically granted access?	User ID / Password, Private Key, Electronic Token, Biometrics
Role-Based	Are you currently in a role that has been specifically granted access?	Manager, Emergency Room Personnel
Group-Based	Are you part of a group that has been specifically granted access?	Accounting, Engineering
Context-Based	Is the context of the request such that access should be granted to this individual?	Time of Day, Location, Emergency, Account Balance
Entitlement-Based	Is this individual entitled to access this class of information?	Clearance Level
Relationship-Based	Is this individual entitled to access the Personal/business information because of a relationship with the person or business?	Primary Care Physician, Manager of Employee, Account Representative, Parent
Rule-Based	Does the policy governing access to the resource allow this individual to access the resource?	Combination(s) of above

Table 4: Access Types

Access Management solutions may also support different types of rules. For example, iLock Security Services supports all of the Policy Types shown above and allows an access policy to have multiple “rules.”

These rules determine whether or not to allow access. Rules are of the following types, and in a “rulebased” policy are evaluated in the following precedence order:

Rule Type	How the rule is evaluated	Example of usage
Nobody	Deny access to everyone.	In a Context-Based Policy, access may be denied during certain times of the day
Deny	Deny access to anyone that has any of these credentials (access ID, group, role).	A security alert is in place. You may wish to Temporarily deny certain groups who normally have access.
Required	Allow access only if the requestor has all the credentials.	Allow only owners who are officers (you must be both an officer and an owner).
Any	Allow access to anyone with any of these credentials.	You wish to allow users who are in the group administrators or.
Anybody	Allow access to anyone.	You may wish to audit the request for the resource even though you do not restrict access.

Table 5: Rule Types

The system described below is directed at enabling healthcare vendors to augment their current systems to address the current market demands and regulatory requirements through non-intrusive techniques that seamlessly handle the many various permissions that are required to effectively support their operations. These permissions include, for example, storage access rights and “execute” rights for methods of individual object classes. The system and methods described below overcome the challenge typically associated with reliably automating the determination of the purpose of requested data accesses.

5.2.3 What to protect?

Traditionally, machines and networks have been the resources commonly protected. However, while integrating applications and expanding the use of systems, it has been noticed that the application assumes the responsibilities for guarding access to business information and/or application functionality. The security community uses the generic term “resource” when discussing business information or concepts that need to be protected.

Protected resources are typically given a unique name (or ID) that is used in communicating with an access manager to request an access decision. Deciding what resources should be protected and assigning them an ID sounds simple – and sometimes it is – but it can also become a time-consuming identification and data classification project not considered in the original application estimates. For example, most hospitals would agree that a patient medical record should be protected. In the next table issues such as:

– ‘Where do I insert the software guard?’,

– ‘What is the actual resource it must protect to ensure medical information is not accessed improperly?’

Granularity of Protected Resource	**Access Policy that protects the system**
Machine and/or network	Only people with the authority to run the application have User IDs on the machines-such as the database administrator
Entire Application	Only people with the authority to view HR information are granted User IDs for the human resources application-the database administrator and the general manager
Specific Application Feature (e.g. Screen, Menu, Button or URL…)	Specific GUI for each group of users: Doctors, Nurses and Receptionists
Entire Database	Only people with the authority to view HR information-The database administrator- have User IDs in the human resources database. The database is accessed using requestors’ ID.
Table (in a database)	According to the context user can see a specific table from the database.
Row (in a table in a database)	According to the context user can see a specific table from the database.
Field (in a Row in a table in a	According to the context user can see

database)	a specific table from the database.
Concept (information that contains multiple fields – potentially from different sources)	According to the context user can see a specific table from the database.

Table 6: Access policies that protect the system

From the literature review, two concepts have been realised:

- There is no adequate personal information ontology in the literature which is related to H1 and H2 of our hypothesis.
- There is no method that provides a sufficiently simple design among the all methods that have been reviewed from the literature which is related to H1 of our hypothesis.

Therefore these were the two basic concepts that the researcher has started to build upon for this investigated system. The need for an ontology that clarifies concepts is of crucial importance in the information age, and its importance in the data access management field is more obvious. In the next section we will show how we have integrated semantic concepts into our system of chain database.

5.3 System Design

The system and method described in this section addresses specification and enforcement of access rights to shared data resources, such as Electronic Medical Records (EMRs) and other forms of individual health records. This system is based on identified information flow models that represent the movement of information and the actions taken on that information.

As shown in Figure 28 for example, an information flow model identifies actors, resources, and processes used to handle protected information (acts or chain of acts). These processes are divided into a limited set of discrete actions. For example, multiple actors are able to create new information, where those actors may or may not own the created information.

The created information is then able to be used at several different locations by various information users. One example of processing medical information includes analysis of the data, such as by editing the medical record.

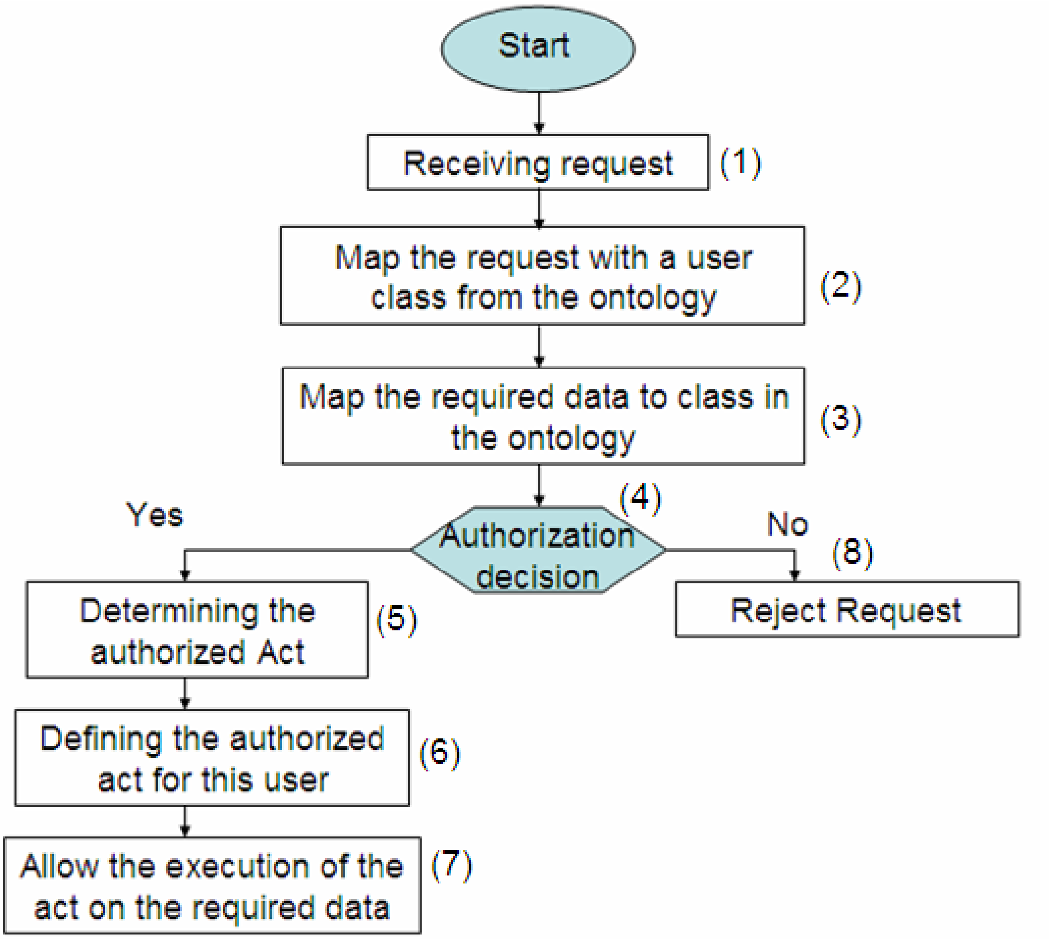
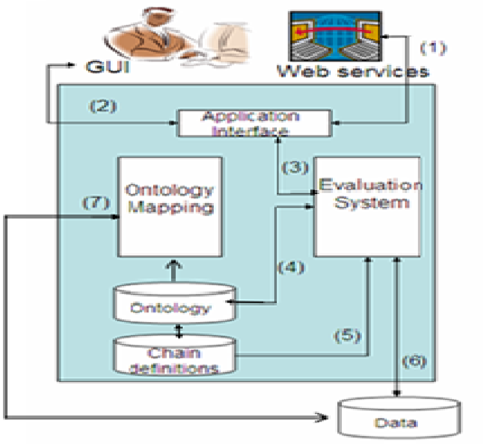

Start
Receiving request
(1)
Map the request with a user class from the ontology
(2)
Map the required data to class in the ontology
(3)
(4)
Yes
Authorization decision
No
(8)
Determining the authorized Act
(5)
Reject Request
Defining the authorized act for this user
(6)
Allow the execution of the act on the required data
(7)

Figure 28: System Processing Flow Model

the developed system works as follows:

At point (1): the system receives a request from a specific user. (e.g. a doctor is requesting to access the medical record of a specific patient). At point (2): the system will first check from the database the group of users which this user belongs to (in the example of point 1 the group of the users will be "doctor"). After checking the group of users, the system will return back to the ontology-and map the given request with the conditions and properties of that class of users (for the above example it will check if the access to patients medical records is one of the properties given to the doctor class of users). At point 3: the system needs to map the class "medical record" in the ontology to the database entity "medical record". At point 4: Authorisation to the user to access the database will be either given to the user or not (in the above example the doctor will be given access to the medical record data because it has this privilege according to his group of users in the "doctor" class in the ontology. Then at point (5) the system will retrieve the acts (from the 7 acts of the chain method) that are authorised for the combination of user/database entity from the ontology (for the example the doctor will be given acts of: collecting, processing and storing to act on the database entity: medical record). At point (6): these acts will be enabled from the GUI as the acts represented in the GUI by buttons (In the example, the buttons of collecting, processing and storing will be enabled in the GUI for the doctor). At point 7: The user can perform all the acts which he has been given authorisation for (Finally in the example, the doctor can perform collecting, processing and storing on the patient medical record).

The abstract terms defined in figure 28 (such as start, receiving request, etc.) are translated to more practice terms (such as web service, GUI…) Figure 29 above illustrates an ontology-based access control processor that includes several data stores (the ontology and the database) an interface (java GUI) and an evaluation system that works as described previously in figure 28.



Figureb29: Ontology-based access control system

The system has two inputs at point 1 from web services (e.g. user trying to access the system from home-such as doctor wants to check his appointments schedule from home). Other input at point 2-each authorised user can access from the system GUI through the hospital LAN. Point 3 works similar to point 4 in figure 28. The system through its application interface needs to receive access requests from either of the two inputs and submit them to the evaluation system at point 3. The evaluation system will need to refer to the ontology that is developed in protégé OWL to check authorisation privileges given to this user as mentioned previously in figure 28. The ontology needs to check from the database the group of user which this user belongs to. To do so, the ontology needs to be mapped to the database at point 7. Checking the group of user, the ontology in protégé owl will check the given properties and conditions to this user at point 5 and accordingly the authorised acts of chain (as in our example the doctor is only allowed to access his appointment schedule to “collect” information only as he can’t “process” means change or edit the appointments as this is the job of other group of users which is the receptionist). The ontology will then retrieve the authorised set of acts to the evaluation system at point 4 and accordingly the evaluation system will allow the user to access the database entity at point 6.

5.4 The Ontology Design

The whole ontology is presented in Figure 30.

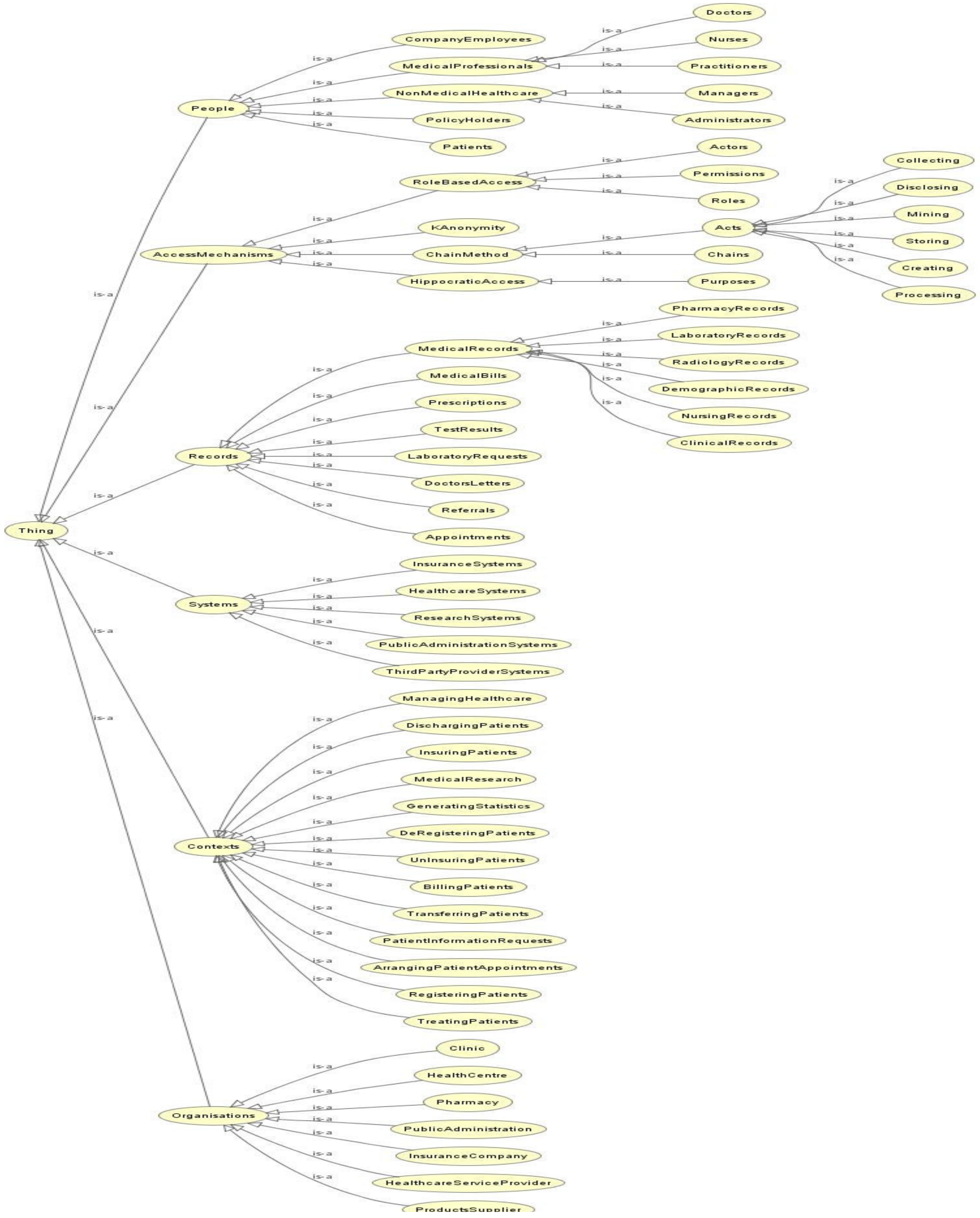


Figure 30: Overall Ontology classes.

While Figure 31 gives more focus on the healthcare part:

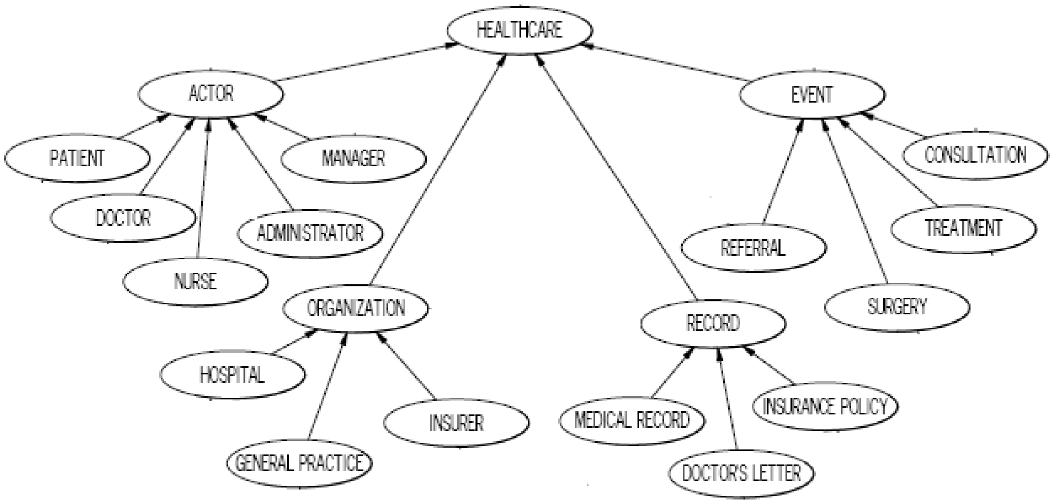


Figure 31: Focus on the healthcare part of the ontology.

The ontology given on the previous figure, defines the components of a healthcare organisation that is used to implement an ontology-based access control. In this healthcare organisation example, different classes of actors or organisations are each provided access to different types of information or stored data. Each of the different classes of actors or organisations is also able to be provided with different permissions with regards to the different acts that can be performed on that information or stored data upon which the acts are performed.

The top level of the ontology states “healthcare” and indicates that all lower level nodes of the example ontology are part of the “healthcare” organisation. The healthcare organisation has an actor node with lower level nodes defining each class of actor. Illustrated are a patient, a doctor, a nurse, an administrator and a manager. Each class of actor is generally provided with different accesses to information and to different acts that can be performed on that information.

The healthcare organisation has an organisation node with lower level nodes defining each class of organisation that is permitted access to information. Illustrated are a hospital, general practice and insurer organisation. Access to information and to different acts that can be performed on that information are able to be restricted or granted based upon the organisation associated with the user who is requesting the act.

The healthcare organisation has a record node with lower level nodes defining each class of data record that is stored and operated upon by acts performed on that data. Each class of data record is a type of data stored in a database or other data storage device and to which access is controlled. Illustrated are a medical record, a doctor’s letter and an insurance policy data set. Access to each of these classes of data, and the type of access such as create, access, delete is able to be restricted or granted based upon the class of requestor or organisation associated with the requestor of the act to be performed on that data.

The healthcare organisation has an event node with lower level nodes defining each type of context that can occur in the healthcare organisation. Illustrated are a referral, surgery, treatment and consultation. The ability to initiate each of these events/contexts, or to be the object of each of these events, is able to be restricted or granted based upon the class of requestor or organisation associated with the requestor of that event or of the object of the requested event. For example, a doctor is able to request medical record of a patient, but an administrator is not able to request patient medical record.

The system receives a request from the user. The system uses the ontology mapping and the ontology definition database to determine the class of actor to which the requestor belongs. The system accesses the chain associated with the requested act and determines if the user is authorised to perform the act associated with that chain on the requested data.

5.5 Example of the Mechanism of the Overall System

The overall mechanism of the system and all its parts is presented in the following two figures.

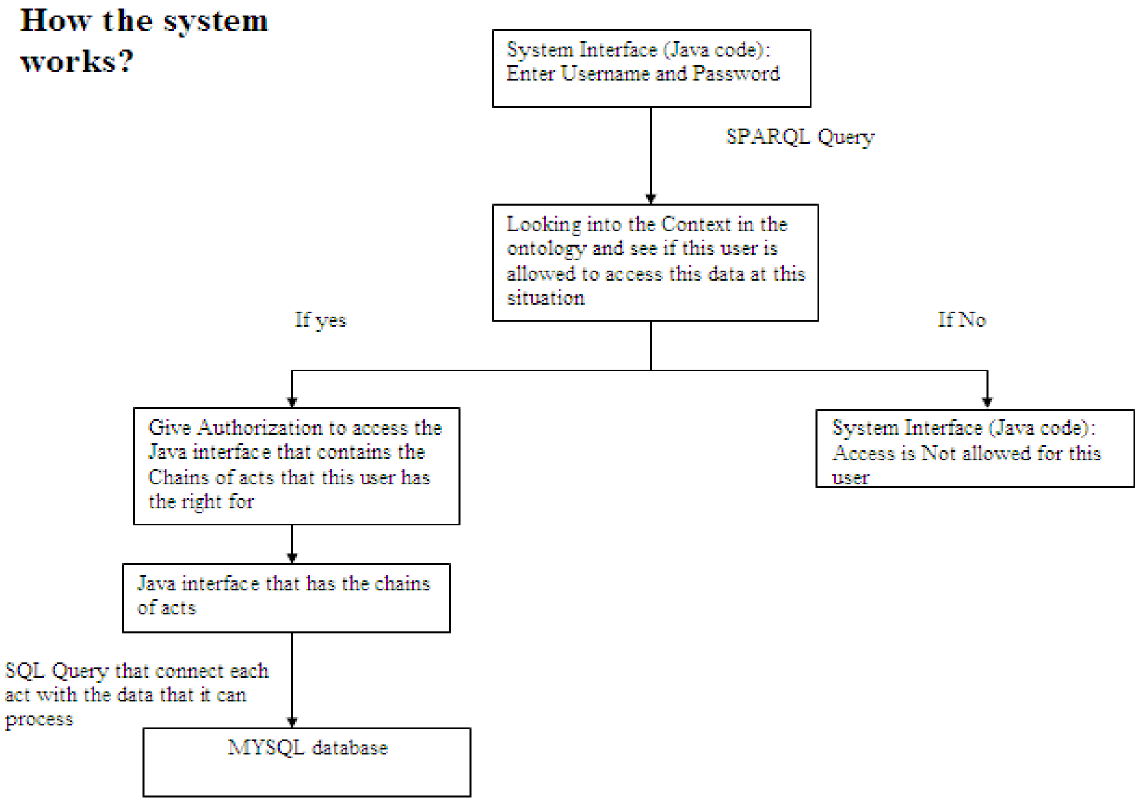

How the system works?
System Interface (Java code): Enter Username and Password
SPARQL Query
Looking into the Context in the ontology and see if this user is allowed to access this data at this situation
If yes
If No
Give Authorization to access the Java interface that contains the Chains of acts that this user has the right for
System Interface (Java code): Access is Not allowed for this user
Java interface that has the chains of acts
SQL Query that connect each act with the data that it can process
MYSQL database

Figure 32: How the system works.

And the following figure (Figure 33) is an example of how the system above works for a specific scenario.

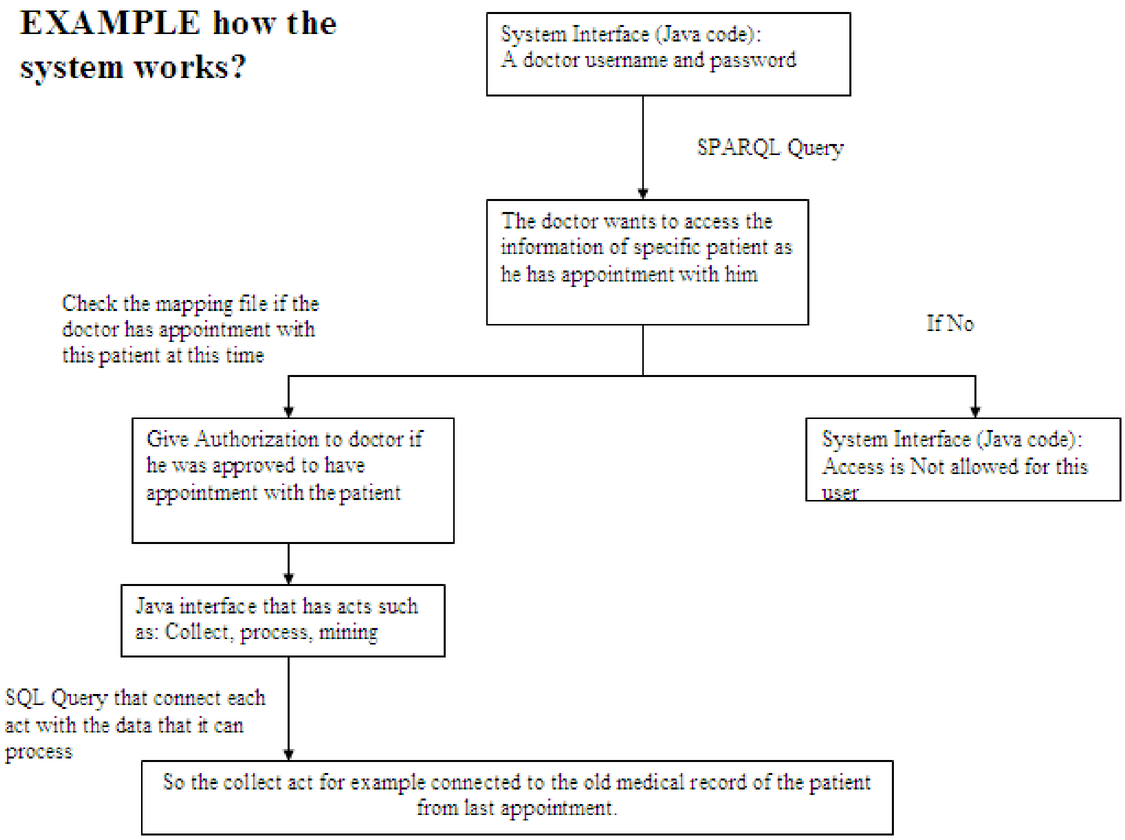

EXAMPLE how the system works?
System Interface (Java code):
A doctor username and password
SPARQL Query
The doctor wants to access the information of specific patient as he has appointment with him
Check the mapping file if the doctor has appointment with this patient at this time
If No
Give Authorization to doctor if he was approved to have appointment with the patient
System Interface (Java code):
Access is Not allowed for this user
Java interface that has acts such as: Collect, process, mining
SQL Query that connect each act with the data that it can process
So the collect act for example connected to the old medical record of the patient from last appointment.

Figure 33: Example on how the system works.

As shown in the two figures: a data access request will begin with the user requesting access to a specific record (for example a doctor requests a patient record). From the profile of the current user, it is possible to determine what kind of user they are designated (for example a doctor). By querying the ontology, one can determine the set of contexts under which this user is allowed to access the data (in this example, appointment, emergency, consultation, etc.). Next, each context must be checked for preconditions until a context is confirmed as being valid for the user (in this case the appointment). Checking the preconditions may involve gathering data from the database, or querying the credentials of the user. Once a given context is valid for the request, access can be granted. Otherwise, access to the record is denied.

5.6 Summary

The method, for controlling access to sensitive data, should contain:

- Storage of chain definitions, each chain definition is defining a respective chain to perform a respective act on respective data, and the respective chain defining the number of processes to be performed on the respective data.
- Receiving a request from a user to perform a requested act on a requested data.
- Defining a class of actor associated with the user to determine permissions for the class of actor associated with the user to execute a chain that performs the requested act on the requested data.

Implementation of the system that has been designed in this chapter will be presented in the next chapter.

Chapter 6

System Implementation and Testing

This chapter will focus on the process undertaken to implement and integrate each component of the system such as:

- Chain method
- Ontology and semantics
- The Java code

Implementation choices according to design strategy and problems experienced during implementation process will be also highlighted in this chapter.

6.1 Overview of the System Architecture

The user data on a working system will be constantly changing, so a requirement of this project is to provide and act upon a live view of the database in order that it can behave dynamically as the database updates and changes. The system also needs a facility to write data back to the database. This means that the system needs to address this issue also, by providing both read/write access to the live data as show in Figure 34.

In order for the prototype to be usable, it was also agreed that a GUI needed to be provided to the user for the presentation and input of data.

In short the system requirements are:

1. To provide a live connection to a database.
2. To apply access control to this database in a context sensitive manner.
3. To allow configuration of the system via a domain ontology as provided by the system administrator.
4. To provide both read and write access to the database.
5. To present some form of GUI, intended for the end user.

The requirements of the project were formally identified. A high level overview of the system was produced in order to provide an overall impression of the structure of the prototype as shown in Figure 34.

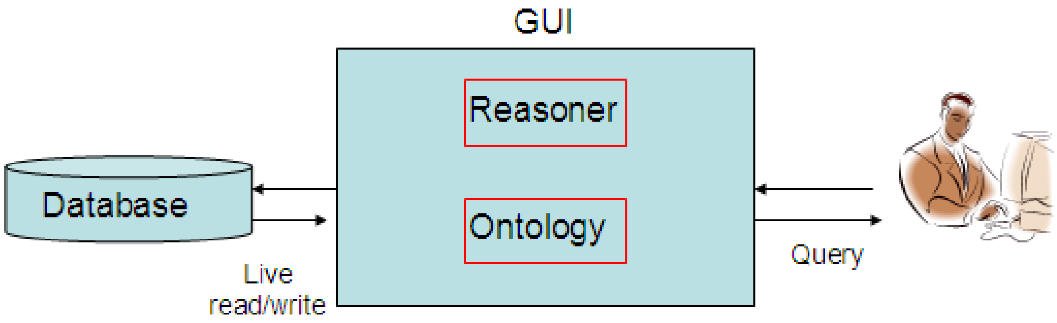


Figure 34: a high level overview of the system according to requirements.

A list of system scenarios has already been presented in Chapter 3. Scenarios were used for the design of the medical ontology and were intended to provide formal documentation of the semantic rules to be applied to the database.

These lists of scenarios provide quite comprehensive coverage for all possible users of the system and all of the possible actions these users at any hospital could undertake.

The system design is an integration of many subsystems such as ontology, reasoner and database. The following diagram, gives a close look at the system back end and the “model” of the MVC design pattern. It illustrates how the subsystems (ODEMapster, Jena) interact with the reasoner to form a whole system. In the following sections the researcher will go through the subsystems in detail.

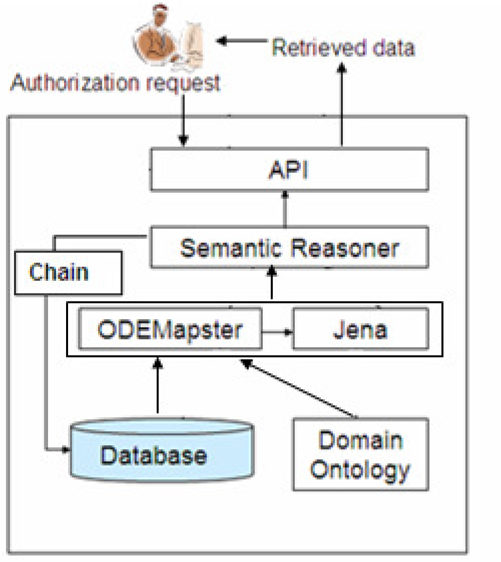


Figure 35: A high level MVC diagram.

The first implementation decision that was made was to select the Java language to code the system, because this language is understandable by all other supporting elements such as Protégé OWL, Jena (HP Labs, 2009) and ODEMapster (UPM, 2010). Using Java simplified the integration between these tools, because they could be imported directly into the project as required.

The other main component was the database. This database has gone through many versions: the first version used ORACLE DBMS while the final version used MySQL DBMS, because some of the supporting tools such as: ODEMapster has not been previously connected with ORACLE. In addition, MYSQL provides also a large database which was required for the experiments.

Another key requirement of the prototype was to provide a live connection to the fundamental database. In order to do this, two choices were available: D2RQ and ODEMapster. ODEMapster was chosen because it provides more GUI features and is easier to use. Also it was essential to define a direct mapping from the fundamental database system to the reasoning engine and ontology which would define the access rules. Therefore colleagues from the University of Madrid have suggested using ODEMapster in conjunction with the NeON toolkit to achieve this. This would provide the required GUI for configuring the database. In addition the latest version of this programODEMapster2 allowed programmatic access to the engine and allowed it to be embedded within the system more efficiently.

In a real system, it is crucial that the back end model of the system must be run directly on the database server or another trusted system, as running the model on the user's computer may cause the passing of unfiltered data across the network and to systems over which the organisation has no control. This would cause a threat to the system security.

A central part of the system is the use of ontologies for modelling and reasoning. As has been mentioned in the previous chapters, our proposed solution to preserving the privacy in the data access management should make use of ontologies to define the problem domain, and in doing so provide a definition of the prerequisites by which the information stored in the database will be controlled. This demands that the system must have the ability of reading, interpreting and manipulating ontologies.

For this purpose it has been decided to use a tool known as Jena. This simplifies a considerable amount of the work required for interfacing with the ontologies. There are other APIs that could have been used, such as Jastor (Szekely and Betz, 2011) and OWL API (University of Manchester, 2011), but Jena is currently the most popular and reliable option as it supports for connection to other tools. Now that the relevant subsystems have been identified, it is possible to discuss how these will fit together to form the system as a whole.

In the following sub sections the researcher will go through the subsystems in detail.

6.2 Chain Method

As mentioned in Chapter 2, Al-Fedaghi proposed an alternative way of dealing with personal information. In order to do this, 7 distinguished types of acts on personal information have been identified, varying from collecting data from a specific proprietor to the disclosing of data to a specific agent. The purpose of accessing or processing is defined by a chain of acts that are carried out during the course of this processing activity. By following the possible paths that data may take in order to transfer from one act to another, potential threats in the data protection policy can be occurred by identifying the possible paths by which personal information may end up in malicious hands. Chains can be used to observe the purposes in a visual sense which helps in the process of simplifying the defining the privacy policy (Al-Fedaghi, 2007). The Chain method will be used according to the context founded in the ontology, in order to decide which authorised users to access what data and what are the functions to be given to those users on this data. The Chain method for a Receptionist/Admin is given in the following figure.

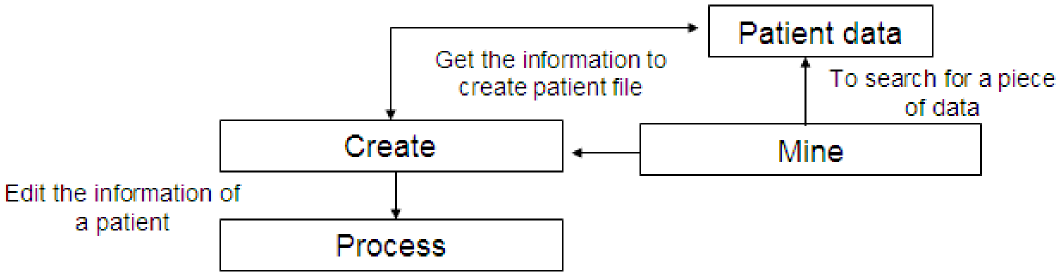
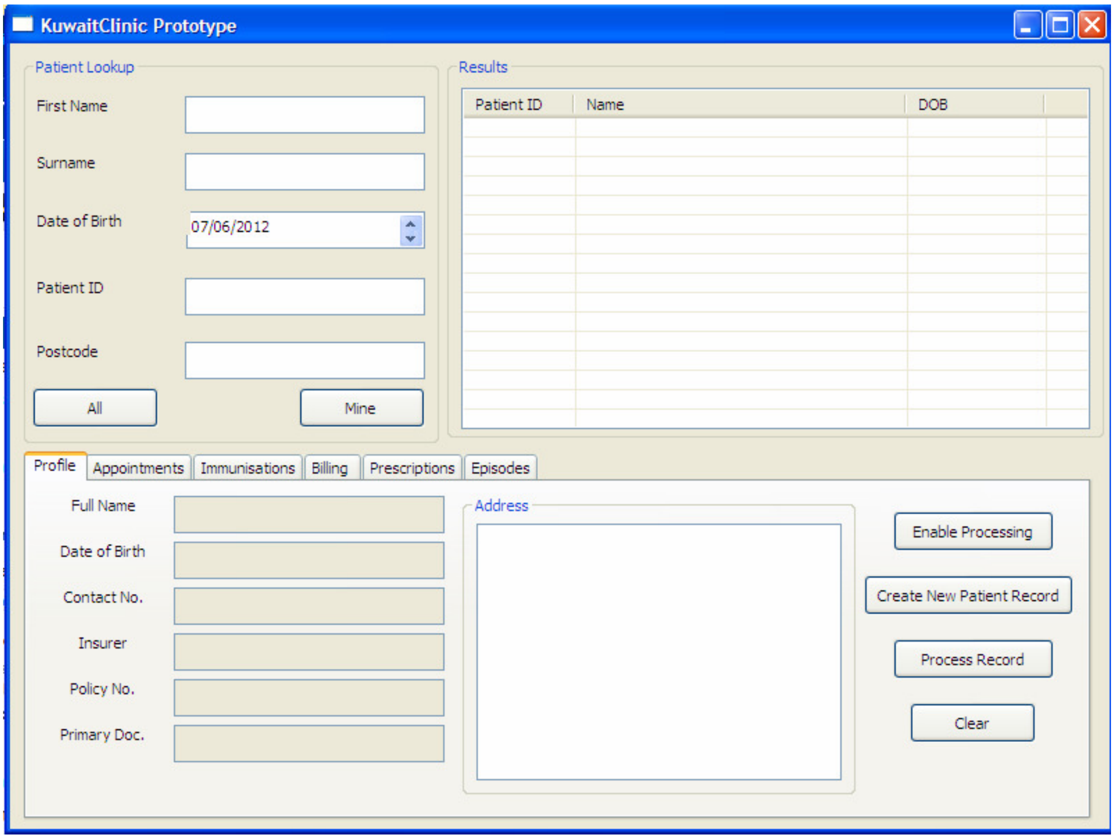

Patient data
Get the information to create patient file
To search for a piece of data
Create
Mine
Edit the information of a patient
Process

Figure 36: Acts of Chain for Admin.

The Chain has been implemented in the system at the application level. As shown in the figure, the acts of the chains for the receptionist are shown as buttons that are connected to the required database. Once the user is authorised these buttons will be enabled to act on the required data. The acts that are applicable for the receptionist in this case are Process, Mine and Create. Each act is associated with specific functions such as: Create-to create new profile file at the registration process, Mine-to search for existing patient data and process-to edit the information of an existing patient.



Date of Birth 07/06/2012

Figure 37: The "acts" of the Chain method as it appears in the Admin GUI.

In addition, those acts (that are presented as buttons) are connected to specific entities of the data and can only be applied to these entities according to the context given.

6.2.1 Overview of the Developed Database

The process that will be followed for the construction of the database is the designing Phase, which includes a number of iterative steps for the end-product to be flexible. This phase actually defines the information (+ its structure) that will go into the database, the assumptions made related to the type or values of the data items and the relationship between the data items within the database. In fact, there is a need to construct a database for the scenarios that have been collected from the hospital. Standard practice will be followed based on exist clinic management systems such as open HER.

In order to construct the required database, the following main steps have been followed:

1. Requirement Analysis

The database requirements are determined. The exact requirement of the user from the system is captured. All the relevant information related to the system is gathered. Therefore the procedures below have been undertaken to get the required information to build the system database:

- Sampling of existing documentation, forms, databases from the IC
- Research and site visits
- Observation of the work environment in the IC
- Questionnaires for the staff at the IC
- Prototyping build a small model of the user's requirement to verify beforehand. Our database went through many version before it reached its final shape
- Joint Requirements Planning (JRP)- group meetings were conducted to analyse existing problems

2. Entity Relationship Diagram (ERD)

The information gathered during the ‘Requirement Analysis’ step is transformed into an ERD (Entity Relationship Diagram) that is the data is organised into entities and relationships between them. So instead of going through a lengthy piece of material, a pictorial representation of the same piece of information will be provided which is easier to read.

The ER diagram is presented in Figure 38.

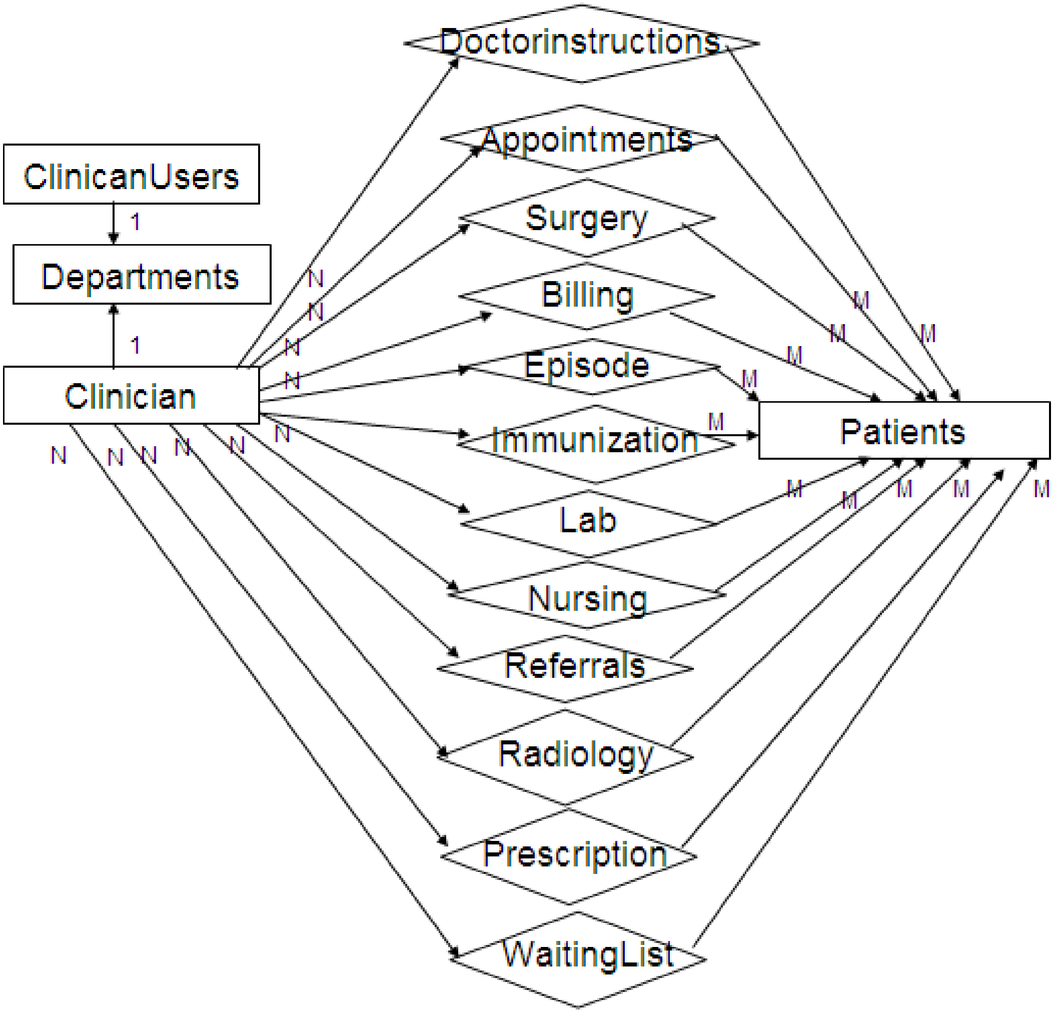


Figure 38: ER of the developed database.

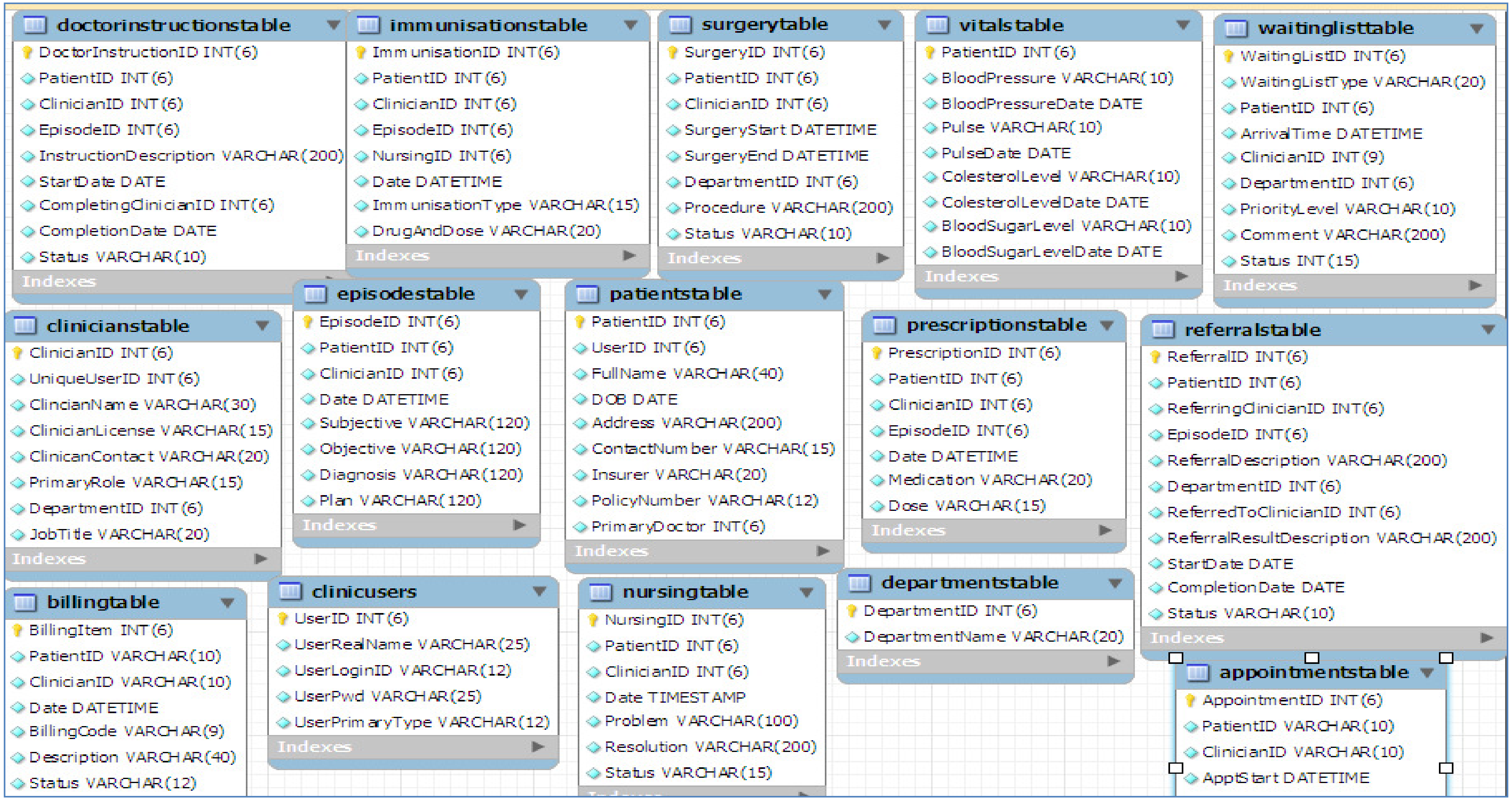
Various data modelling languages can be used to create an ERD such as crow's foot notation, Chen notation, IDEFIX (Integration Definition for Information Modelling), Shading notation, Bachman notation, UML (Unified Modelling Language) standard etc.

3. Relational Model

After developing the ER diagram, it will be converted into a relational model through the following three steps:

- Turned each non-weak entity set into its corresponding table with the same set of attributes
- Replaced a relationship by a relation whose attributes are the keys of the connecting entity sets
- Replaced a weak entity set by a relation whose attributes are its own attributes (if any) plus the borrowed attributes that help to make its primary key.

Figure 39 shows the main database tables and attributes.



doctorinstructionstable

- DoctorInstructionID INT(6)
- PatientID INT(6)
- ClinicianID INT(6)
- EpisodeID INT(6)
- InstructionDescription VARCHAR(200)
- StartDate DATE
- CompletingClinicianID INT(6)
- CompletionDate DATE
- Status VARCHAR(10)

Indexes

immunisationstable

- ImmunisationID INT(6)
- PatientID INT(6)
- ClinicianID INT(6)
- EpisodeID INT(6)
- NursingID INT(6)
- Date DATETIME
- ImmunisationType VARCHAR(15)
- DrugAndDose VARCHAR(20)

Indexes

surgerytable

- SurgeryID INT(6)
- PatientID INT(6)
- ClinicianID INT(6)
- SurgeryStart DATETIME
- SurgeryEnd DATETIME
- DepartmentID INT(6)
- Procedure VARCHAR(200)
- Status VARCHAR(10)

Indexes

vitalstable

- PatientID INT(6)
- BloodPressure VARCHAR(10)
- BloodPressureDate DATE
- Pulse VARCHAR(10)
- PulseDate DATE
- ColesterolLevel VARCHAR(10)
- ColesterolLevelDate DATE
- BloodSugarLevel VARCHAR(10)
- BloodSugarLevelDate DATE

Indexes

waitinglisttable

- WaitingListID INT(6)
- WaitingListType VARCHAR(20)
- PatientID INT(6)
- ArrivalTime DATETIME
- ClinicianID INT(9)
- DepartmentID INT(6)
- PriorityLevel VARCHAR(10)
- Comment VARCHAR(200)
- Status INT(15)

Indexes

clinicianstable

- ClinicianID INT(6)
- UniqueUserID INT(6)
- ClincianName VARCHAR(30)
- ClinicianLicense VARCHAR(15)
- ClinicanContact VARCHAR(20)
- PrimaryRole VARCHAR(15)
- DepartmentID INT(6)
- JobTitle VARCHAR(20)

Indexes

episodestable

- EpisodeID INT(6)
- PatientID INT(6)
- ClinicianID INT(6)
- Date DATETIME
- Subjective VARCHAR(120)
- Objective VARCHAR(120)
- Diagnosis VARCHAR(120)
- Plan VARCHAR(120)

Indexes

patientstable

- PatientID INT(6)
- UserID INT(6)
- FullName VARCHAR(40)
- DOB DATE
- Address VARCHAR(200)
- ContactNumber VARCHAR(15)
- Insurer VARCHAR(20)
- PolicyNumber VARCHAR(12)
- PrimaryDoctor INT(6)

Indexes

prescriptionstable

- PrescriptionID INT(6)
- PatientID INT(6)
- ClinicianID INT(6)
- EpisodeID INT(6)
- Date DATETIME
- Medication VARCHAR(20)
- Dose VARCHAR(15)

Indexes

referralstable

- ReferralID INT(6)
- PatientID INT(6)
- ReferringClinicianID INT(6)
- EpisodeID INT(6)
- ReferralDescription VARCHAR(200)
- DepartmentID INT(6)
- ReferredToClinicianID INT(6)
- ReferralResultDescription VARCHAR(200)
- StartDate DATE
- CompletionDate DATE
- Status VARCHAR(10)

Indexes

billingtable

- BillingItem INT(6)
- PatientID VARCHAR(10)
- ClinicianID VARCHAR(10)
- Date DATETIME
- BillingCode VARCHAR(9)
- Description VARCHAR(40)
- Status VARCHAR(12)

clinicusers

- UserID INT(6)
- UserRealName VARCHAR(25)
- UserLoginID VARCHAR(12)
- UserPwd VARCHAR(25)
- UserPrimaryType VARCHAR(12)

Indexes

nursingtable

- NursingID INT(6)
- PatientID INT(6)
- ClinicianID INT(6)
- Date TIMESTAMP
- Problem VARCHAR(100)
- Resolution VARCHAR(200)
- Status VARCHAR(15)

departmentstable

- DepartmentID INT(6)
- DepartmentName VARCHAR(20)

Indexes

appointmentstable

- AppointmentID INT(6)
- PatientID VARCHAR(10)
- ClinicianID VARCHAR(10)
- ApptStart DATETIME

Figure 39: Tables of our database.

4. Normalisation

An example medical database has been prepared according to the discussion and information collected from experts in the field and especially the database administrator from the hospital. This was a prototype relational database running on a MySQL environment. The researcher attempted to include all the necessary fields and tables required to cover the given scenarios from Chapter 4 in this database. The final version of the relational database is composed of 17 tables and a description will be provided of each table in the following:

- **Appointments table**– Records a list and details of all appointments arranged between patients and medical staff.

- **Billing table** – Records a list of billable items resulting from services provided to a patient.

- **Clinicians table** – Records a list of all medical staff, along with their information and specific job role.

- **Clinical users** – Records a list of all system users. This includes both medical and non-medical staff and can also include patients who have access to their own data.

- **Departments table** – Records a list of the different departments which form the clinic.

- **Doctorinstructions table** – Records a list of the instructions that have been specified by a doctor to be completed for the patient.

- **Episodes table** – Records a list of “episodes” which are simply individual cases where the patient has needed to visit the clinic.

- **Immunisations table** – Records a list of immunisations that have been given to each patient.

- **Lab table** – Records the results of laboratory testing, undertaken in order to diagnose the patient.

- **Nursing table** – Records instances where the patient has been kept under the care of the clinic.

- **Patients table** – A very crucial table. Stores a demographic record of each of the patients.

- **Prescriptions table** – Records the prescriptions that have been given by doctors to each patient.

- **Radiology table** – Records instances where a patient has visited the radiology department, along with subsequent results.

- **Referrals table** – Records a list of referrals, where patients have been referred to other areas of the clinic for further diagnosis or treatment.

- **Surgery table** – Records a list of instances where a patient has undergone surgery.

- **Vitals table** – Records each instance where a patient has had their vitals checked (blood pressure, etc.) along with the results of these checks.

- **Waitinglist table** – Records a list of patients who are waiting to be seen by a clinician. This differs from an appointment as the patient does not have an agreed time to be seen and patients should generally be seen by assigned priority based on the severity of their condition.

In this section, a description of the database has been given: starting with its basic requirements and then the creation of the ER diagram and finally, the transfer of the ER diagram into tables.

“The proof is in the pudding”—the quality of the database is assessed only by using it in applications for which it has been designed, as will be described later on this chapter.

6.3 Ontology

As discussed in previous chapters, the main chain method problem is in defining its terms (the 7 acts) and how to apply them in real applications such as the healthcare domain. The solution has been found in defining its terms in an ontology, as analysing domain knowledge is possible once a declarative specification of the terms is available. Formal analysis of terms is extremely valuable when both attempting to reuse existing ontologies and extending them (McGuinness et al., 2000).

Though the developed ontology for personal information in healthcare, this ontology can then be used as a basis for many applications but the researcher has chosen to use it in data access management.

The ontology has been constructed using the Protégé Owl environment in order to be connected to a real project from the health sector. Web Ontology Language (OWL) is part of the growing stack of W3C Recommendations related to the Semantic Web. The Semantic Web is a vision for the future of the Web, in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web.

Another main decision in developing the otology was whether to use an existing ontology from the literature or start working on it from scratch. There are libraries of reusable ontologies on the Web and in the literature. For example, (Ontolingua, 2011) or (DAML, 2011) could be used. There are also a number of publicly available commercial ontologies (e.g., UNSPSC, RosettaNet and DMOZ).

There are also a number of medical ontologies in the literature such as: open clinical ontology, clinical ontology website, UMBEL, open wetware and much more. However, none of them cover all the process in the hospital, the health personal information and the rules controlling the access management of the hospital database that are covered by the ontology proposed in this thesis.

The second phase of ontology implementation was writing down a list of all terms the researcher would like either to make statements about or to explain to a user. For our case, important healthcare-related terms according to our survey in the hospital was doctor, patient, nurse, medical record, prescription, etc. It was essential to get a comprehensive list of terms without worrying about overlap between concepts they represent, relations among the terms, or any properties that the concepts may have, or whether the concepts are classes or slots. Once this phase has been finished, a class hierarchy needed to be constructed.

There are several possible approaches in developing a class hierarchy (Uschold and Gruninger, 1996). In this thesis a combination approach has been followed (the combination approach is a combination of the top-down and bottom-up approaches). The researcher has defined the more salient concepts first and then generalises and assigns them appropriately. She started with a few top-level concepts such as AgentUser, and a few specific concepts, such as doctor. She then related them to a middle-level concept, such as AgentRole as shown in figure 40.

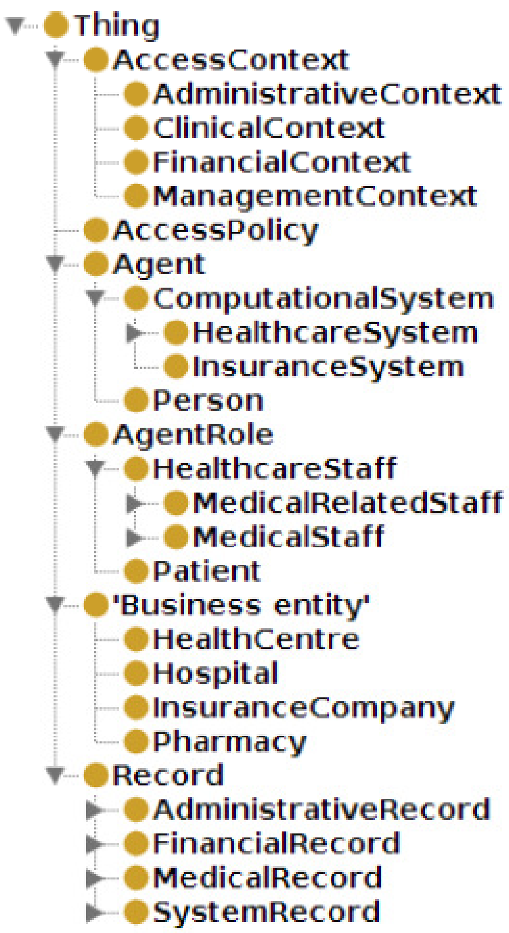


Figure 40: Basic classes of the developed Ontology.

In this thesis the third approach has been followed.

Figure 41 provides an overview of the basic structure of the classes, subclasses and relations.

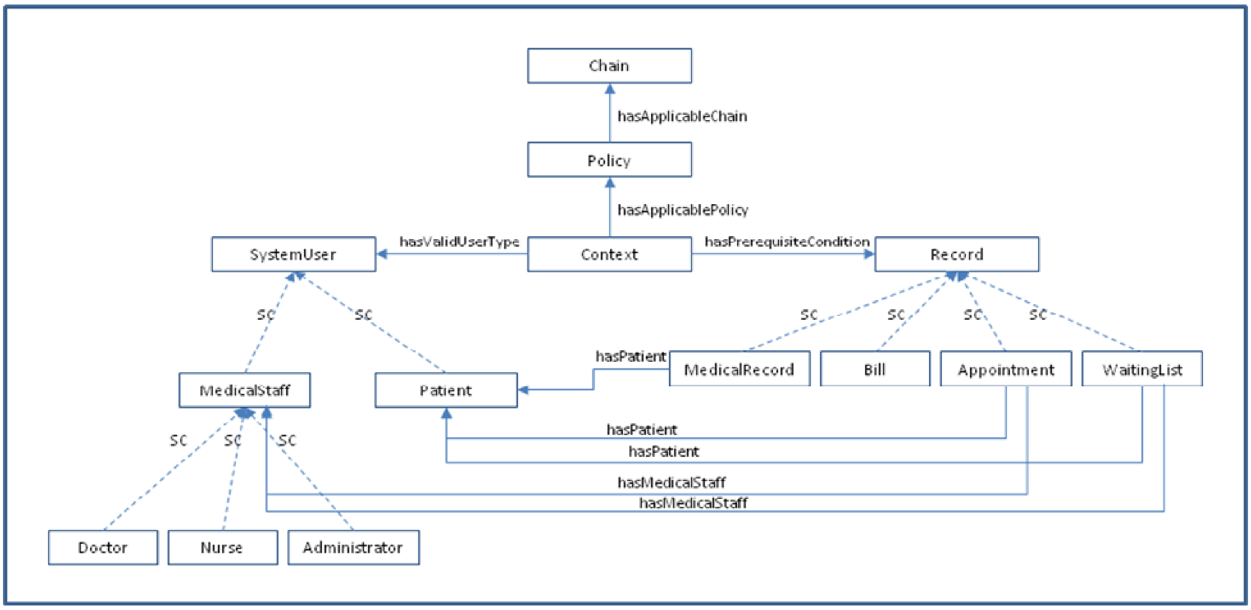
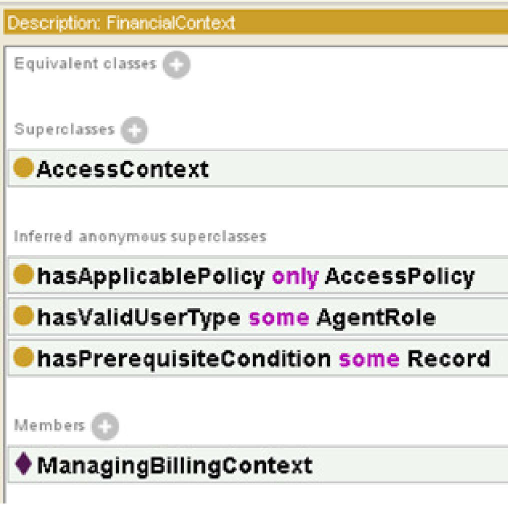


Figure 41: Basic structure of our ontology.

Classes from the list of terms created in the previous step will be used. Most of the remaining terms are likely to be properties of these classes. These terms include, for example, hasContext of, isusedby and isusedfor as the properties for the medical record and hasValidUserType for classes like Admin as shown in this part of the ontology shown in the following figure extracted from the Protégé OWL editor.



Description: FinancialContext

Equivalent classes

Superclasses

● AccessContext

Inferred anonymous superclasses

● hasApplicablePolicy only AccessPolicy

● hasValidUserType some AgentRole

● hasPrerequisiteCondition some Record

Members

◆ ManagingBillingContext

Figure 42: Example on Properties in the ontology.

These properties become slots attached to classes. Thus, the Record class for example will have the following slots: isUsedby and is usedfor.

Some systems allow specification of a minimum and maximum cardinality (e.g. the value of a name slot (as in “the name of a patient”) is one string. That is, name is a slot with value type String) to describe the number of slot values more precisely. Minimum cardinality of N means that a slot must have at least N values. For example, a doctor should have a maximum of one patient at each appointment, and a patient should have minimum of one address to be registered.

After defining a considerable number of new classes, it is helpful to stand back and check if the emerging hierarchy conforms to guidelines in (Natalya et al., 2003).

The class hierarchy represents an “is-a” relation: a class A is a subclass of B if every instance of B is also an instance of A. For example, MedicalRecord is a subclass of Record. Another way to think of the taxonomic relation is as a “kind-of” relation: MedicalRecord is a kind of Record.

A common modelling mistake is to include both a singular and a plural version of the same concept in the hierarchy making the former a subclass of the latter. For example, it is wrong to define a class Doctors and a class Doctor as a subclass of Doctors. Once you think of the hierarchy as representing the “kind-of” relationship, the modelling error becomes clear: a single doctor is not a **kind of** doctors. The best way to avoid such an error is always to use either singular or plural in naming classes. In this thesis the researcher uses singular naming for classes such as: doctor, record etc.

Some of the relations in this thesis have been made transitive. For example, a class AgentUser is defined, and then a class AgentRole as a subclass of AgentUser is defined. Then a class Doctor is defined as a subclass of AgentRole. Transitivity of the subclass relationship means that the class Doctor is also a subclass of AgentUser.

To summarise the steps that have been followed to construct this ontology are:

- defining classes in the ontology,
- arranging the classes in a taxonomic (subclass–superclass) hierarchy,
- defining slots and describing allowed values for these slots,
- Filling in the values for slots for instances.

The validity of the developed ontology will be tested using it in the application for which it has been designed for, as will be shown on next sections.

6.3.1 Ontology Mapping

A central concept in this system design is linking the relational database that has been made to the ontology that has been developed. For this purpose a kind of software called ODEMapster2 has been used.

ODEMapster2 was used in defining these mappings, first using the R2O schema, but this was then changed to R2RML (W3C, 2011) upon the advice of the colleagues in Madrid, as ODEmapster2 was considered to be a more accurately defined standard. The R2RML mapping format is expressed as RDF graphs, which was very helpful in simplifying the work.

Figure 43 shows the mappings made to link the relational database to the ontology. This illustrates how the tables in the database directly relate to classes in the ontology.

Each table is defined as containing a list of objects; this creates a list of individuals within the ontology. Then, the relationships between tables are defined as being properties of these objects.

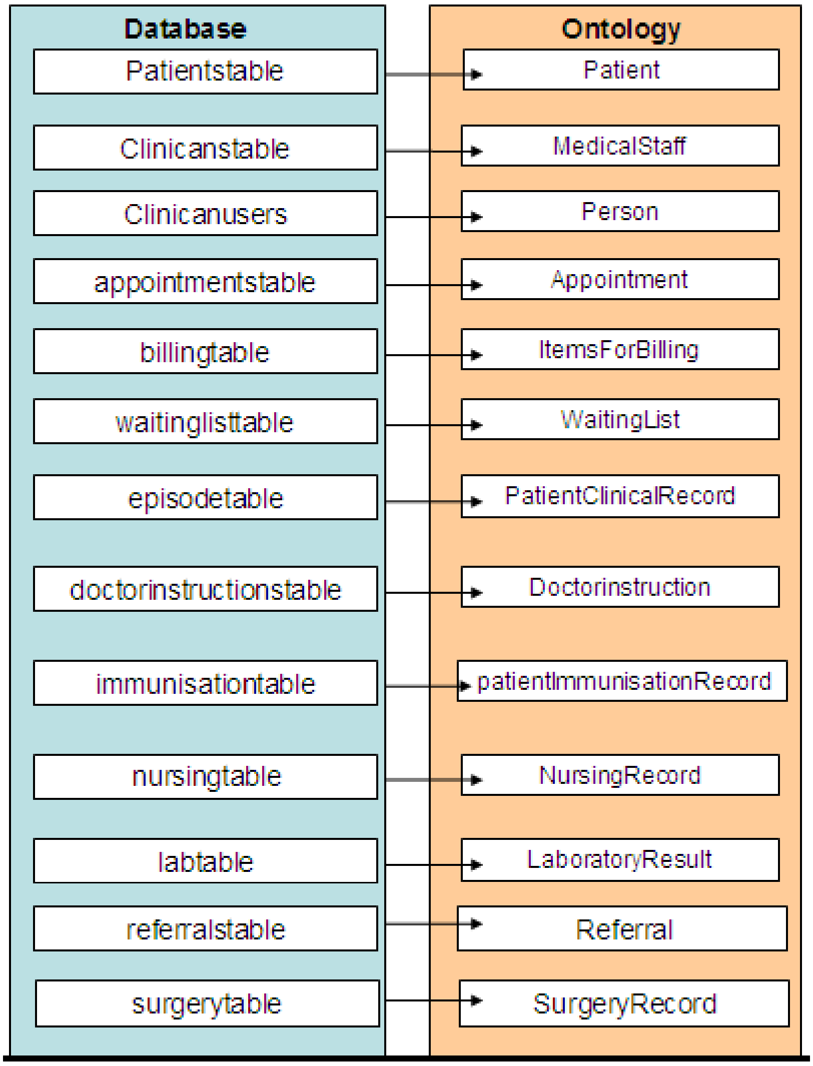


Figure 43: The mapping between the ontology and the database.

As shown in the figure above, the researcher has connected the main concepts from the ontology to the main entities of the database. This would help the reasoner and the system works effectively as will be shown on Chapter 7.

6.3.2 The Jena Framework

One of the key features of this system is that it needs to integrate a number of subsystems. So in order to get use of the ontology, there is a need to import Jena v2.6.4 as a library in Eclipse. JENA is a Semantic Web Rule Language. The purpose of using JENA in this project is creating a mapping between the RDF file created and OWL ontology created.

Jena was then used to import and manipulate ontology OWL files inside the Java code.

The methods for doing this are so clear and straight forward (see appendix for the java code details).

Getting the defaultNS (namespace) right is very important. If it is mixed up, the files would successfully be loaded into the model but then this causes a silent issue when creating the bridge, it does not produce any visible errors. This has been fixed by using the same namespace when creating the RDF file (using ODEMapster), and the OWL ontology.

"**http://www.q8onto.org/healthcareOntology/**" is used as the project namespace.

6.4 ODEMapster

A complete discussion and code used for ODEMAPSTER can be found in Appendix.

6.5 The Semantic Web

As mentioned in Chapter 3, semantics is the study of meanings. It is needed in this thesis to define the prerequisites and chains for the system. This is necessary because the aim is to retrieve data which can be understood and shared across many applications. This will eventually enable data access management systems to not only display information, but also understand this information, gather other relevant data and make decisions based on this. This would help significantly in making the work of the database administrators more efficient (Berners-Lee, 2001). In order to achieve this goal, a number of technologies are currently under development. These include RDF, Ontologies and SPARQL. And the researcher will integrate all of these technologies in order to produce our new data access management system.

6.5.1 Semantic Reasoner

The reasoning API is encapsulated in the edu.stanford.smi.protegex.owl.inference package. The main classes that will be used are the ReasonerManager (used to obtain a reasoner) and Protégé OWLReasoner (an interface to the external DIG reasoner.

Usually, the first step when using the reasoning API is to obtain an instance of Protégé OWLReasoner for an OWL model. This instance of the reasoner can then be used to obtain inferred information about the model such as inferred super classes, inferred equivalent classes, and inferred types for individuals. The Protégé OWLReasoner manages communication with the external DIG reasoner, ensuring that it is always properly synchronised with the internal Protégé-OWL model.

In order to get an instance of Protégé OWLReasoner for an OWL model, the ReasonerManager, which is found in the edu.stanford.smi.protegex.owl.inference.protegeowl package, must be used.

The ReasonerManager is a singleton class (a class with only one instance), whose instance can be obtained by using the static method getInstance().

The getReasoner(OWLModel kb) method on the ReasonerManager can be used to obtain the reasoner for the specified OWLModel. This method will return the same instance of Protégé OWLReasoner for a given model each time it is called. This method of reasoning is preferred since the default implementation of Protégé OWLReasoner only synchronises the external DIG reasoner with the Protégé-OWL model when necessary (when changes have occurred to the OWL model) - creating a new reasoner every time will cause the external DIG reasoner to be resynchronised every time, which could be costly in terms of time for large ontologies.

Once the connection to a reasoner has been established, the reasoner can be queried for information about the ontology.

6.6 JAVA Graphical User Interface

From these design diagrams, it was now possible to start developing the system as it would appear in the form of Java classes. The prototype system was divided into five packages. These are:

Model: contains the back end of the system. These classes deal with overseeing the running of the system back end, which includes calling the semantics, and then retrieving and interpreting the data received.

View: Contains the GUI of the system.

Control: Deals with interpreting events from the GUI as triggered by the user, and in turn driving the model to produce the required data to be displayed in the GUI.

Semantics: Contains the decision-making engine of the system. Classes in this package rely on the use of ODEMapster2 and Jena in order to interface with the domain ontology and database.

Consts: A package dedicated solely to defining constants for use in the system internally. These include constants for identifying the GUI components, as well as variables that form part of the system configuration (location, username and password of the database for example).

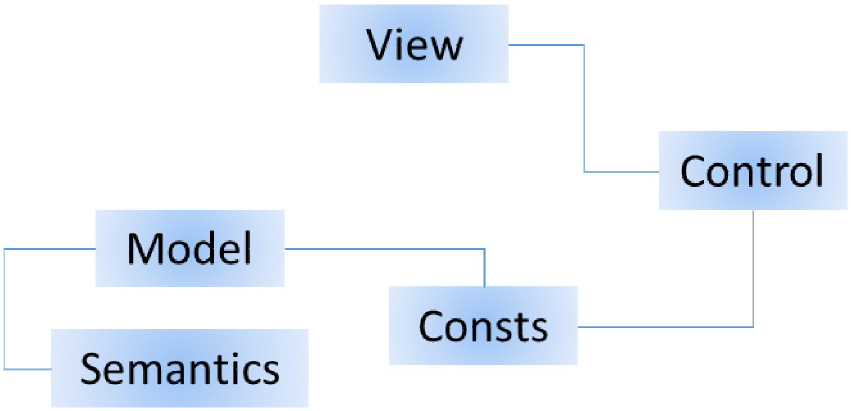


Figure 44: Main Java classes of the system.

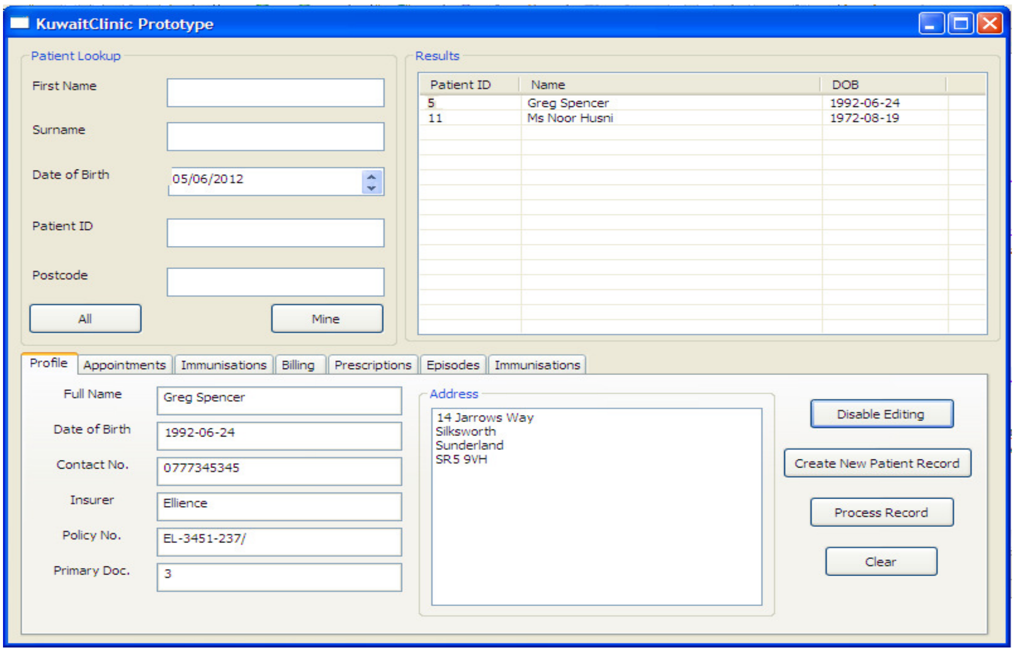
Figure 45 is a snapshot for the system that shows how the final GUI is presented by the prototype. Simple Widget Toolkit SWT has been used in order to implement the system interface. This involved importing an SWT library into the project. This also presented an issue, as SWT requires a specific library depending on the host operating system (i.e. Mac/Windows/Linux). As a result, it was necessary to either:

1) Include multiple versions of the SWT library in order to support multiple operating systems.

Or,

2) Package multiple versions of the prototype to support different systems.

Using approach number one would involve producing a prototype that was unnecessarily bloated, and option two would involve providing a program for each specific OS. In a real case scenario, the difference between the two options would decide the correct version at the point of installation, or installing a more bloated program which would then take more disk space. It has been chosen to package multiple versions for each OS, as this was easier for the sake of a prototype than detecting the OS and loading the correct library at runtime**.** The figure below shows a view of the final GUI as it appears in the prototype.



First Name
Surname
Date of Birth: 05/06/2012
Patient ID
Postcode

Patient ID	Name	DOB
5	Greg Spencer	1992-06-24
11	Ms Noor Husni	1972-08-19

Full Name: Greg Spencer
Date of Birth: 1992-06-24
Contact No.: 0777345345
Insurer: Ellience
Policy No.: EL-3451-237/
Primary Doc.: 3

14 Jarrows Way
Silksworth
Sunderland
SR5 9VH

Figure 45: Final Java GUI.

6.7 Testing the System Implementation

In this section and Appendix I, an oversight will be provided on how each component was tested, and the rationale behind each set of tests. In a recent paper, (Liu and Tan, 2009) have stated on the subject, the "for a system with ordinary complexity, the number of input conditions can be very large". This statement supports the common belief that it is simply not feasible to test every single possible input to ensure the correct behaviour of the program.

Instead, it is important that one carefully selects a specific set of tests that will focus on identifying cases where the tests are most likely to fail (Hayes and Offutt, 1999). Over the following section and Appendix I, overview will be given how each component was tested, including the results of these tests.

Testing the integrated system as a whole in real application

After each part of the system has been tested separately, the performance of the system as a whole will be examined in real application. The first thing that shows up when the researcher runs the project as a java application is the login screen as shown on Figure 46.

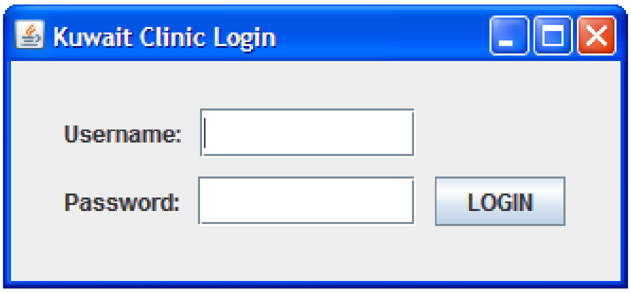


Figure 46: Login screen of the investigated system.

If the user access with his username and password, then the system will check his authority to access which chains by checking the ontology through spiral queries.

Once it checks the validity of the username and password according to the ontology conditions he would be directed to the following interface as shown in Figure 47.

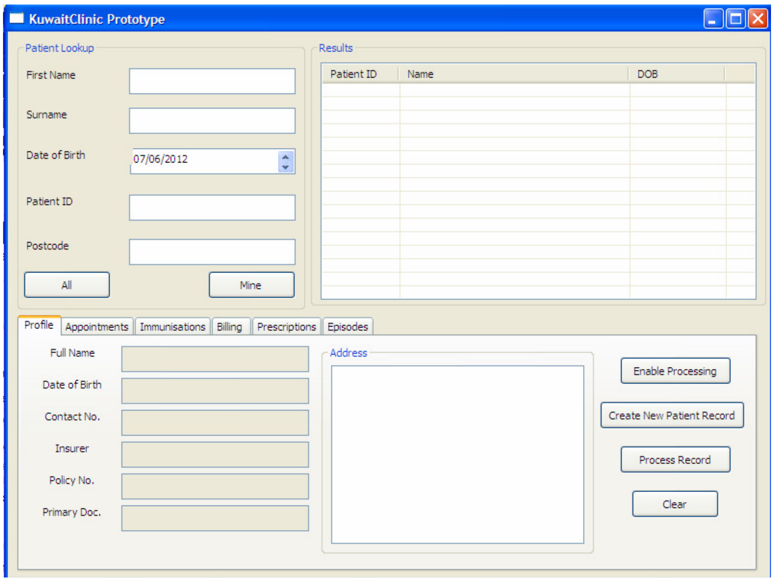


Figure 47: Main GUI of the investigated system.

As shown in the figure above, there is a table called Results. The patient IDs, names and DOB appears in that table according to the context of the authorisation. This context is a condition that has been stated in the ontology and according to them not all the patients that are saved in the database and have relation with that user will appear.

Only those who fulfil the ontology conditions will appear. Also as shown in the figure above, there are buttons at left top and right bottom. Those buttons have been named according to the chain terminologies such as: Mine, process and create. These buttons enabled and disabled according to the conditions in the ontology. The conditions in the ontology can be easily edited in protégé owl editor and saved and then used directly in the system.

6.8 Conclusion

In this chapter design principles that have been set in Chapter 6, are translated into implementation procedures to compose the whole integrated system: starting by analysing the system architecture and overall shape, and finally analysing the implementation of each part separately. The three main parts of the system are:

- Chain and Database
- Ontology
- Java GUI

In addition to technical presentation of the process that have been followed to implement each part. Finally, a series of tests have been done to each part of the developed system and the system as a whole to prove its functionality. On next chapter, the hypothesis will be tested through set of experiments using the system that has been implemented and tested in this chapter.

Chapter 7 Experiments and Evaluations

Recalling back the hypotheses from Chapter 4, it says that the purpose of the present thesis is to improve the data access management by:

- Simplifying the configuration of a data access management system for a given database through a reduction of complexity (H1)
- Increasing the precision with which the algorithm discerns between legitimate and illegitimate access (H2).

In the last two chapters the researcher has presented the design and implementation of the proposed solution. Several experiments were conducted to evaluate the proposed solution by comparison with existing techniques and the experiments and results are reported here. The experiments are divided into two main stages as will be presented in the following sections. In this chapter, the term ChBAC will be used to refer to the Chain that has been designed and implemented in this thesis and distinguish it from Al-Fedaghi's Chain. The Chain that has been proposed by Al-Fedaghi lacks the required parameters and specifications for design and implementation purposes.

7.1 Experiments Overview

As presented in Chapter 4, the experiments were divided into two main parts:

- To validate the first hypothesis (H1) that has been presented on the introduction of this chapter: a set of experiments that compare RBAC to the Chain would be carried to show the simplicity/difficulty of the design of the two methods.
- While for the second hypothesis (H2) another set of experiments to examine the performance (Accuracy /precision, context sensitivity and time for retrieving) of the semantic chain method against RBAC and classical Chain. Consequently there are two main sections in the chapter to present these.

In the next two sections, the two sets of experiments mentioned above will be discussed in detail. First a presentation of the purpose of the experiment and how it is linked to the hypothesis will be given; then a description of the content of the experiments, type of respondents and type of data that will be collected. Details of the data that have been collected, how, with whom, and using what instruments will be given. Finally, discussion and analysis of the results are shown at the end of each set of experiments.

7.2 Ease of Design Experiments

In these experiments the chain method was evaluated against the most common data access control paradigm, RBAC to validate H1. The main contention was that RBAC despite its popularity is complex both in the setting up and its use during run-time and that the Chain method, as claimed in the literature, provides a simpler method (Al-Fedaghi, 2007). As discussed in Chapter 2, the classical Chain method has neither been implemented nor tested to date. Thus the present thesis provides a first application of it and evaluates it against the RBAC method-as both methods provide management to database access in the application level. Before presenting the results, the design of the experiment will be described.

7.2.1 Design of Experiments

As discussed in Chapter 4, this set of experiments was designed to evaluate the required number of: SQL commands, tables and constraints to construct each method for each specific scenario from Table 3. The respondents were asked to write down all the SQL statements, create tables and constraints needed to apply the principles of each method for each specific scenario. The respondents were experts in database administration.

The objective of the evaluation was to compare the complexity of the process of configuring access permissions using the ChBAC method versus the RBAC one. The Hypothesis that relates to the reduction of complexity will be verified in this set of experiments. To evaluate this, the criteria for success will be:

- Reducing the steps of the design

The experiments were conducted to evaluate the scenarios in Table 3 from Chapter four for the following main criteria:

- Number of SQL statements needed to construct each scenario.
- Number of tables needed to fulfil requirements of each scenario.
- Number of constraints needed to fulfil requirements of each scenario.

The respondents were asked to go through the scenarios one by one and implement the necessary restrictions and tables using SQL statements entered from the command line. It was decided to go through the construction of scenarios one by one as in practice if any access policy needed to be changed, the database administrator will go through the scenario independently (each scenario will be treated separately because this what happens in reality, e.g. the database administrator could either register a patient or disclose other patient billing) . In addition, he will need to change the related, tables, SQL commands and constraints. Every database administrator was given the scenarios table (Table 3 of Chapter 3). They were asked to design the database for each scenario (which would require certain number of SQL commands, number of tables and number of constraints). The database administrators used SQL statements for Oracle to create the database such as the following:

```
CREATE TABLE [ IF NOT EXISTS ] table_name
( column_declare1, column_declare2, constraint_declare1, ... )
constraint_declare :: = [ CONSTRAINT constraint_name ]
```

Details of the SQL commands are given in Tables 7 and 8 later in this chapter.

The numbers of SQL commands, tables and constraints received from the different database administrators were almost the same exactly, and in case of different numbers, the number who has been agreed most by the database administrators has been selected.

7.2.2 Technical details of Experiments

The tests were carried out by 5 database administrators as follows:

- 3 database administrators from the IT department in the American University of Kuwait.
- A database administrator from the “International Clinic” hospital in Kuwait
- A database administrator from “Zain Company” for telecommunications.

The respondents were asked to implement the required restrictions using the ChBAC and the RBAC methods and record the number of tables they had to create, as well as the number of SQL statements required and the number of constraints. The reason behind choosing these measures, which may be overlapping to some degree, is to gain some insight into the typical complexity of implementing them from the point of view of the database administrator, as well as the results produced in the database that will affect the complexity with assessing access requests when users try to access the database.

The respondents set up two complete and working designs, one for the RBAC method and the other for the ChBAC method. Their design and feedbacks were recorded in large Excel sheets. Samples of these feedbacks are given in Tables 7 and 8.The complete set of data is summarised in graphs 48, 49 and 50. The five database administrators produced their results in a lab in the American University of Kuwait, in the IT department. The experimental setup consisted of five computer units, each with 1066MHz Quadcore processors and 16 GB of RAM. They used SAN Storage and the database storage was 500 GB (mirrored with raid5) with ORACLE 10g. This software platform was the clients’ preferred back-end. The database administrators were asked to use the basic SQL statements. The respondents were asked to do this in the standard SQL format command prompt so that the researcher would have comparable results.

7.2.3 Results

In this section the researcher presents all the results of the database administrators using the RBAC and Classical Chain methods.

The following two tables go through examples from the results, to show the type of results that one would have in each field of the table.

Scenario	Name of scenario	Number of steps- Number of SQL commands	Number of tables	Number of Constraints	Constraints

1.1	New patient registration	4
		1-Create table patient;
		2-Create table Role-Privilege for administrators;
		3-Insert data;
		4-Insert data;
			2
			1-Patient,
			2-Role-Privilege for administrators;
				3	For each table constraints for: privilege and describing action on that table

1.2	Booking appointments	6
		1-Create table patient;
		2-Create table Role-Privilege for administrators;
		3-Create table appointments;
		4-Insert data;
		5-Insert data;
		6-Insert data;
			3
			1-Patient,
			2-Role-Privilege for-administrators,
			3-Appointments
				7 At least	For each table constraints for: privilege and describing action on that table

1.3	Visit for appointment	6
		1-Create table patient;
		2-Create table Role-Privilege for administrators;
		3-Create table appointments;
		4-Insert data;
		5-Insert data;
		6-Insert data;
			3
			1-Patient,
			2-Role-Privilege for-administrators,
			3-Appointments
				6 At least	For each table constraints for: privilege and describing action on that table

1.4	Emergency case	6
		1-Create table patient;
		2-Create table Role-Privilege for administrators;
		3-Create table A&E;
		4-Insert data;
		5-Insert data;
		6-Insert data;
			3
			1-Patient,
			2-Role-Privilege for-administrators,
			3-A&E
				6 At least	For each table constraints for: privilege and describing action on that table

Table 7: Sample of detailed scenario for RBAC

Scenario	Name of scenario	Number of steps-Number of SQL commands	Number of tables	Number of Constraints	Constraints

1.1	New patient registration	2
		1-Create table patient;
		2-Insert data;
			1
			Patient
				2
				As in the case of the chain the constraints are the same as the chain
					Create, Store
					(As the privilege and action on data are specified by the act of the chain)

1.2	Booking appointments	4
		1-Create table patient;
		2-Create table appointments;
		3-Insert data;
		4-Insert data;
			2
			1-Patient,
			2-Appointments
				4
				For each table two constraints
					Create, Store

1.3	Visit for appointment	4 1-Create table	2 1-Patient,	4 For each table	Create, Store

		patient;
		2-Create table appointments;
		3-Insert data;
		4-Insert data;
			2-Appointments	two constraints	

1.4	Emergency case	4
		1-Create table patient;
		2-Create table A&E;
		3-Insert data;
		4-Insert data;
			2
			1-Patient,
			2-A&E
				4
				For each table two constraints
					Create, Store

Table 8: Sample of detailed scenario for Chain or ChBAC

In Figures 48, 49 and 50, the results agreed by the five respondents are shown each scenario from Table 3. Considering the number of SQL statements, Figure 48 shows that in all cases the ChBAC method required fewer statements, tables and constraints to set up except for two situations 4.1 and 4.2. The results are mirrored for the three measures (Number of SQL statements, Number of Tables and Number of Constraints) with the only exception of the managerial access scenarios (scenarios 4.1 and 4.2) where the results are the same.

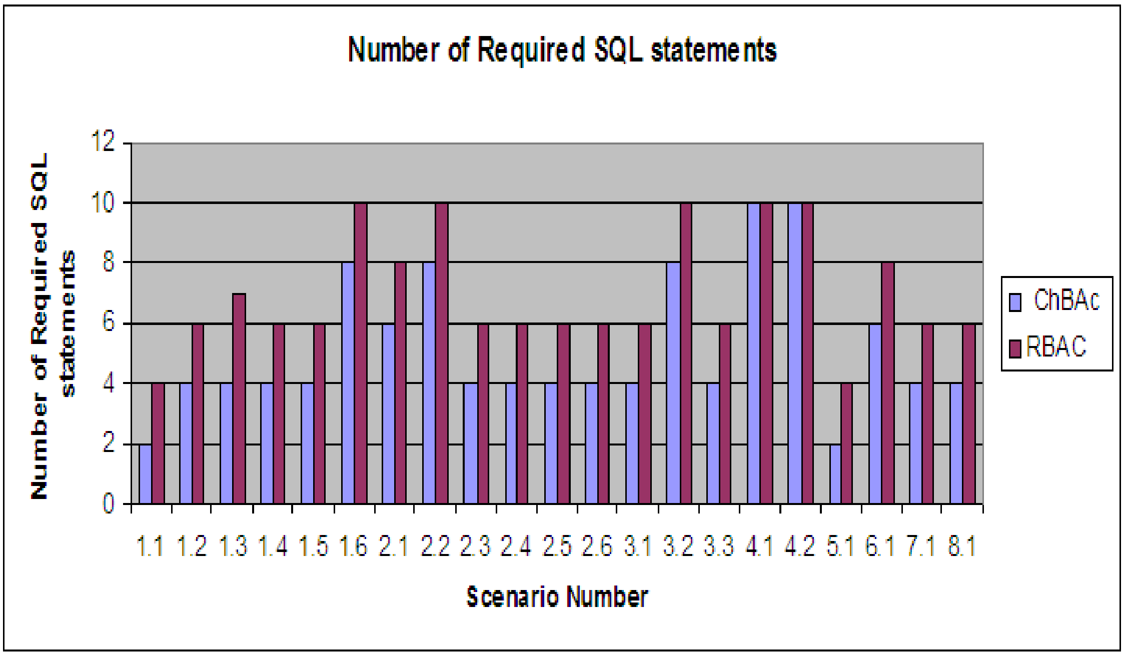


Figure 48: Comparison by Total number of SQL statements.

Figure 48 shows that the number of SQL statements required for the Chain method is by 50% for scenarios 1.1 and 5.1. While the percentage is 60% for scenarios 1.2, 1.4, 1.5, 2.3, 2.4, 2.5, 2.6, 3.1, 3.3, 7.1 and 8.1. The percentage becomes 80% 1.6, 2.2 and 3.2.

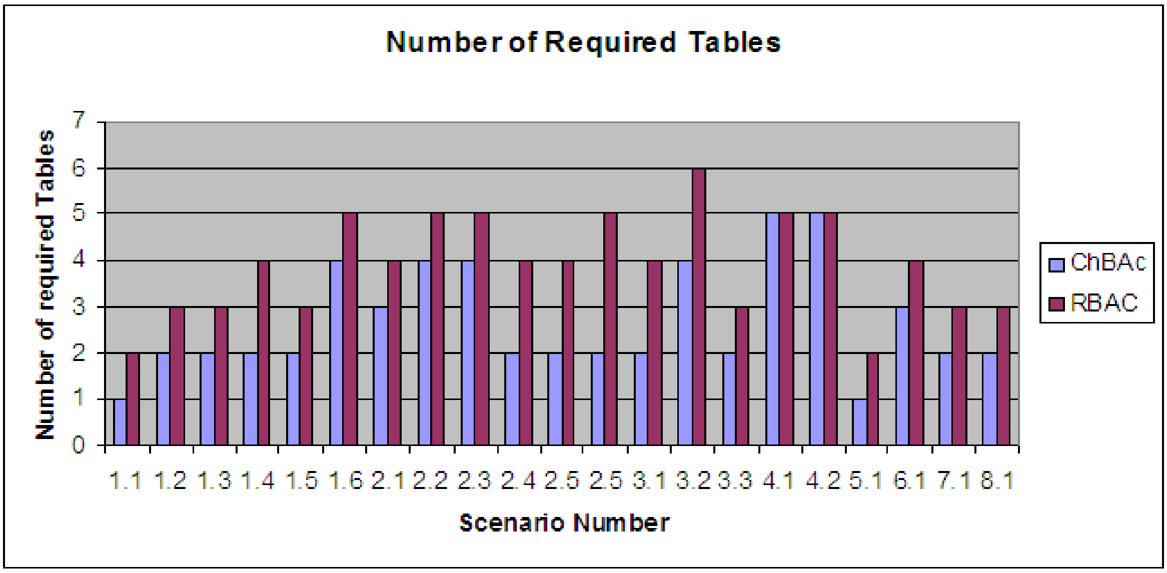


Figure 49: Comparison by Total Number of Tables.

The results presented in Figure 49 refer to the number of tables that had to be created to accommodate the restrictions and show in all but 2 scenarios a reduction of tables for ChBAC as compared to RBAC.

Though the economy (i.e. reduction in number of tables and constraints required) was using 1 table per scenario, yet in some cases 2 or 3 are used. Again there was no difference for the management related scenarios.

Figure 49 shows that number of tables required for the Chain method is less by 50% for scenarios 1.1, 1.4, 2.4, 2.5, 3.1 and 5.1. While the percentage is

60% for scenarios 1.2, 1.3, 1.5, 2.3, 2.4, 2.5, 2.6, 3.1, 3.3, 7.1 and 8.1. The percentage becomes 80% 1.6, 2.2, 2.3 and 3.2.

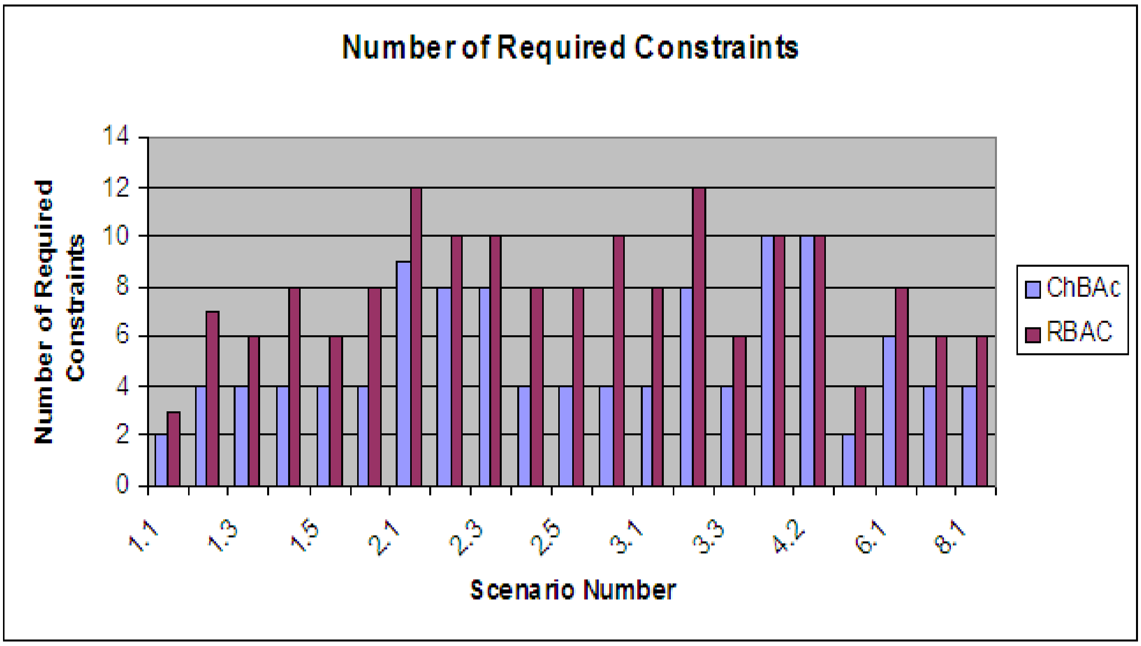


Figure 50: Comparison by Total Number of Constraints.

Figure 50 shows that number of constraints required for the Chain method is less by 50% for scenarios 2.4, 2.5, 3.1 and 6.1. While the percentage is 66% for scenarios 1.3, 1.5, 2.3, 2.4, 2.5, 2.6, 3.1, 3.3, 7.1 and 8.1. The percentage becomes 80% 1.6 2.3 and 3.2.

7.2.4 Analysis and Discussion for the First Set of Experiments

It should be noted that the results presented in Figures 48, 49 and 50 are not independent in that the SQL statements are used both for the setting up of tables and specifying constraints.

There appears to be a definite advantage of using ChBACs over RBAC from a configuration perspective and the researcher expects that if suitably implemented this reduced complexity could also speed up the assessment of privileges as users access the database to retrieve records, though this remains to be demonstrated.

The benefits in terms of the measures presented are also reflected by the participants' responses to an exit poll. This was conducted by running experiments with database administrators. In the experiments differing numbers of records, for differing scenarios, were used across the two methods. Despite being seasoned implementers of RBAC access restrictions the database administrators did prefer the ChBAC method and felt that it was less complex and easier to implement All five of our respondents, when questioned about their views on these two methods following the test implementations, agreed on the potential of the ChBAC method for their work in the database administration of the hospital. They were considering applying the ChBAC method on the system of the new branches of the hospital. They felt that the limited acts would help reduce the time to complete the database design. They were impressed by the fact that setting up the required restriction took them half the time using ChBAC as compared to RBAC as will be shown in the next section.

7.3 Run-time as opposed to configuration

In order to see the effect of this difference in the number of required SQL statements, tables and constraints on the speed of the system and retrieving data, the researcher has developed two prototype Oracle databases using the two methods. The objective of this set of experiments is:

- To measure the time required to retrieve required data for each scenario with an increasing number of records.

7.3.2 Results

The researcher conducted the experiments for the four first scenarios from Table 3 (as they contain all types of functions and privileges that can be performed on a database such as: read, write, edit, delete and transfer) for different numbers of records: 100, 200, 300, 400, 500, 600 and 700. The results were as shown in Figures 51, 52, 53 and 54 below.

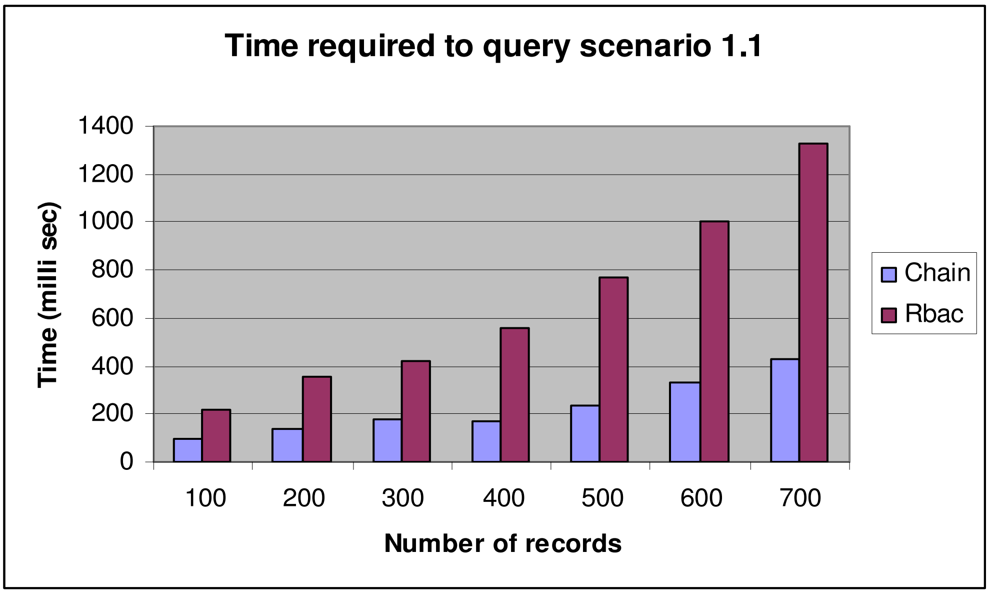


Figure 51: Time required to query scenario 1.1 from Table1.

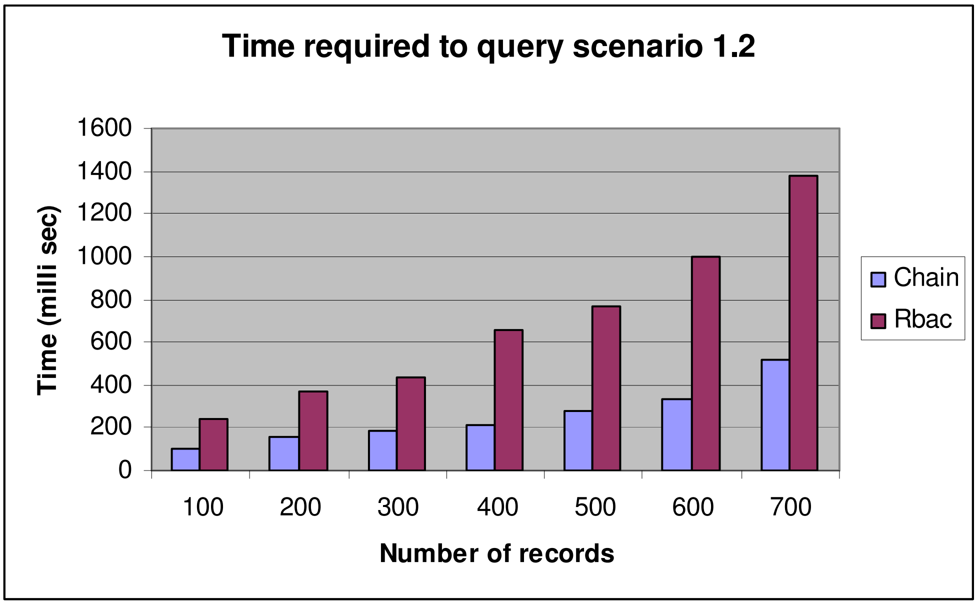


Figure 52: Time required to query scenario 1.2 from Table1.

:

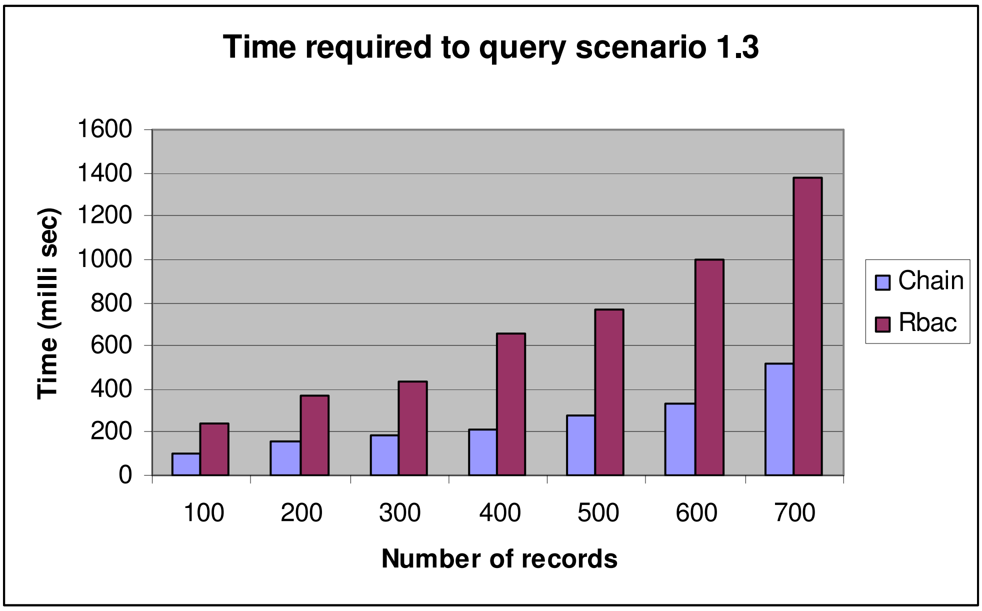


Figure 53: Time required to query scenario 1.3 from Table1.

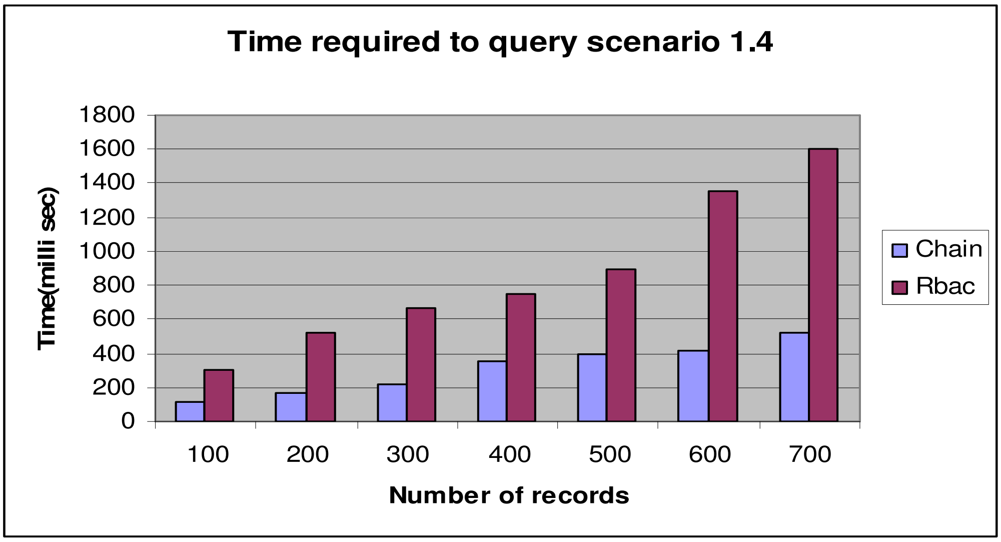


Figure 54: Time required to query scenario 1.4 from Table1.

Those scenarios have been selected carefully to present all type of acts and privileges have been observed from the real hospital. As the real number of patients records were varying from 300-700 a day in the hospital, it has been decided to examine the two methods on the possible number of records from least possible scenario 100 to the maximum which is 700. The researcher has been advised by the expert database administrator to select the most relevant scenarios that usually happen in the hospital to concentrate on them and create a prototype database for them. She has been also advised to repeat each experiment 5 times to check that no machine or human error has happened while recording the results.

The results of the above figures show that the ChBAC outperforms the RBAC in the time required to perform the SQL queries. Chain requires half the time that is required by the RBAC on average. This is because the Chain method needs a lower number of tables, constraints and SQL commands than the RBAC method as shown in the previous section on Tables 7-11.   As shown also in the figures, the time increases exponentially while the number of records increases.

7.3.3 Analysis of the results

The results were carefully examined by the five database administrators on different number of records: 100, 20, 300, 400, 500 and 600 on 4 different scenarios. Each experiment has been repeated five times to assure that the results are really representative of the real case. As seen in Figures 54, 55, 56 and 57 using the chain/ChBAC needed less time than the RBAC by 50-60 % in all scenarios. This would help very much in developing the Chain ontology based system as the time required to translate the SPARQL query to the SQL query and to download the Jena model wouldn't affect the total time of retrieving data from the system, as the ChBAC needs much less time than the RBAC method as shown in the results above. After being the first in the literature to implement and test the Chain/ChBAC idea, but in the world of semantic this wouldn't be enough when different users/agents need common language (agreement on basic concepts definitions) to communicate and share data. Other criteria, which have equal importance to the time of retrieving and simplicity of design, come to the surface. A detailed discussion of these criteria is given in the next section.

7.4 Overview of Semantic Experiments

The first part of the experiments was regarding the simplicity of the design and it was shown that the Chain outperformed the RBAC in this regard. However, according to findings from the literature (see Chapter 2), the outstanding problems from the RBAC and classical Chain have not been solved yet. The most outstanding among these problems is the lack of context sensitivity, flexibility, and accuracy.

Flexibility here means that if you want to add, remove or edit a policy for a user; you need to reconstruct your database. And in this case, Chain would be easier as it requests fewer tables and constraints but it is still impractical to reconstruct your database each time you want to change a policy. On the other hand, context sensitivity means that the data retrieved according to the situation that the user in. So if the doctor who has an appointment would deal with data differently than if he had an emergency case. Finally, accuracy means that the user will get exactly the information he needs, no more and no less. Because, if the user gets less information than he needs he will not be able to perform the task as required, while if he gets more than the required information, the possibility of disclosing sensitive information and the difficulty of auditing this disclosure will increase.

In this section, the researcher will validate the hypothesis “H2” which is:

- ***To increase the precision with which the algorithm discerns between legitimate and illegitimate access.***

The criteria of success for this are:

- The need for fewer parameters to implement the method in the semantic language OWL. This point relates to the simplicity of the design.
- Comparison between the ideal-results user expected from inquiries and the one resulted after using the developed prototype. This point is to rewrite improve the data access management by improving precision (which means more focused set of results for answering a query) of the retrieved data.
- Measuring the time required to assess access requests and retrieve information. To check that improving the precision of the retrieved data did not affect the time of retrieving data and make it slower compared with traditional methods.

In this section, the researcher has established experiments by which the performance of the method under investigation against RBAC can be evaluated in terms of retrieval results. The researcher will show on next sections how the system needs much less parameters to convert its principles in semantic languages than RBAC and TBAC. Then and most important, the researcher evaluates the semantic system with the classical Chain and RBAC methods in terms of a scenario based evaluation where a multi-pronged approach is taken by going through the scenarios with an information retrieval based evaluation and also by involving a user-based blind evaluation to confirm user neutrality towards the three systems.

7.4.1 Design of Semantic Experiments

Our strategy for this set of experiments was to divide it into two main parts:

- Experiments to show the difficulty or simplicity in implementing policies in OWL for three methods: RBAC, TBAC and Chain
- Experiments to show the reliability of the system in handling real scenarios with real users from the hospital to test specific performance criteria

The first set of semantic experiments were carried out with the collaboration of the IBM research manager and are vital to show the relative difficulty in implementing the different methods (RBAC, TBAC and Chain) in OWL.

7.5 Experiments to show the possibility and ease of implementing the chain in semantic language

As mentioned earlier in this chapter, the Chain method proposed by Al-Fedaghi (2007) lacks design and implementation specifications.

Accordingly when the researcher started her experiment with IBM, she found at that she is not able to neither design nor implement the original Chain because of lack of the design parameters. In addition, in order to translate these Chains to the ontology, it needs a clear definition for these parameters and to find its candidates in the other two methods RBAC and TBAC and this was the aim of the following part of the experiments. In addition, examples have been added to demonstrate the meaning of each parameter when it is applied to a real HL7 rule.

In order to set specifications, the first set is to set parameters of each method as will be shown in this section.

For generality, the researcher will walk through the process and show the results for a typical example, the IC and the OSF Healthcare System[1]. The researcher represents and implements it in RBAC, TBAC, and Chains.

As these representations will be required to be translated into a standard form, both for healthcare domain reasons and in order to do a fair comparison, each representation was transformed into OWL[2], which has a direct mapping into HL7.

Representing the three models above in the OWL language, the researcher found that the Chain model was the easiest to be translated as it contains fewer statements and simpler syntax.

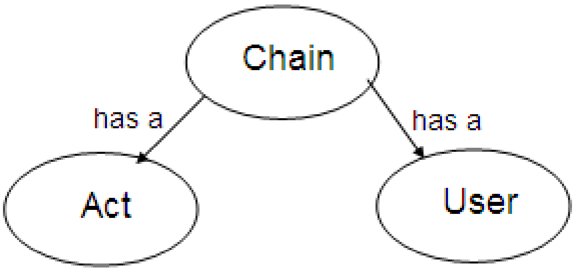


Figure 55: Chain model using OWL.

Figure 55 shows that the result of the transformation process would be of the form:

<User ID>.... <User ID>

<Act ID>......<Act ID>

An example for this:

[1] OSF HealthCare is a multi-state corporation operating facilities in Illinois and Michigan.

[2] OWL (The Web Ontology Language) is a family of knowledge representation languages for authoring ontologies that is endorsed by the World Wide Web Consortium.

<User ID> receptionist1 <User ID>

<Act ID>collect1<Act ID>

Figure 56 shows the refinement process from the RBAC statements to OWL.

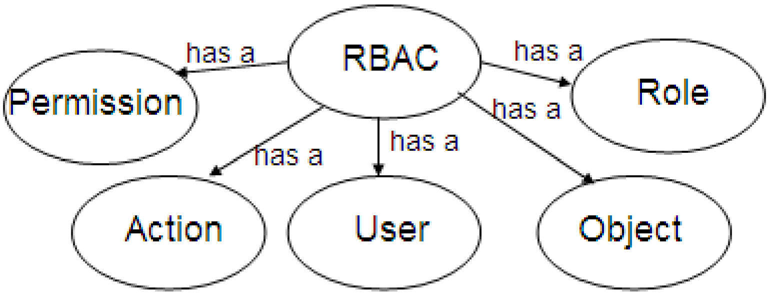


Figure 56: RBAC model using OWL.

The resulting policy statement is of the form:

<User ID> …… <User ID>

<Object>………<Object>

<Action> ……… <Action>

<Role>……<Role>

<Permission> …. <Permission>

An example for this:

<User ID>receptionist1<User ID>

<Object>registration<Object>

<Action> write<Action>

<Role>receptionist<Role>

<Permission>Allow<Permission>

As TBAC is based on RBAC, the translation of the TBAC statements is similar to the translation of RBAC statements but with more requirement policies added to it the command:

The resulting policy statement is of the form:

<User ID> …… <User ID>
<Object>………<Object>
<Action> ……… <Action>
<Role>……<Role>
<Permission> …. <Permission>
<Task> …. <Task>

An example for this:

<User ID>receptionist1<User ID>
<Object>registration<Object>
<Action> write<Action>
<Role>receptionist<Role>
<Permission>Allow<Permission>
<Task> register <Task>

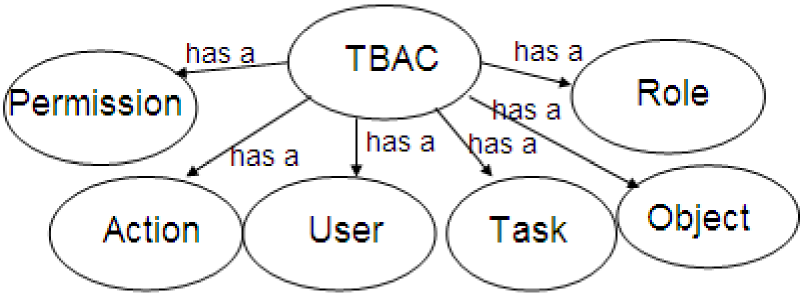


Figure 57: TBAC model using OWL.

The following example shows the privileges that are requested while building a new policy in RBAC. This sequence of privilege requests closely follows the methods that were used to create the policy in question. The policy constructed consists of one rule that has three prerequisites (one of each type) with each prerequisite having a single parameter. The rule is:

In addition, the Chain combines the action and the policy/reason of access in one thing which is the act of the chain.

This means that if one wants to model the three systems using their parameters:

RBAC = (U, O, A, R, P)

Where:

U: Users
O: Objects
A: Actions
R: Roles
P: Set of Permissions

TBAC = (U, O, A, R, P, T)

Where:

U: Users
O: Objects
A: Actions
R: Roles
P: Set of Permissions
T: Set Of Tasks

Chain = (U, A)

Where:

U: Users
A: Acts of Chains

For ease of enforceability, the researcher measures the number of tables that needs to be accessed in the determination of an access or disclosure decision.

7.5.1 Technical Details

The responders for this set of experiments were:

- The database administrator
- Myself
- The experiments with the collaboration of the IBM research manger

The respondents have gone through basic access policies and how to translate them into OWL for the three methods. The results, which have been published in (Omran *et al,* 2010a) and (Omran et al, 2010b), are discussed below.

7.5.2 Results

In Figure 58 and Figure 59, the researcher has chosen statements from the HL7 website and designed the access policies of them for the three methods: RBAC, TBAC and Chain based on the system designs shown above.

Figure 58 shows that the number of required accessed tables in the Chain method is always the minimum (1), while, for this example (OSF Healthcare), TBAC and RBAC needs more implementation parameters. While the effort required may vary in TBAC and RBAC from policy to policy, the trend is that the effort is always more than Chains.

For the effort required in enforcement, this is measured by the work that has to be performed in evaluating the attributes and conditions in a statement (all other things, such as low-level enforcement platform details, being equal).

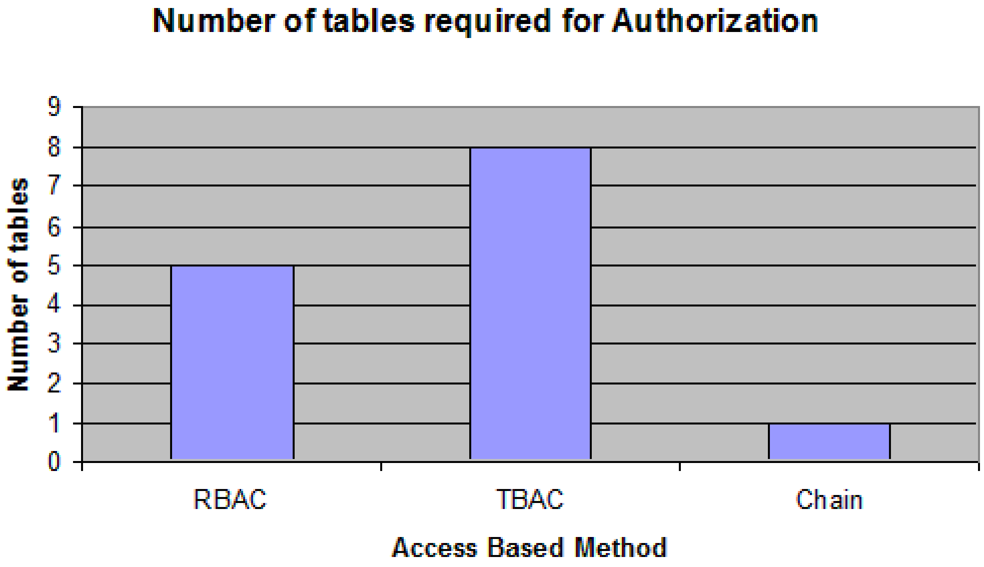


Figure 58: Ease of Enforceability.

Figure 58 shows the re-thinking of the underlying representational model in Chains yields benefits in terms of the number of checks that have to be performed during policy enforcement.

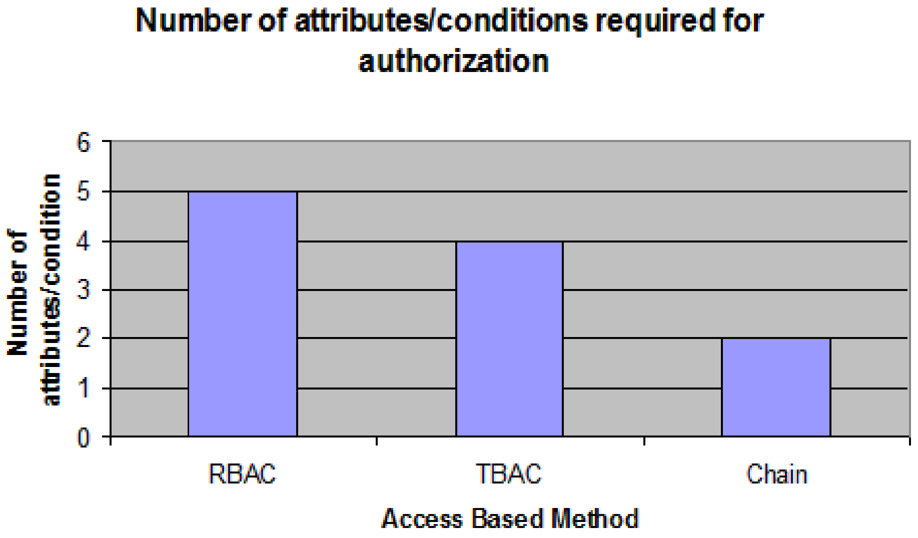


Figure 59: Number of attributes/conditions required for authorisation.

The results conducted from this set of experiments shows the ease of the Chain method implementation in OWL compared to the RBAC and TBAC. Starting from this point, the researcher proceeds with the semantic experiments to the next stage.

7.6 Semantic Experiments for real application

This set of experiments concentrates on issues that differ from the previous sets and are more important to measure. The experiment here to verify whether the methods are able to discern between situations where access needs to be granted from when access should be denied as this is a question of precision.

The objective of this set of experiments is as mentioned in H2:

To show that using the developed system, one can improve data access management through having more accurate results.

In the case of Information system for a hospital, the availability of the information is not the only required factor to ensure efficient service providing. Other critical factors are also crucial such as: precision (more focused results) and accuracy (correct results). Unlike the availability, precision and accuracy are hard to be evaluated. Therefore, the research has decided to make a questionnaire to number of users from a real hospital to examine those two sensitive criteria for carefully chosen scenarios. Each user has been given a set of scenarios to apply them using three prototypes for three systems:

System A: Using Classical Chain method for data access management.

System B: Using Chain ontology based method for data access management.

System C: Using Classical RBAC method for data access management.

The three systems have exactly the same GUI to ensure that the users wouldn't recognise the difference between them which may affect their feedback accordingly. Also the researcher took care of the user trainee after iteration, and for this purpose she put her system to be system "B", i.e. in the middle between the two other systems. . The researcher has optimised the scenarios from Table 3 to the selected tasks, which appear in the questionnaires tables presented in this section, given that if similar processes appear in different scenarios; we select the more comprehensive ones. The questionnaires concentrated on the four main types of users in the hospital: doctor, nurse, admin and database administrator. These users really cover the comprehensive and main critical scenarios for most of healthcare applications that have been covered by the researcher survey. The tasks covered some scenarios shared between different type of users such as doctor and nurse and doctor and admin. But different users need to see different parts of the data according to the context and this was one of the most critical points that the researcher intended to figure out. As she was trying to find how each system (A, B and C) will help the user to get precise and accurate data taking into consideration the context changes.

In order to evaluate these criteria, the researcher has decided to give questionnaires to a randomly chosen set of users from the hospital. The only requirement that she asked for is that this set should contain at least one member of each group of users.

The main scenarios that have been tested are:

For users of type Receptionist/Admin:

- New patient registration
- Booking appointments
- Visit for appointment
- Billing
- Managing Patients

For users of type Doctor:

- Routine Patient Consultation
- Outgoing referral
- Incoming referral

For users of type Nurse:

- Nurse Consultation
- Incoming Referral

For users of type Database Administrator

- Overall database managing
- Changing access policies.

The evaluation for the performance of our system was about four criteria:

- Accuracy
- Correctness
- Context sensitivity/Flexibility of the system
- Relative time to perform a task

7.6.1 Technical Details

The responders for this set of experiments were staff from real hospital and their roles were as follows:

- Three admin/receptionist(s)
- A doctor
- Three nurses
- The database administrator

Then, the questionnaires have been given to those users and experiments have taken place at the hospital on a computer with the following specifications:

Windows Edition: Windows 7 Ultimate
Processor: Intel® Core™ Duo CPU T2450 @ 2.00GHz
Installed memory (RAM): 1.00 GB
System type: 32- bit Operating System
Memory type: DDR2

Those users has been given the tasks on an Ms Word sheet, after giving them a short tutorial by the researcher on how to use the three systems and then asked them to perform the scenarios that have been presented at the bottom of the introduction of 7.6 above (each group of users has been given a questionnaire that contains the basic selected scenario related to their group and which describes the tasks they were to complete).The users then were allowed to complete these tasks an saving their responses to a text file. Also observation by the DA has been told to give assistance or answer questions if needed. After testing the Chain ontology based system, they have been asked to test three anonymised systems: System A: System of Chains without semantics, System B: System of chains with semantics and System C: system of RBAC without semantics. All the questions and answers are presented in this section. The researcher has tried to have the same GUI for the three systems so that not to make any indication for the users about identity of the systems. The users started first with the authorisation experiments, as they were asked to enter user name and password that are related to their group of users. According to this user name and password they were directed to their GUI which contains the rules in the case of RBAC and acts in the case of the Chain method. In the developed system "Chain with semantics" this step pass through the ontology layer to check the authorised acts through properties of the group of users' class. All the experiments results will be presented in next sections. Screen shots for the GUI for different users in different scenarios are placed in the Appendix (figures 63-68).

7.6.2 Results

In Table 3 of Chapter 4, the receptionist/admin is given the authority to create new patient record for no specific condition-except to give valid name, DOB and address. And he is given the authority to access the patients' record to set appointment for the condition that the user has arrived to his specified appointment and want to have new one according to the doctor request-as seen in the appointment tab for a doctor. And the receptionist also can access the billing of a patient given the condition that the record of that patient has been disclosed to this specific admin to pay. Otherwise he will not be able to access them.

The researcher in this thesis has asked three admin/receptionists to go through these scenarios to test the developed system and to get their feedback. The results were as follows:

User	Task	Action	Status
Admin1	Access with username and password	Read information from the java code check with the ontology through Sparql query	Success-As shown in Figure 60-Appendix
Admin2	Access with username and password	Read information from the java code check with the ontology through Sparql query	Success
Admin3	Access with username and password	Read information from the java code check with the ontology through Sparql query	Success
Admin1	Create new profile record for a new patient	Write information to the database	Success
Admin2	Create new profile record for a new patient	Write information to the database	Success
Admin3	Create new profile record for a new patient	Write information to the database	Success
Admin1	Process/edit the information of an existing patient	Read and write information to the database	Success-As shown in Figure 61-in Appendix
Admin2	Process/edit the information of an existing patient	Read and write information to the database	Success
Admin3	Process/edit the information of an existing patient	Read and write information to the database	Success

Admin1	Check the billing of a specific patient	Read information from a database	Success
Admin2	Check the billing of a specific patient	Read information from a database	Success
Admin3	Check the billing of a specific patient	Read information from a database	Success-As shown in Figure 62-Appendix

Table 9: Admin(s) feedbacks on the questionnaire-part1

After checking the ability of the Chain ontology based system to perform the basic required functions for different Admin tasks, the researcher has then asked the three admin to do the same actions to three systems: System A, System B and System C.

- System A: System of classical Chains without semantics
- System of chains with semantics
- System of RBAC without semantics.

The evaluation was according to their answers to the following questions:

Question	User	System A	System B	System C
1- Is there a delay in Accessing the system at the login	Admin1	No	No	No
	Admin2	No	No	No
	Admin3	No	No	No
2-Did you have any problem or experience any delay while creating the new patient profile	Admin1	No	No	No
	Admin2	No	No	No
	Admin3	No	No	No
3-Did you have any problem in editing the patient record	Admin1	No	No	No
	Admin2	No	No	No
	Admin3	No	No	No
4-If you want to search the profile of a specific patient and you are not sure about the patient ID nor name but you know	Admin1	All the 30 patients' records.	I get four patients-including the one I search for –according to the system condition	All the 30 patients' records.

he has a condition of: he has appointment today with the doctor and you need to press the all button what you will have?			Noor Husin Paula Jones Greg Spencer	
	Admin2	The system is retrieving all the 30 patients records	I get four patients-including the one I search for –according to the system condition	The system is retrieving all the 30 patients records
	Admin3	The system is retrieving all the patients record	I get four patients-including the one I search for –according to the system condition	The system is retrieving all the patients record
5-Can you edit all patient fields in the billing tab?	Admin1	Yes-as shown in Figure 61	Yes-except the description-its transformed from the doctor-as shown in Figure 60 the description field is disabled (grey colour)	Yes-as shown in Figure 61
	Admin2	Yes	Yes-except the description-The attribute is locked	Yes

	Admin3	Yes	Yes-except the description-I can’t edit it but I can read it	Yes
6-If you want to search the bill of a specific patient-who didn’t pay- and you are not sure about the patient ID nor name and you need to press the all button what you will have	Admin1	The system is retrieving all the 30 patients records	The system is retrieving only the 6 patients with specific condition-who didn’t pay	The system is retrieving all the 30 patients records
	Admin2	The system is retrieving all the 30 patients records	The system is retrieving only the patients -who didn’t pay	The system is retrieving all the 30 patients records
	Admin3	The system is retrieving all the 30 patients records	The system is retrieving only the patients -who didn’t pay	The system is retrieving all the 30 patients records
7-What is the time you get from the timer for the login	Admin1	100 ms	109 ms	108 ms
	Admin2	98 ms	110 ms	110 ms
	Admin 3	98 ms	110 ms	110 ms
8-What is the time you get from the timer for	Admin1	190 ms	219 ms	240 ms
	Admin2	192 ms	219 ms	238 ms

retrieving the billing information	Admin3	192 ms	219 ms	240 ms

Table 10: Questionnaire given to admins and their answers.

7.3 Analysis of Results Related to Admin Feedbacks

The reason behind choosing the tasks on Table 9 is to test the validity of the system to cover required daily hospital system functions. The results show that the system can read from the ontology, can translate the sparql queries and finally write to the database.

While questions given on Table 10 were to evaluate the criteria of: Accuracy, Context Sensitivity and Time required to retrieve data. The questions have been carefully set to reflect real scenarios that take place in the hospital and on the same time measuring the above criteria of performance for the three systems.

Below is a description given to show questions related to evaluate each criterion. The expected (ideal) answer is also given to be compared to the results collected from the users.

1- Accuracy:

The questions that have been put to evaluate this criteria for the three systems where: Q4, Q5 and Q6.

The expected result For Q4 is: Mohammed Imran, but the doctor has appointment today with the following patients:

Mohammed Imran
Noor Husin
Paula Jones
Greg Spencer

This means that system “B” has reached the expected result according to the condition given by the user.

The expected result for Q5: According to Table3 in Chapter 3, the admin can access the billing fields but the description is related to the doctor to decide it. This also mean that the only system which has meet the expected function is system “B” as it gives only read capability for admin type of users.

The expected result for Q6: Noor Hussni But 6 patients didn’t pay also including Noor Hussni. So the system “B” has retrieved the data according to the condition given by the user while the other two retrieved the whole set of patients.

System (B) shows more accurate results according to conditions that have been added to the ontology and hard coded in the java code to narrow the spectrum of the search results and be more specific which makes the work of the admin easier and more accurate.

2-Context sensitivity:

The questions that have been put to evaluate this issue for the three systems where: Q4, Q5 and Q6.

It’s obvious from the results retrieved in this part and the previous part that system (B) is changing with the context and affected by it: the context is: appointment, billing, registration and patient profile editing. This is not the case of the other two systems as they give access rights according to the role only without affecting by the context. This result is not surprising according to the discussion of the disadvantages of RBAC and classical Chain that have been given in Chapter 2.

3-Time required to perform the queries:

The questions that have been put to evaluate this issue for the three systems where:

Q1, Q2, Q7 and Q8.

The data should be retrieved without making the user experience a delay.

The researcher was looking for a system that has semantics without a significant delay compared to other systems and this is exactly what she got as the users didn't recognise a delay while using the system, in addition this system has overcome the time of the RBAC in some cases-as the time required to translate the Sparql query has not affected the time of the system considerably because of three things: 1- there is chain based for the database design, which means less table and constraints to consider 2-not all attributes and tables need to be checked in this system-just the one related to the context 3- in the case of the read from the database, the system deals only with snapshot from the database

The following table presents clearer analysis of Table 10. The table show comparison between the three systems for the given criteria. Symbol ☑ means the system meet this criterion and ☒ means it doesn't. Also average time to fulfil scenarios on questions 7 and 8 (of three iterations for each user) is given.

Criteria to be evaluated	Questions related to this criteria from Table 11	Does the system meet the criteria of evaluation for each question?		
		System A	System B	System C
1- Accuracy	Q4	☒	☑	☒
	Q5	☒	☑	☒
	Q6	☒	☑	☒
2-Context sensitivity	Q4	☒	☑	☒
	Q5	☒	☑	☒
	Q6	☒	☑	☒
3-Time required to perform the queries	Q1	☑	☑	☑
	Q2	☑	☑	☑
	Q7	Avg.=98.66 ms	Avg.=109.667ms	Avg.=109.33 ms
	Q8	Avg.=191.33 ms	Avg.=219.33 ms	Avg.=239.33ms

Table 11: Analysis for data in Table 10-receptionist feedback.

As can be seen from the above table, that system (B) is showing significant improvement in the accuracy and the context sensitivity while keeping a competitive time of retrieving. In order to evaluate these factors in more complicated situations were two different types of users are working with the system, the researcher gave the questionnaire for two nurses and one doctor and their answers have been recorded in Table 12.

7.6.4 Analysis of Results Related to Doctor and Nurses Feedbacks

Then the researcher moves on to the next phase of the real scenario experiments. In this phase, the researcher asks one doctor and two nurses to answer the following questionnaire.

Question	User	System A	System B	System C
1- Can you check the profile of the patient you're searching for	Doctor	Yes-as shown in Figure 65	Yes-as shown in Figure 66	Yes-as shown in Figure 65
	Nurse 1	Partially-DOB and address	No-only his name -as shown in Figure 66	Partially-DOB and address
	Nurse 2	Some-DOB and address	only his name	Some-DOB and address
2- Can you edit the profile of the patient you're searching for	Doctor	No	No	No
	Nurse 1	No	No	No
	Nurse 2	No	No	No
3-Can you check all the attribute in the patient immunisation	Doctor	Yes	Yes	Yes
	Nurse 1	Yes	Yes	Yes
	Nurse 2	Yes	Yes	Yes
4-Can you update all	Doctor	Yes	Yes	Yes

the patient immunisation attributes	Nurse1	Yes	No-only status of the immunisation -as shown in Figure 68-only status is editable	Yes
	Nurse2	Yes	I can’t-only status of the immunisation	Yes
5-What is the time you get from the timer for the login	Doctor	110 ms	187 ms	190 ms
	Nurse 1	80ms	94 ms	92 ms
	Nurse 2	82ms	93 ms	92 ms
6-What is the time you get from the timer for retrieving the immunisation information	Doctor	168 ms	219 ms	230 ms
	Nurse 1	168 ms	219 ms	230 ms
	Nurse 2	168 ms	218 ms	232 ms

Table 12: Questionnaire for doctor and nurses and their answers.

7.6.5 Analysis of Results Related to Doctor and Nurse Feedbacks

Below are the criteria that have been evaluated through this set of experiments:

1- Accuracy:

Questions that have been put to evaluate these criteria:

Q1, Q2, Q3 and Q4

For Q1, the expected result is: According to Table 3 in Chapter 3, the doctor has the authority to access the full patient profile, but the Nurse doesn't need to access only the patient name from patient profile. It's found that system "B" is the only system which meets this criterion as it gives authority to the doctor that differs from that given to the nurse according to the context which gives more accuracy to the retrieved data.

For Q2, the expected result is: The Doctor and the Nurse shouldn't be able to edit the patient profile. This is what we get from system "B" which satisfies the hospital policy.

For Q3, the expected result is: To deliver better services, both doctors and nurses should be able to check all the fields of the immunisation tab. This is what we get out of the three systems.

For Q4, the expected result is: Only doctor can edit all the immunisation fields but the nurse shouldn't be able to do that, she needs only to edit the status of the immunisation which is her job.

System (B) showed more accurate results according to conditions that have been added to the ontology and hard coded in the java code to narrow the spectrum of the search results and be more specific which makes the work of the admin easier and more accurate. As for tables that can be viewed by both doctor and nurse, not all the information should be editable and viewable for both of them. And this is what system (B) offer.

2-Context sensitivity:

Questions that have been put to evaluate this criterion:

Q1, Q2, Q3 and Q4

System (B) is changing with the context and affected by it: the context is: Immunisation (can be fully edited and viewed by the doctor and partially by the nurse), patient profile: can be viewed by the doctor but without being able to edit it and only patient name is viewable by the nurse.

3-Time required to perform the queries:

Questions that have been put to evaluate this criterion:

Q5 and Q6

The data should be retrieved without making the user experience a delay.

The researcher was looking for a system that has semantics without a significant delay compared to other systems and this is exactly what she gets as the users didn't recognise a delay while using the system, in addition the system has overcome the time of the RBAC in some cases-as the time required to translate the sparql query has not affected the time of the system considerably because of three things: 1- there is chain based for the database design, which means less tables and constraints to consider 2- not all attributes and tables need to be checked in the system-just the one related to the context 3- in the case of the read from the database the researcher deal only with snapshot from the database .

Criteria to be evaluated	**Questions related to this criteria from Table**	**Does the system meet the criteria of evaluation for each question?**		
		System A	**System B**	**System C**
1- Accuracy	Q1 Q2 Q3 Q4	☒ ☑ ☑ ☒	☑ ☑ ☑ ☑	☒ ☑ ☑ ☒
2-Context sensitivity	Q1 Q2 Q3 Q4	☒ ☑ ☑ ☒	☑ ☑ ☑ ☑	☒ ☑ ☑ ☒

3-Time required to perform the queries	Q5
	Q6
		Avg.=90.667 ms
		Avg.=168 ms
			Avg.=125 ms
			Avg.=218.667 ms
				Avg.=124.67 ms
				Avg.=230.667ms

Table 13: Analysis for data in Table 12

This set of scenarios has verified the evaluation from Table 12 that this system is showing significant signs in the way of semantics because it shows signs of accuracy preserving, context sensitivity while providing good time for retrieving.

7.6.6 Results Related to Database Administrator Feedback

Then the researcher asked the database administrator to change some of the policies for the three systems as follows:

Task	System A	System B	System C
1- What do you need to do if you want to add authority for the admin/receptionist on the patient profile	Need to audit the SQL statements in the patient table and change and add SQL statements	Only change one statement in the admin class in the protégé OWL editor-add property to that class	Need to audit the SQL statements in the patient table and the policy/user table and change and add SQL statements
2- What do you need to do if you want to add authority for the admin/receptionist on the immunisation profile	Need to audit the SQL statements in the patient table and the immunisation table and the nursing table and change and add SQL statements	Only change one statement in the admin class in the protégé OWL editor add property to that class	Need to audit the SQL statements in the patient table, the policy/user table, the immunisation table and the nursing table and change and add SQL statements

Table 14: Database Administrator feedbacks on the questionnaire.

7.6.7 Analysis of Results Related to Database Administrator Feedback

There is still one criterion to be highlighted about developed system which is “Flexibility”. In Table 14 the researcher asked the database administrator to change two conditions and see the effect of that on the three systems. It is clear that system (B) is easier to change the conditions in as most of the cases you don’t need to go and change the construction of the database as it’s the case in system “C”. As in system “C” you need to go and change the conditions saved in the table of the Role/policies and also change the conditions of the attribute of the related tables. But in the case of system “B” the changing of the conditions is controlled from protégé OWL-and the researcher tries to put instructions inside the ontology so that changing the policies would be easier. An email has been received from the hospital database administrator that shows his feedback and recommendation to use the developed method in the hospital according to the flexibility and the ease of changing the policies while preserving the privacy of the data.

In this chapter a presentation of the experiments that have been carried through the years as raw data with primary analysis. In the next section, Findings and summary of the experiments will be highlighted.

7.7 Findings and Summary of the Experiments Analysis

The experiments in this chapter have been designed according to the hypothesis and criteria of success presented in Chapter 4. The three main points that have shaped the experiments were:

- The ease of the design and implementation for a reliable data access management method related to H1;
- The precision of the data retrieved by the developed method related to H2;

- The time required for data retrieving which is a result of the ease of design-first bullet.

Accordingly, the experiments have been designed, implemented, recorded and analysed.

To verify the first criteria of success which is related to H1, the ChBAC and the RBAC have been implemented by a set of expert data base administrators. Results showed a clear advantage towards the ChBAC compared to the RBAC. The criteria that have been tested in this set of experiments:

- Number of required SQL statements to implement the two methods,
- Number of required tables to implement the two methods.
- Number of required constraints to implement the two methods.
- In addition to the required time to retrieve the information.

The recorded results have shown that the ChBAC needed half number of SQL statements, number of tables, constraints and time for retrieving in almost all the scenarios. This gives the advantage to the ChBAC over the RBAC in the first and the third criteria of success above.

But according to the results that the researcher has got out of these experiments, none of the two methods provides a solution for the second criteria which is precision. This raises the importance of integrating the semantic to the chain or ChBAC principles. But before going further to the second criterion of success, the researcher needs to check that this integration will not affect the first criterion which is the ease of design. And this was the reason behind carrying the second set of experiments related to the ease of implementing the principles of the RBAC, ChBAC and TBAC in semantic language. Because the chain has less parameter to be implemented, the results have shown that its implementation in OWL was much easier and straight forward than the TBAC and RBAC.

Then, the experiments have been transformed into another critical phase, where precision is the criterion of success which is related to H2. And for this purpose, consultation has been asked from experts from IBM Company, Trento University, Eindhoven University and Madrid University in order to design the required experiments.

It has been agreed that users' feedback is needed. Also, those users need to be given carefully chosen scenarios to test the reliability of the system in retrieving accurate information in context sensitive manner. The feedbacks of the users were very positive, as they have noticed the difference in the precision of the data retrieved by the developed system.

The system shows also flexibility in working in the different situations by focusing more on the required data.

The insights gained from the user study are that the Chain ontology based method:

- is simple and precise in policy specification;
- is flexible in its expressiveness;
- uses less tables and conditions than the system currently used at the healthcare provider;
- is faster than the RBAC method.

The conclusion of the study was that the Chain ontology based method was a simpler and clearer way for preserving database privacy without losing the highest standard of database design and administration.

In the next chapter, an overview conclusion will be given in addition to the future work suggestions.

Chapter 8

Conclusions and Future Work

The overall contribution of this thesis is to address several issues related to the use of data access management and try to investigate a new method that adopts of the advantages of the classical methods and avoid their disadvantages. In order to use semantics to intelligently manage data access, the researcher started by looking at the need to protect personal data. Usually enterprises and organisations are entrusted with the information of their clients and are required by law and obligation to keep it hidden from unauthorised people. But this is not an easy task as every day there are reports of cases of privacy mismanagement. Laws and guidelines have been put in place to help clients stipulate how they want their personal information to be used. But even with these laws and guidelines, the personal information abuse is still growing.

A healthcare system is a perfect example, where sensitive information is collected on a daily basis, and where there is also an increasing possibility that data access management would be a problem because healthcare systems continue to grow. This growth is not local, but global as there is a need to share information. Hence the local databases where healthcare records are stored have to be protected against privacy leaks both from within and externally.

From the literature, it is easy to note that in order to manage access to a system, one may specify rules; either on the resource, or on a user or on a role. And with this it is easy to try out a new solution that allows for the specification of a lot of rules.

A semantic data access model was developed as a solution. This semantic data access model is more like a semantic role based access control; semantically specifying access conditions based on user roles in the hospital. For example, if the system identifies a user as a doctor, using semantic rules it decides what part of the information the doctor is allowed to see. This is a safe data access style because its implementation ensures that the user can query only the amount of information allotted to him/her be separating the information from the bulk.

Semantics is a safe way to manage data access because it promises a flexible, interoperable and reusable solution that can easily be improved upon. The only concern that the researcher had at the beginning was that a system using semantics could be limited by speed, because a lot inference is carried out to find out what amount of data a user should view. But later on, it was found out that an optimised way of using semantics will not affect the speed of data retrieving.

In the next section, the researcher presents the contributions of this thesis in more detail, and outlines some directions for future work in Section 8.2.

8.1 Summary of contributions

Research was carried out in order to investigate the technologies and techniques which must be employed to produce a working prototype. This area of research focuses on the application of ontologies, semantics and the surrounding technologies that have been developed to support this. A thorough understanding and grasp of these concepts was crucial to developing the system. The literature review also investigated alternative solutions to the problem in hand. Specifically an overview of database access control systems that are already being used worldwide. By assessing alternative methods that are already in use or are currently under development, the researcher was able to analyse their relevant pros and cons, and this provided a lot of guidance in the subsequent design and implementation of the prototype. Decisions were also made based on the experiences and recommendations revealed in the research material.

To the researcher's knowledge, the classical chain method that has been suggested by (Al-Fedaghi, 2007) has never been implemented nor tested in any hypothetical nor real enterprise. In addition, it has never been designed to solve any particular problem such as the problem of managing access to personal information in healthcare without loss of privacy. The thing which makes the design and implementation of the chain method the first of its kind and the challenges the researcher faced during the design has enriched the literature about the chain method (Omran *et al.,* 2008), (Omran *et al.,* 2009 a,b,c), (Omran *et al.,* 2010 a,b) and (Omran *et al.,* 2012).

The second main contribution of this thesis was the development of a dynamic data access management approach. This new approach decided access attempts not statically, as does the role based access method, but dynamically based on the situation in which a request is made, which is expected to improve on existing approaches in terms of recall and precision

The major contribution of this thesis can be summarised as follows:

1. A methodology has been created for applying privacy policies based on the idea of Chain method integrated with the ontology

2. Privacy assurance has been applied not just inside conventional databases but an approach that can deal with more challenging flexible and dynamic data streams.

3. Personal information ontology has been developed and used it as a classification layer in database access management.

4. The major challenges in the design of chain based solutions have been identified and solved.

5. An architectural design of a Chain data stream that addresses those challenges and used it as a reference framework for research in the area of security and privacy techniques has been developed.

6. This is the first implementation and evaluation of the chain method reported in the literature.

In order to prove hypothesis (H1 and H2) and contribution the researcher had arranged set of comparative type of experiments. In this comparative study, a set of scenarios have been examined, often in the form of a table where a column is reserved for each case. The experiments were divided into two main lines.

The first was at the database level. To show the ease of the design that the Chain/ChBAC method is providing compared with the RBAC. The experiments that have been carried out by the expert database administrators shows that the Chain requires 50% less number of tables, constraints and SQL statements to set up.

We also showed the effect of choosing the ChBAC method on the time required to retrieve data using the two methods. The ChBAC needed almost half the time required by the RBAC method.

The second phase has been undertaken by creating the whole semantic system. The researcher started by testing each part of the system separately and after checking that each part is working as supposed individually she moved to the next step.

In the next step the researcher decided to test the system as a whole to check its performance for four criteria:

1- Accuracy
2- Context sensitivity
3- Time required to retrieve data
4- Flexibility

In order to test these criteria, staff at the hospital were given a number of questionnaires.

The answers that the researcher gets out of these questionnaires gave us a good indication about the reliability of the system in fulfilling the requirements of a real system while maintaining good results for the above four criteria.

In this thesis the researcher has presented a flexible method to improve personal information protection in information system at both the implementation and subsequent access levels. The ontology would add a flexibility layer that is not available in existing methods as it can classify new users and distribute them amongst authorised user groups. She has chosen to create a prototype medical information exchange system to enable industry collaboration and accelerate development of a standards-based national healthcare information system. This approach aims to automatically derive the minimum set of authorisations needed to achieve a service, and as such, the researcher addresses the problem of a privacy preserving data management technique for stream data. Such a problem is already challenging within conventional database systems, but is much more difficult in a data stream context characterised by huge amounts of fast arriving data and by strong performance requirements. This is largely due to the fact that numerous services are being moved online. These services are collecting gigantic amounts of personal information. The need for excessive and increasing collection habits is a cause for concern. This practice needs to be questioned and stopped as it represents a serious threat to personal privacy.

8.2 Future Work

This application has given evidence that semantics can be used as a tool to control access to a database using simple rules. But still there is more work that could be done to make the enhance application functionality and make it more effective.
There are some improvements to the system which the researcher suggests to be taken forwards in the future:

1) Semantic reasoning was investigated, and partially designed but ultimately not fully realised in the prototype (The semantics that has been implemented in this prototype is focusing mainly on user's authorisation). It would be a great achievement to develop a completely general-purpose semantic reasoning engine which could apply the required rules.

This thesis started a method for optimising how the system reasoner works but it could be developed more in the future. In the reasoner that has been envisaged, in which the user does not have strictly defined roles, the researcher stated that in some conditions it may be necessary to iterate through all possible contexts and check all preconditions.

However if one wants to apply all the HIPPA rules for example in this system, the size and complexity of a system grows and more contexts are defined, one may find that it is very processor intensive to do. This could make reasoning very difficult on large databases, where the amount of database requests required to check the preconditions may make semantic reasoning prohibitively slow.

The researcher started by applying semantics only in the authorisation stage. This is where the system checks the users and the chains/acts they can apply on the database and then we work normally with the database. But if the researcher wants to check all the possible tasks that require semantics in the system, it may be beneficial to investigate methods of optimising this. This could be possibly done by devising a way of grouping methods automatically by analysing their preconditions. In this way, a group of contexts could be evaluated together in order to speed things up. Of course, an ontology would provide the ideal structure for defining these groupings. Such an optimisation method and ontology would be one of the basic research goals in the future.

2) Although ODEMapster worked for reading from the relational database, it was not capable of writing data back. This is a possible improvement that would benefit the Kuwait Clinic project enormously as it would provide a uniform method of database access.

3) Instead of bridging the gap between a relational database and an ontology, it would be very useful to devise a way of providing persistent storage in the form of an ontology which conforms to the ACID principles of traditional database systems. This

way, the information could be stored solely inside the ontology, and one could forget about the relational database altogether.

Finally, this solution can be developed and used as a web service. When loaded onto a web server, an authenticated user could login in to the system using an interface designed by an application programmer. Based on his credentials, this present system would retrieve the required semantic information and pass it on to the users system. Then from his system, he can query his piece of information via his interface.

References

Achim D. Brucker and Helmut, P. 2009, “Extending access control models with break-glass”. *In the proceedings of 14th ACM symposium on Access control models and technologies*, New York, USA, pp. 197-206

Agrawal, R., Kiernan, J., Srikant, R. and Xu, Y. 2002, “Hippocratic databases”. *In Proceedings of Very Larga Data Base,* Hong Kong, China, pp. 143-154.

Al-Fedaghi S. 2007, “Beyond purpose-based access control”. *In Proceedings of the 18th Australasian database conference*, Australia, pp. 23-32.

Al-Fedaghi, S. 2006a, “Anatomy of Personal Information Processing: Application to the EU Privacy Directive”, *In the proceedings of the International Conference on Business, Law and Technology (IBLT 2006)*, Copenhagen, Denmark, pp. 129-138

Al-Fedaghi, S. 2006b, “Aspects of Personal Information Theory”, *In the proceedings of the 7th, The Seventh Annual IEEE Information Assurance Workshop (IEEE-IAW)*, West Point, New York, United States Military Academy, pp. 155 - 162.

Al-Fedaghi, S. 2006c, “Personal Information Flow Model for P3P”, *W3C Workshop on Languages for Privacy Policy Negotiation and Semantics-Driven Enforcement*, Ispra, taly.

Al-Fedaghi, S. Fiedler G. and B. Thalheim B. 2005, “Privacy Enhanced Information Systems”. *Proceedings of the 2006 conference on Information Modelling and Knowledge Bases XVI,* pp. 94-111.

Adèr, H. J., Mellenbergh, G. J., & Hand, D. J. 2008, “Advising on research methods: a consultant's companion”, Huizen: Johannes van Kessel Publishing.

Antoniou, G. and Harmelen F. 2003, “A semantic web primer”. London. The MIT press. Retrieved from:
http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=10140
[Accessed 21 October 2011].

Applebaum, P. 2002, “Privacy in Psychiatric Treatment: Threats and Response”, *American Journal of Psychiatry*, Vol. 159, pp 1809-1818.

Ashley, P., Hada, S., Karjoth, G., and Schunter, M. 2002, “E-P3P privacy policies and privacy authorization”, *In the Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society*, New York, USA, pp. 103–109.

Baker, S. 2008, “Access control by action control”. *In the Proceedings of ACM symposium on Access control models and technologies*, New York, USA, pp. 143-152.

Ball, M. 2003, “Hospital information systems: perspectives on problems and prospects, 1979 and 2002”, *International Journal of Medical Informatics,* Vol 69, No. (2-3), pp. 83-89.

Banisar, D. 2000, “Privacy and Human Rights 2000: An International Survey of Privacy Laws and Development”, USA.

Baumer, D., Earp, J. and Payton, F. 2000, “Privacy of medical records: IT implications of HIPAA”, *ACM SIGCAS Computers and Society,* Vol 30, No. 4, pp. 40 – 47.

Becker, T., Streng, C., Luo, Y., Moshnyaga, V., Damaschke, B., Shannon, N. and Samwer, K. 2002, “Intrinsic Inhomogeneities in Manganite Thin Films Investigated with Scanning Tunneling Spectroscopy”, *Phys. Rev. Lett*, Vol. 89, No. 23.

Berners-Lee, T. Hendler, J. and Lassila, O. 2011, "The Semantic Web", *Scientific American*, May 2001, pp. 29-37.

Berg, M. 1999, “Patient care information systems and health care work: a sociotechnical approach”. *International journal of medical informatics.* Vol 55, pp. 87 – 101.

Bertino, E., Bettini, C. and Samarati, P. 1994, “A Temporal Authorization Model”. *In the Proceedings of the 2nd ACM Conference on Computer and Communications Security*, Fairfax, VA, USA, pp. 126–135.

Bertolino A. (2008). Software Testing Research: Achievements, Challenges, Dreams**.** *In the proceedings of the IEEE International conference on software engineering*, Vancouver, Canada, pp. 85 – 103.

Bizer C. 2009, “The D2RQ Platform - Treating Non-RDF Databases as Virtual RDF Graphs”. [Online]. Freie Universität Berlin. Retrieved from:

http://www4.wiwiss.fu-berlin.de/bizer/d2rq/ [Accessed on 11 November 2010].

Buxton, J. N. and Randell, B. (1970). Software Engineering Techniques. Report on a conference sponsored by the NATO Science Committee, Rome, Italy, 27–31 October 1969, pp. 16.

Byun, j.-W. and Li N. 2008, “Purpose based access control for privacy protection in relational database systems”, *VLDB Journal,* Vol. 17, No.4, pp. 603–619.

Carminati, B., Ferrari, E., Heatherly, R., Kantarcioglu, M. and Thuraisingham, B. 2009, “A semantic web based framework for social network access control”. *In the Proceeding the 14th ACM symposium on Access control models and technologies*, New York, NY, USA. pp 177-186.

Carminati B., Ferrari E., and Tan L. 2007, “Enforcing access control over data streams”**,** *In the proceedings of the12th ACM symposium on Access control models and technologies,* Sophia Antipolis, France.

Crampton, Jeremy W. 2003, “ The Political Mapping of Cyberspace”. *Chicago:*

University of Chicago Press.

Crampton, Jeremy W. 2004, “GIS and geographic governance: Reconstructing the

choropleth map”. *Cartographica*, Vol: 39, pp. 41-53.

Checkland, P. and Holwell, S. 1998. “Action Research: Its Nature and Validity”. Systemic Practice and Action Research, Vol. 11, (Issue 1, Feb), pp. 9-21.

The Child Online Privacy Protection Act of 1998, 15 U.S.C. § 6501-6506 (1998). [Online]. Retrieved from:

http://www.ftc.gov/ogc/coppa1.htm [Accessed 11/2/2012].

Chong, F., 2004, “Identity and Access Management”.[Online]. Retrieved from:

http://msdn.microsoft.com/en-us/library/aa480030.aspx [Accessed 1/5/2008].

Computer Misuse Act 1990. (c.18), London: HMSO. [Online]. Retrieved from:

http://www.legislation.gov.uk/ukpga/1990/18/contents [Accessed 11/2/2012].

Covington, M., Panda, S., Liu, L., Strayer, C., Wagner, D., and Kay, S. 2001, “ELF3 modulates resetting of the circadian clock in Arabidopsis”. *Plant Cell* , Vol.13, pp. 1305–1316.

Crook, N., Scheper, T. and Pathirana, V. (2003), “Self organised dynamic recognition statesfor chaotic neural networks, Information Sciences”, Vol: 150(1-2), pp.59–75.

DAML ontology library, 2011. [Online]. Retrieved from:

http://www.daml.org/ontologies/ [Accessed 1/6/2011].

Data Protection Act 1998, (c.29), London: HMSO. [Online]. Retrieved from:

http://www.legislation.gov.uk/ukpga/1998/29/contents [Accessed 11/2/2012].

Denker, G., Kagal, L., Finin, T., Paolucci, M. and Sycara, K. 2003, "Security for DAML Web Services: Annotation and Matchmaking," *in the Proceedings of the. Second International Semantic Web Conference (ISWC2003)*.

Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995, 1995, 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. [Online]. Retrieved from:

http://eurlex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en :HTML [Accessed 1/3/2013].

Dogac, A., Laleci, G., Kirbas S., Kabak Y., Sinir S., Yildiz A. Gurcan, Y. 2006, "Artemis: Deploying Semantically Enriched Web Services in the Healthcare Domain", *Information Systems Journal (Elsevier)*, Vol. 31, No. 4-5, pp.321-339.

Doumen, J.M. and Hulsebosch, B. and Iacob, S.M. and Koster, P. and Kragt, E. and Montaner, J. and Muijen, R. and Petkovic, M. and Vrielink, K. 2005, "Security Aspects of the MESEC Architecture". Technical Report TI/RS/2005/018, Telematica Instituut, Enschede.

Eclipse Foundation. 2011, "WindowBuilder". [Online]. Available

http://www.eclipse.org/windowbuilder [Accessed 3/12/2011].

Elbassuoni, S., Ramanath, M., Schenkel, R. and Weikum, G. 2010, "Searching RDF Graphs with SPAQL and Keywords". Bulletin of IEEE Computer Society Technical Committee on data engineering Bulletin, Vol. 33, No.1.

Data Protection Act 1998, Part IV (Exemptions), Section 36, Office of Public Sector Information, accessed 6 September 2007

Enterprise Privacy Authorization Language (EPAL 1.2), 10 November 2003. [Online]

Retrieved from:

http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/ [Accessed October 15/10/ 2009]

EPIC website. [Online]. Retrieved from:

http://www.epic.com/ [Accessed 10/10/ 2011]

Fair and Accurate Credit Transactions Act of 2003, 15 U.S.C. § 1681, 2003. [Online]. Retrieved from:

http://www.gpo.gov/fdsys/pkg/PLAW-108publ159/html/PLAW-108publ159.htm [Accessed 3/12/ 2011]

Ferraiolo, D. and Kuhn, D. 1992, "Role Based Access Control", *In the proceedings of15th National Computer Security Conference Oct 13-16, 1992*, Baltimore, USA, pp. 554-563.

Freedom of Information Act 2000. (c.36), London: HMSO. [onlone]. Retrieved from:

http://www.legislation.gov.uk/uksi/2001/1637/contents/made [Accessed 3/2/ 2011]

Friedman, R. 2008, "Protecting Customer Privacy". *Information Today*, Vol. 25, No.1. [Online]. Retrieved from:

http://news-business.vlex.com/vid/protecting-customer-privacy-64784117

García-Crespo, A., Gómez Berbís, J., Palacios, R., Alor-Hernández, G 2011, "SecurOntolgy: A Semantic web access control framework", *Computer Standards & Interfaces*, Vol. 33, No. 1, pp. 42 – 49.

Georgiadis C. K., Mavridis I., Pangalos G. and Thomas R.K. 2001, "Flexible team-based access control using contexts", *In the Proceedings of the 6th*

ACM Symposium on Access control models and technologies, Virginia, USA, pp. 21-27.

Guarda, P., Zannone, N. 2009, “Towards the development of privacy-aware systems. Information & Software Technology”, *Information and Software Technology*, Vol. 51, No.2, pp. 337-350.

Gruber, Thomas R. 1993, "A translation approach to portable ontology specifications", *Knowledge Acquisition,* Vol. 5, No. 2, pp. 199–220.

Han, J., and Kamber, M. 2006, “Data Mining – Concepts and Techniques”, 2nd edition, Morgan Kaufman.

Haux R. 2006, “Health information systems – past, present, future”. *International journal of medical informatics*. Vol. 75, pp. 268-281.

Hayes, J.H. and Offutt, A.J. 1999, “Increased software reliability through input validation analysis and testing”. *In Proceedings of the 10th International Symposium on Software Reliability Engineering*, ISSRE'99, Boca Raton, FL, pp 199-209.

He, Q. 2003, "Privacy Enforcement with an Extended Role-Based Access Control Model". NCSU Computer Science Technical Report TR-2003-09.

The Health Insurance Portability and Accountability Act of 1996, 5 U.S.C.A. § 601 1996. [Online]. Retrieved from:

http://www.cms.gov/Regulations-and-Guidance/HIPAA-Administrative-Simplification/HIPAAGenInfo/index.html?redirect=/HIPAAGenInfo/[Accessed 3/2/ 2011].

Health Level Seven Inc.2009, “HL7 Standard”. [Online]. Retrieved from:

http://www.hl7.org/ [Accessed 15/10/ 2009].

Hebeler J., Fisher, M., Blace, R., Perez-Lopez, A. and Mike Dean 2009, “Semantic Web Programming”. Canada: Wiley Publications.

Hitzler, P., Krötzsch, M., Rudolph, S. 2010, “Foundations of semantic web technologies”. Semantic-web.org. [Online]. Retrieved from: http://www.semantic-web-book.org/w/images/7/73/Informatik09-Semantic-Web-1-RDF.pdf [Accessed 12/5/2012].

Hochmüller E. and Mittermeir R. T. 2008, “Agile process myths”. *2008 international workshop on Scrutinizing agile practices or shoot-out at the agile corral. May 10*, 2008. pp. 5-8.

Hodge, G. 2003, “Health Information Privacy and Public Health”, *Journal of Law, Medicine & Ethics*, Vol.31, No.4, pp. 663-671

HPLABS, HPDC. 2009, Jena - Semantic Web Framework [Online]. Retrieved from:

http://www.openjena.org [Accessed 1/12/2011].

Humphreys, B.L. and Lindberg, D.A.B. 1993, “The UMLS project: making the conceptual connection between users and the information they need”. *Bulletin of the Medical Library Association*, Vol. 81, No. 2, pp. 170.

ICO, Security Breaches Reported to ICO, 2010.[Online]. Retrieved from: http://www.ico.gov.uk/upload/documents/library/corporate/research_and_reports/breach_notification_spreadsheet_may2010.pdf.

International Clinic. 2012, IC website [Online]. Retrieved from :

http://www.international-clinic.com. [Accessed on 17 /5/2012].

Jeffries, R., Anderson, A. and Hendrickson, C. 2001, “Extreme programming installed”. Upper Saddle River, New Jersey, Addison-Wesley.

Jones, W. 2008, “How is information personal?” *In the Proceedings of PIM Workshop, SIGCHI.* Florance, Italy.

Kasper, D. 2005, “The Evolution (or Devolution) of Privacy”, *Sociological Forum*, Vol. 20, pp. 69-92.

Karjoth, G., Schunter, M., and Waidner, M. 2003, “Platform for Enterprise Privacy Practices: Privacy-enabled management of customer data,” *In the proceedings of the Privacy Enhancing Technologies: Second International Workshop*, PET 2002, San Francisco, CA, USA, April 14-15, 2002, Revised Papers,Springer.

Ko, J. and Kang, W. 2008. “Enhanced access control with semantic context hierarchy tree for ubiquitous computing”, *International journal of computer science and network security*, Vol 8, No 10, pp.114 – 120.

Koivunen ~~M.~~ R. and ~~E.~~ Miller E. 2001, “W3C Semantic Web Activity”, *In the Proceedings of the Semantic Web Kick-off Seminar in Finland*. [Online]. Retrieved from: http://www.w3.org/2001/12/semweb-fin/w3csw [Accessed 11/11/2010].

Kolter J., Kernchen T., Pernul G. 2010, “Collaborative privacy management. *Computers and security”,* Vol 29, pp. 580 – 591.

Komlenovic M., Tripunitara M., and Zitouni T. 2011, “An empirical assessment of approaches to distributed enforcement in role-based access control (RBAC)”, In the first ACM conference on Data and application security and privacy, pp. 121-132.

Lafky B. H. and Horan A. T. 2008, “Prospective personal health record use among different user groups: results of a multi-wafe study”. *In the Proceedings of the 41st Hawaii International Conference on System Sciences. IEEE*, Hawaii, USA, pp 1530-1605.

Lampson, R. and Butler, W. 1971, “Protection”. *In* the Proceedings of the 5th Princeton Conference on Information Sciences and Systems, pp. 437.

Larsen, E. 2005, “A Unified Approach to Personal Information Management in Interactive Systems”, Technical University of Denmark. 2005, Ph.D. Thesis, CICT Ph.D. Series No. 6.

Lee M., Delaney, C. and Moorhead, S. 2007, “Building a personal health record from a nursing perspective”. *International journal of medical informatics*. Vol. 76, pp. 308 – 316.

Legal directory 2012, “Personal Information Definition”. [Online]. Retrieved from:

http://www.duhaime.org/LegalDictionary/P/PersonalInformation.aspx.

Liu, H., Kuan T, H. B. 2009, “Covering code behavior on input validation in functional testing*”* *Information and Software Technology*, Vol. 51, No 2, pp. 546-553.

Loscocco, P. Smalley, S., Muckelbauer, P., Taylor, R., Turner, S. and Farrell, J. 1998, “The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments”. *In the Proceedings of the 21st National Information Systems Security Conference*, pp. 303–14.

Manchester, University of. 2011, “The OWL API”. [Online]. Retrieved from: http://owlapi.sourceforge.net/ [Accessed 3/12/ 2011].

Massacci, F., Mylopoulos, J. and Zannone, N. (2005), “Minimal Disclosure in Hierarchical Hippocritic Database with Delegation”. *Technical report from the department of information and communication technology*, University of Trento, December 2005.

Mathew S. 2007, “Software engineering”. New Delhi, Chand and Company Ltd., 2007.

Media, J. 2006, “The Laws of Simplicity (Simplicity: Design, Technology, Business, Life)”. Publisher | MIT Press Publish Date | 2006 ISBN | 978-0-262-13472-9.

Mercuri, R. 2004, “The HIPAA-potamus in Health Care Data Security”, *Communications of the ACM*, Vol.47, No.7.

Moffett J. D. 1998, “Control principles and role hierarchies”, *In the Proceedings of the 3rd ACM workshop on Role-based access control*, pp. 63-69.

Mohania M. 2007, “Issues in data privacy management”*, Data & knowledge Engineering*, Vol. 63, pp. 591 – 596.

National Biometric Secrity Project 2006, “United States Federal Laws Regarding Privacy and Personal Data and Applications to Biometrics” , NBSP Publication 0105.

New South Wales Consolidated Acts, 2012. ” [Online]. Retrieved from:

http://www.austlii.edu.au/au/legis/nsw/consol_act/hraipa2002370/ [Accessed on 30 /11/2012].

Ni, Q. Bertino, E. Lobo, J. Brodie, C. Karat, C.-M. Karat, J. and Trombetta, A. 2010, “Privacy-Aware Role-Based Access Control”, ACM Transactions on Information and System Security, Vol. 13, No 24.

NIST, 2012. Online]. Retrieved from:

http://www.nist.gov/index.html [Accessed on 30 /11/2012].

Noffsinger, R. and Chin, S. 2000, “Improving the delivery of care and reducing healthcare costs with the digitization of information”, *Journal of Healthcare Information Management*, Vol 14, No 2, pp. 23—30

Noy, N. and McGuinness, D. 2001, “Ontologies Development 101: A Guide to Creating your First Ontology”, Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and SMI- 2001-0880, March 2001.

OASIS, 2003. “A Brief Introduction to XACML” [Online]. Retrieved from:

https://www.oasisopen.org/committees/download.php/2713/Brief_Introduction_to_XACML.html#xacml-example [Accessed on 30 /11/2010].

OASIS. 2005, “eXtensible Access Control Markup Language (XACML) Version 2.0. OASIS Standard, 2005”. [Online]. Retrieved from:

http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf [Accessed on 30 /11/2010].

Obitko, M. 2007, “Translations between Ontologies in Multi-Agent Systems”, Ph.D. dissertation, Faculty of Electrical Engineering, Czech Technical University in Prague.

Osborn, T. J. & Hulme, M. 2000, “Observed trends in the intensity of daily precipitation over the UK”, *In the proceeding of the 12th Conference of Applied Climatology*, Asheville, Boston, pp. 111–114.

Omelayenko, B., Fensel, D. and Bussler, C. 2002, “Mapping Technology for Enterprise Integration”, *In the proceedings of the 15th International FLAIRS conference*, AAAI Press.

Omran, E., Bokma, A. and Abu Al-maati, S. 2008, "Chain ontology based: A model for protecting personal information privacy". *In the proceedings of the ICDIM conference,* pp.363-368.

Omran, E., Bokma, A. and Abu Al-maati, S. 2009a, "A K-anonymity Based Semantic Model For Protecting Personal Information and Privacy", *In the* *proceedings of Advance Computing Conference. IACC 2009. IEEE International*, pp. 1443-1447.

Omran, E., Bokma, A. and Abu Al-maati, S. 2009b "Hippocratic ontology based: A model for protecting personal information privacy", 11th International Conference on Enterprise Information Systems (ICEIS), *In the Proceedings of the 11th International Conference on Enterprise Information Systems*, Vol. 3- ISBN 978-989-8111-86-9 , pp. 376-382.

Omran, E., Bokma, A. and Abu Al-maati, S. and Nelson, D. 2009c "Implementation of a Chain Ontology Based Approach in the Health Care Sector", *Journal of Digital Information Management* ,Vol. 7, No. 5 , pp. 270-275.

Omran E., Grandison T., Kumar P. 2010a, "Optimizing the Design and Implementation of Privacy Controls for Healthcare Database Systems". *AMIA Now! 2010. May 25-27, 2010. Phoenix, Arizona.*

Omran E., Tyrone G., Abu Almaat S. 2010b, "Healthcare Chains - Enabling Application and Data Privacy Controls for Healthcare Information System*". In the proceedings of the 13th World Congress on Medical and Health Informatics (MEDINFO) 2009. September 12-15, 2010. Cape Town, South Africa.*

Omran, E., Tyrone G., Bokma, A. and Abu Al-maati, 2012, "Evaluating Chain-Based Access Control in Healthcare Information Systems". Submitted to IEEE Transactions on Dependable and Secure Computing (TDSC).

Ontolingua. (2011), “ontology library”. [Online]. Retrieved from: http://www.ksl.stanford.edu/software/ontolingua/ [Accessed 1/6/2011].

Paolucci, M., Shehory, O. and Sycara, K. 2001, “ Interleaving Planning and Execution in a Multiagent Team Planning Environment”. *Electronic Transactions of Artificial Intelligence*. [Online]. Retrieved from:

http://www.daml.org/services/ETAI2001-CMU.pdf. [Accessed 1/6/2011]

Pefiers, K., Tuunanen, T., Rothenberger, M. 2008, “ A design science research methodology for information systems research”, *Journal of Management Information Systems archive,* Vol. 24, No. 3, pp. 45-77.

Protégé 2010, “Protégé Documentation”. [Online]. Retrieved from: http://protege.stanford.edu/. [Accessed on 23/11/ 2010].

Philip M. 2010, “Introduction to Jena”. [Online]. IBM, DeveloperWorks. Retrieved from: https://www.ibm.com/developerworks/java/library/j-jena/ [Accessed on 11/11/ 2010].

Price, C. and Spackman, K. 2000, “SNOMED clinical terms”. *BJHC&IM-British Journal of Healthcare Computing & Information Management*, Vol. 17, No. 3, pp. 27-31.

Privacy NSW Privacy Management Plan. 2006, “Privacy and Personal Information Protection Act and the Health Records and information Privacy Act 2”. [Online]. Retrieved from: http://www.privacy.nsw.gov.au/lawlink/privacynsw/ll_pnsw.nsf/pages/privacy_ourmanagementplan. [Accessed 1/5/ 2012].

Qin, L. and Atluri, V. 2003, “Concept-level access control for the semantic web”. *In Workshop on XML Security, held in conjunction with the 10th ACM Conference on CCS.*

Quilitz, B., Leser, U. 2008, “Querying Distributed RDF Data Sources with SPARQL”, *In the proceedings of Semantic Web: Research and Applications,* Springer, Berlin / Heidelberg, Germany, Vol. 5021, pp. 524-538.

Sandhu, R. 1998, “Role-based Access Control”. *Advances in Computers*, Vol. 46, pp. 237-286. Editor: Marvin Zelkowitz. ISBN-13: 978-0120121465.

Sawma, V. 2002, “E-commerce Security Anew Methodology for Deriving effective Countermeasures Design Model”. PhD. Ontario, Canada: University of Ottawa.

SCFBIR. 2010, “Protégé - Open source ontology editor and knowledge-base framework”. [Online]. Stanford Center for Biomedical Informatics Research. Retrieved from:

http://www.protege.stanford.edu/ [Accessed 1/12/ 2011].

Schach R. 2005, “Object Oriented & classical software engineering”, Sixth Edition. New York, McGrawHill

Shields B. 2006, “Using semantic rules to determine access control for web services” *In the Proceedings of the 15th international conference on World Wide Web*, Vol. 332 No 9, pp. 913-914.

Shore J. and Warden S. 2008, “The art of agile development”. California, O’Reilly, 2008.

Stallings, W. 2011, ”Network Security Essentials: Applications and Standards”, Fourth Edition, W. Stallings, Prentice Hall, 2011.

Szekely, B. and Betz, J. 2011, “Jastor: Typesafe, Ontology Driven RDF Access from Java” [Online]. Available:

http://jastor.sourceforge.net/ [Accessed 3/12/2011]

Tektonidis D., Bokma A., Oatley G., Salampasis M. 2005, “ONAR: An Ontologies-Based Service Oriented Application Integration Framework*”. In proceedings of the First International Conference on Interoperability of Enterprise Software and Applications”, February 23-25, Lecture Notes in Computer Science (Interoperability of Enterprise Software and Applications), ISBN: 1-84628-151-2, Geneva, Switzerland, 2005.*

Thomas R. and Gruber 1993, “A Translation Approach to Portable Ontology Specifications”. *Knowledge Acquisition*, Vol. 5, No. 2, pp. 199-220.

Thomas, R and Sandhu, R. 1997, " Task-based Authorization Controls(TBAC): A Family of Models for Active and Enterprise-oriented Authorization Management", *In the Proceedings of the IFIP WG11.3 Workshop on Database Security, Lake Tahoe, California, August 11-13, 1997.*

Thomson R. 2009, "100 Top Hospitals: 2009". [Online]. Retrieved from: http://www.modernhealthcare.com/section/lists?djoPage=product_details&djoPid=10537&djoTry=1249923457 [Accessed 15/10/ 2009].

Teufel. 2006, “ChapterAnOverviewofevaluationmethods inTRECAd-hocInformationRetrievalandTRECQuestionAnswering”.In:L.Dybkjaer,H.Hemsen,W.Minker(Eds.)EvaluationofTextandSpeechSystems.Springer,Dordrecht,The Netherlands.

University of Alberta, Health Law Institute, University of Victoria, School of Health Information Science. 2005, “Electronic Health Records and the Personal Information Protection and Electronic Documents Act”, *Report prepared with generous funding support from the Office of the Privacy Commissioner of Canada.*

United States Department of Defense. 1985, “Trusted Computer System Evaluation Criteria”. December 1985. DoD Standard 5200.28-STD.

Uschold, M. and Grüninger, M. 1996, “Ontologies: Principles, Methods, and Applications”, *Knowledge Engineering Review*, Vol. 1, pp.96-137.

UPM. 2010, ODEMapster Plugin [Online]. Available:

http://neon-toolkit.org/wiki/ODEMapster [Accessed 1/12/2011].

"US State Privacy Laws." Protegrity. N.p. 2008. Web. [Online]. Retrieved from:

http://www.protegrity.com/solutions/compliance-solutions/. [Accessed 25/10/2010].

W3C Community, 2011. Notation3 (N3): A readable RDF syntax. [Online] Retrieved from: http://www.w3.org/TeamSubmission/n3// [Accessed 21/10/2011].

W3C Community, 2011. N-Triples - W3C RDF Core WG Internal Working Draft. [Online]Retrieved from: http://www.w3.org/2001/sw/RDFCore/ntriples [Accessed 21/10/2011].

W3C Community, 2004. OWL Web Ontology Language Reference. [Online] Retrieved from: http://www.w3.org/TR/owl-ref/ [Accessed 2/10/2011].

W3C Community, 2004. OWL Web Ontology Language Reference. [Online] Retrieved from: http://www.w3.org/P3P/ [Accessed 21/11/2012].

Warren and Brandeis. 1980, “The Right to Privacy”, 4 Harvard Law Review 193 (1890). [Online]. Retrieved from:

http://www.law.louisville.edu/library/collections/brandeis/node/225. [Accessed 21/5/2012].

Wilikens, M., Feriti, S., SAnna, A. and Masera, M. 2002, “A context-related authorization and access control method based on RBAC: A case study from the health care domain”. *In Proceedings of eventh ACM symposium on Access control models and technologies,* California, USA, pp. 117-124.

Yague M., Maña, A., López, J. and Troya, J. 2003, “Applying the semantic web layers to access control”, *In the proceedings of the 14th International workshop in database and expert systems applications*, Washington, DC, USA, pp. 622.

Achievements

1- Publications:

- Esraa Omran, Albert Bokma and Shereef Abu Al-Maati, "Chain ontology based: A model for protecting personal information privacy". *ICDIM,* page(s): *363-368. IEEE,* (*2008*). This paper has been referenced in: Belanger F. and Crossler R.E. 2011. ‘Privacy in the Digital Age: A Review of Information Privacy Research in Information Systems.’ MIS Quarterly, 35(4): 1017-1041.

- Esraa Omran, Albert Bokma and Shereef Abu Al-maati. "A K-anonymity Based Semantic Model for Protecting Personal Information and Privacy", Advance Computing Conference. IACC 2009. IEEE International, Page(s): 1443-1447, (2009). This paper has been referenced in “Biometrics, Device Metrics and Pseudo Metrics in a Multifactor Authentication with Artificial Intelligence”*, Broadband and Biomedical Communications (IB2Com), 2011 6th International Conference on,* Page(s): 157- 162

- Esraa Omran, Albert Bokma and Shereef Abu Al-Maati. "Hippocratic ontology based: A model for protecting personal information privacy", 11th International Conference on Enterprise Information Systems (ICEIS), Proceedings of the 11th International Conference on Enterprise Information Systems, Volume 3- ISAS, Milan, Italy, May 6-10, 2009. 2009, ISBN 978-989-8111-86-9, page(s): 376-382, (2009).

- Esraa Omran, Albert Bokma, Shereef Abu Al-Maati, David Nelson. "Implementation of a Chain Ontology Based Approach in the Health Care Sector", Journal of Digital Information Management ,Volume 7 Number 5 , October 2009, pages: 270-275, (2009).

- Esraa Omran, Tyrone W Grandison , Shereef Abu Al-maati , Albert Bokma and David Nelson. "Healthcare Chains – Enabling Application and Data Privacy Controls for Healthcare Information Systems", (Has been accepted in MEDINFO 2010 conference).

- Esraa Omran, Albert Bokma, Tyrone Gradison, Pratheep Kumar. "Optimizing the Design and Implementation of Privacy Controls for Healthcare Database Systems". (has been accepted by AMIA Now 2010 journal- Manuscript ID: AMIA-041-N2010, 28-Jan-2010).

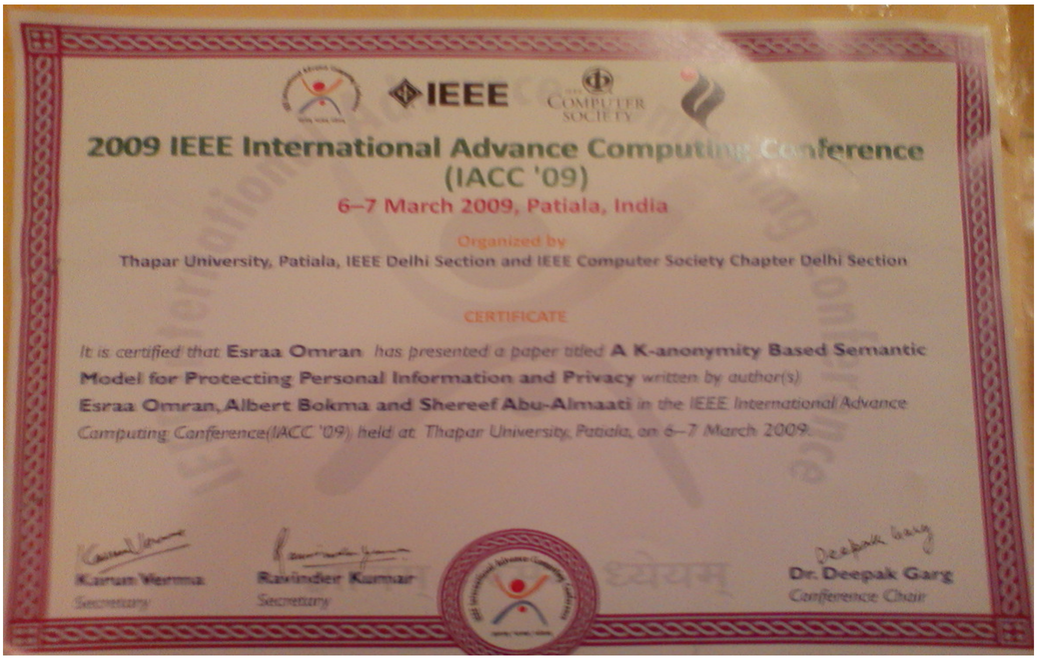
- Esraa Omran, Albert Bokma, Nicola Zannone, David Nelson, Shereef Abu Almaati. "Comparing Data Privacy Controls in Healthcare Information Systems. (has been accepted for publication as a workshop paper at the FARES 2011 workshop) .

- Esraa Omran, Tyrone Grandison, Albert Bokma, Shareef Al-Maati “Evaluating Chain-Based Access Control in Healthcare Information Systems”. Submitted to IEEE Transactions on Dependable and Secure Computing (TDSC).

- Albert Bokma, Sheila Garfield, David Nelson, Esraa Omran, Oscar Corcho, Cerif4Datasets (C4D) - Utilising Semantics for the Discovery and Exploration of Datasets in Research. CRIS 2012: e-Infrastructures for Research and Innovation - Linking Information Systems to Improve Scientific Knowledge Production, 11th International Conference on Current Research Information Systems, June 6-9, Prague, Czech Republic

2- Invitation for Esraa Omran from Trento University in Italy to work on real European project there for two months under the supervision of Prof Fabio Masacci.

3- A certificate has been given to me-Esraa Omran for participating in the 125th anniversary of IEEE because of participation with papers.



2009 IEEE International Advance Computing Conference
(IACC '09)
6–7 March 2009, Patiala, India

Organized by
Thapar University, Patiala, IEEE Delhi Section and IEEE Computer Society Chapter Delhi Section

CERTIFICATE

It is certified that **Esraa Omran** *has presented a paper titled* **A K-anonymity Based Semantic Model for Protecting Personal Information and Privacy** *written by author(s)* **Esraa Omran, Albert Bokma and Shereef Abu-Almaati** *in the IEEE International Advance Computing Conference(IACC '09) held at Thapar University, Patiala, on 6–7 March 2009.*

Karun Verma
Secretary

Ravinder Kumar
Secretary

Deepak Garg
Dr. Deepak Garg
Conference Chair

4- Esraa Omran has been chosen as a reviewer in the Fourth International Conference on Digital Information Management

Appendix

This Appendix represents some basic java codes required for system implementation and then shows some early system testing.

The following codes snippet show the basic steps required to develop the following basic parts:

1- To connect to Jena program

“

A JENA model is created using JENA’s model factory.

```
Model = ModelFactory.createOntologyModel() ;
```

Once the model is created, the RDF and OWL files are loaded into it;

```
InputStreammodelFile = FileManager.get().open(rdfFile);
model.read(modelFile, defaultNS);
```

And

```
InputStreaminFoafInstance = FileManager.get().open(ontologyFile);
model.read(inFoafInstance,defaultNS);
```

“

2- To connect to OdeMapster

In the earlier version of the system prototype the RDF-file has been produced manually, semantic statements like:

“

Doctor (OWL class) is an equivalentClass to Clinicianstable (GP)

Nurse (OWL class) is an equivalentClass to Clnincianstable(nurse)
Patient (OWL class) is an equiventClass to patientstable(patient)

The whole process involves creating a Resource to represent the OWL class (subject), a property to represent the predicate and another resource to represent the RDF object. An example is given in the Appendix.

So that JENA would find relationship between the two semantic data's (RDF and OWL).

After this, the *schema* is passed into a JENA's OWLReasoner to create a new model that sees the RDF and OWL files as one. For this thesis, the JENA was the approved SWRL, but this posed no restriction to what reasoner that can be used for this thesis, as any reasoner that supports SWRL rules can be used to perform semantic inferencing. For example SweetRules and Pellet reasoners can be used for inference. (Carminati et. al., 2009) But a decision made to stick to the basic JENA reasoners for simplicity.

```
Reasonerreasoner = ReasonerRegistry.getOWLReasoner();
reasoner = reasoner.bindSchema(schema);
inferredModel = ModelFactory.createInfModel(reasoner, model);
```

ODEMapster has been utilised then for producing the RDF file and then to collaborate it with JENA for the mapping between the database and the ontology. To import the software, the researcher retrieved the source code directly from the subversion development server, using the following instruction directly on the command prompt:

```
esraa@esraa-desktop:~$ svn checkout http://oeg-obdi.googlecode.com/svn/trunk/ ODEM2
```

This “checked-out” the latest version of the source code into a directory on the local machine so that the directory would be imported into Eclipse.

Configuration of ODEMapster engages two files. The first file describes the database mappings. In this prototype, this file is called “kuwaitclinic.ttl”. The database mapping file is consisted of RDF-Triples which map the tables of the relational database to an ontology structure. Once generated, this ontology can then be imported into Jena.

Example of this is the “clinicusers” table is described as holding a list of “Person” objects; this creates a list of individuals within the ontology named according to their UserID and defines the users’ names as being properties of these objects.

The second file for this prototype is called “kuwaitclinic.r2rml.properties”. This file contains the name of the mapping file, details of the database to be accessed, its location, username and password.

3- To develop the semantic interface part:

```
“
// Get the reasoner manager and obtain a reasoner for the OWL model.
ReasonerManager reasonerManager = ReasonerManager.getInstance();
Protégé OWLReasoner reasoner = reasonerManager.getReasoner(model);
“
```

Communication with an external DIG compliant reasoner is done over HTTP, so the URL of the external DIG Reasoner needs to be provided. This is done with the setReaonerURL (String url) method on Protégé OWLReasoner. Having set the URL, the connection may be tested using the isConnected () method. An example is shown in the following code snippet:

```
“
```

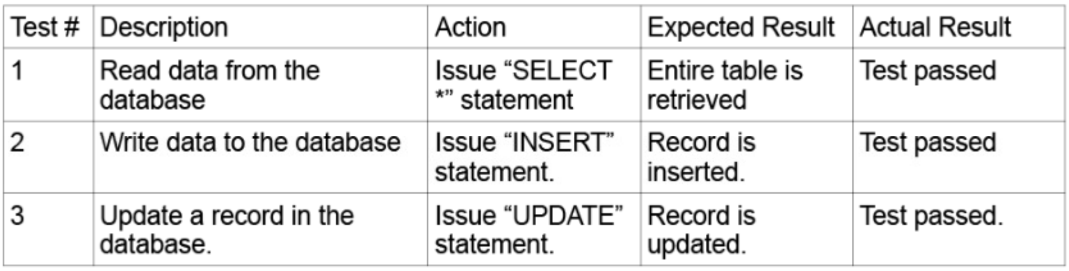
```
// Set the reasoner URL (using the default URL for Racer here) and test the
connection.
reasoner.setURL(http://localhost:8080);
if (reasoner.isConnected()) {
   // Get the reasoner identity - this contains information
   // about the reasoner, such as it's name and version,
   // and the tell and ask operations that it supports.
   DIGReasonerIdentity reasonerIdentity = reasoner.getIdentity();
   System.out.println("Connected to " + reasonerIdentity.getName());
}
```

“

a – Database read/write testing

As the system is using connection to the database, one of the most basic tests was to ensure that the system is able to read and write data to the MySQL database successfully. It is important to note that these tests do not need to test the database itself, only that the prototype can interact with the database correctly.

These tests were carried out from within the Java system, as this would provide proof that the prototype has been successfully connected to the database.



Test #	Description	Action	Expected Result	Actual Result
1	Read data from the database	Issue “SELECT *” statement	Entire table is retrieved	Test passed
2	Write data to the database	Issue “INSERT” statement.	Record is inserted.	Test passed
3	Update a record in the database.	Issue “UPDATE” statement.	Record is updated.	Test passed.

These three tests cover the three actions that software will perform on the database (read, write, and update) on so this was sufficient to test the database connection.

b- JENA Model Test

The purpose of this test is to ensure that a model was created and all the required files were loaded into it. These files includes

- The RDF database file
- OWL ontology file
- FOAF ontology file

Test Method: Print Model to screen by using a created print method

“

```
//printModel, prints JENA models to standard output
        public void printModel(){
                model.write(System.out);
        }
```

Execution:

```
public SemanticModel() throws IOException{
                createModel();
                //test
                //printModel();

                loadRDFModel() ;
                //test
                //printModel() ;

                loadOntology() ;
                //test
                //printModel();

                loadFOAFModel();
}
“
```

Result: The result below is printed to output showing RDF header after createModel() was called.

```
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" >
</rdf:RDF>
```

Test conclusion: This indicated that a model has been successfully created.

c- Bridge Test

After integrating the RDF and OWL files into a model, a test is required to ensure that the reasoner merged the namespaces.

Method: Run a query to select all clinicians. If this is successful, if would also return the Individuals created in the Medical ontology

Execution: Run the query below before creating bridge

```
str = ("select?name {" +
        "?doc vocab:clinicianstable_ClincianName ?name}");
```

Result:

```
---------------------
| name              |
=====================
| "Ms Molly Matron" |
| "Dr Whitbread"    |
| "Dr Dixon"       |
```

Run Query again after bridge creation.

Result: Error,

Error Analysis: The result is the same as the first one. Try again but this time use the inferfferedModel and not the model.

Result: Success!

```
---------------------
| name             |
=====================
| "Ms Molly Matron" |
| "Dr Whitbread"    |
| "Dr Dixon"        |
| "Ontology Doc1"   |
| "Ontology Nurse1" |
---------------------
```

It is easy to see that “Ontology Doc” and “Ontology Nurse” are individuals created with the ontology and not from the database

d- Ontology Mapping testing

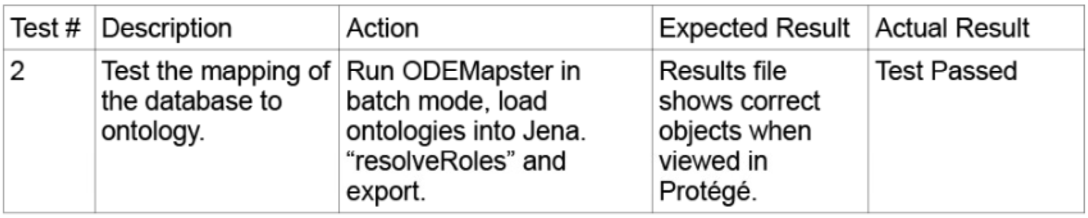
Testing the ontology mappings was a little more involved. In order to do this, it was necessary to run ODEMapster in batch mode. It was then possible to load the results file into the Protégé ontology editor in order to view the results. Test # Description Action Expected Result Actual Result 1 Test the mapping of the database to ontology. Modify mappings file, run ODEMapster in batch mode. Results file shows correct objects when viewed in Protégé. Test Failed Initially this test failed, because of the failure of the mapping file to differentiate between

Doctors, Nurses, Admin and Patients are objects in the clinicusers table. In order to correct this, Jena has been used to import the domain ontology and then the results file. I then called the “resolveRoles” method, as described in the implementation chapter. Once this was done, it was necessary to write the results back to a file to be loaded into Protégé. This was done with the following code:

“

```
public void printModel(Model model){
model.write(System.out);
}
```

“

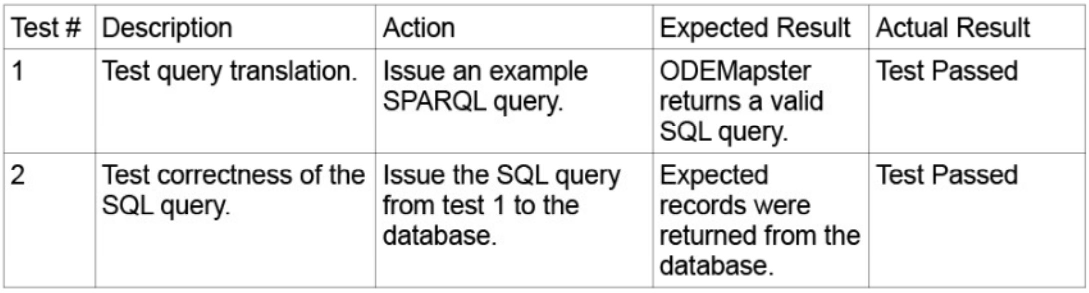


Test #	Description	Action	Expected Result	Actual Result
2	Test the mapping of the database to ontology.	Run ODEMapster in batch mode, load ontologies into Jena. “resolveRoles” and export.	Results file shows correct objects when viewed in Protégé.	Test Passed

It should be noted that these test were run multiple times, every time the mappings file was modified. By doing this, the mappings file was constructed and tested step by step until complete.

e – SPARQL Query Test

The SPARQL query test was the last to be completed, as it relied on a complete and correct mapping file and an assurance that both ODEMapster and the database connection were working correctly. As mentioned in the implementation, ODEMapster was used to translate the SPARQL queries into SQL statements, and these SQL statements were then issued to the database. This provides two junction points at which the researcher can test. Firstly, the query translation was tested (test 1). Then the resulting SQL statement was issued, and results inspected (test 2). By linking the two tests together, the researcher can check that the SPARQL query issued ultimately results in the expected set of data from the database.



Test #	Description	Action	Expected Result	Actual Result
1	Test query translation.	Issue an example SPARQL query.	ODEMapster returns a valid SQL query.	Test Passed
2	Test correctness of the SQL query.	Issue the SQL query from test 1 to the database.	Expected records were returned from the database.	Test Passed

For an example of this, here are the results of a test ran on the system:

SPARQL Query Issued:

```
PREFIX qqq: <http://www.q8onto.org/healthcareOntology.owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?patient ?name ?dob
WHERE {
?patient a <http://www.q8onto.org/healthcareOntology.owl#Patient> .

?patient foaf:name ?name .
?patient qqq:dob ?dob .
?patient qqq:hasPrimaryDoctor ?primarydoctor .
FILTER ( ?primarydoctor = "3" )
}
```

Translated into SQL:

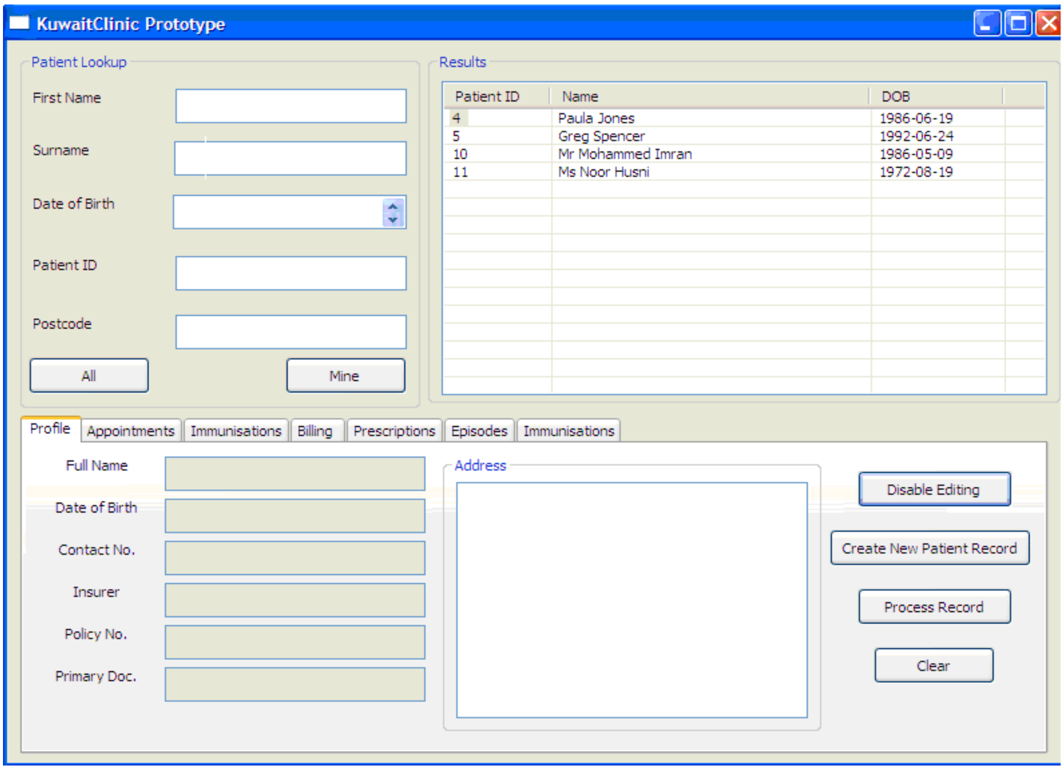
```
SELECT var_patient AS patient, var_dob AS dob, var_name AS name
FROM (SELECT *
FROM (SELECT var_primarydoctor AS var_primarydoctor, var_name AS var_name,
v_9257.var_patient AS var_patient, uri_dob1075654325 AS
uri_dob1075654325, var_dob AS
var_dob, uri_name1396749066 AS uri_name1396749066,
uri_hasPrimaryDoctor1710373065 AS
uri_hasPrimaryDoctor1710373065
FROM (SELECT v_8217.PatientID AS var_patient,
'http://xmlns.com/foaf/0.1/name' AS
uri_name1396749066, v_8217.FullName AS var_name
FROM patientstable v_8217
WHERE (v_8217.FullName IS NOT NULL) ) v_9257
INNER JOIN (SELECT v_2231.var_patient AS var_patient,
uri_hasPrimaryDoctor1710373065 AS
uri_hasPrimaryDoctor1710373065, var_dob AS var_dob, uri_dob1075654325
AS uri_dob1075654325,
var_primarydoctor AS var_primarydoctor
FROM (SELECT v_4940.PatientID AS var_patient,
'http://www.q8onto.org/healthcareOntology.owl#dob' AS uri_dob1075654325,
v_4940.DOB AS var_dob
FROM patientstable v_4940
WHERE (v_4940.DOB IS NOT NULL) ) v_2231
INNER JOIN (SELECT v_8725.PatientID AS var_patient,
'http://www.q8onto.org/healthcareOntology.owl#hasPrimaryDoctor' AS
uri_hasPrimaryDoctor1710373065, v_8725.PrimaryDoctor AS
var_primarydoctor
FROM patientstable v_8725
WHERE (v_8725.PrimaryDoctor IS NOT NULL) ) v_2910 ON
((v_2231.var_patient =
v_2910.var_patient) OR (v_2231.var_patient IS NULL) OR
(v_2910.var_patient IS NULL)) ) v_1346
ON ((v_9257.var_patient = v_1346.var_patient) OR (v_9257.var_patient IS
NULL) OR
(v_1346.var_patient IS NULL)) ) v_8149
```

WHERE (var_primarydoctor = '3')) v_9720

<u>Returned results:</u>

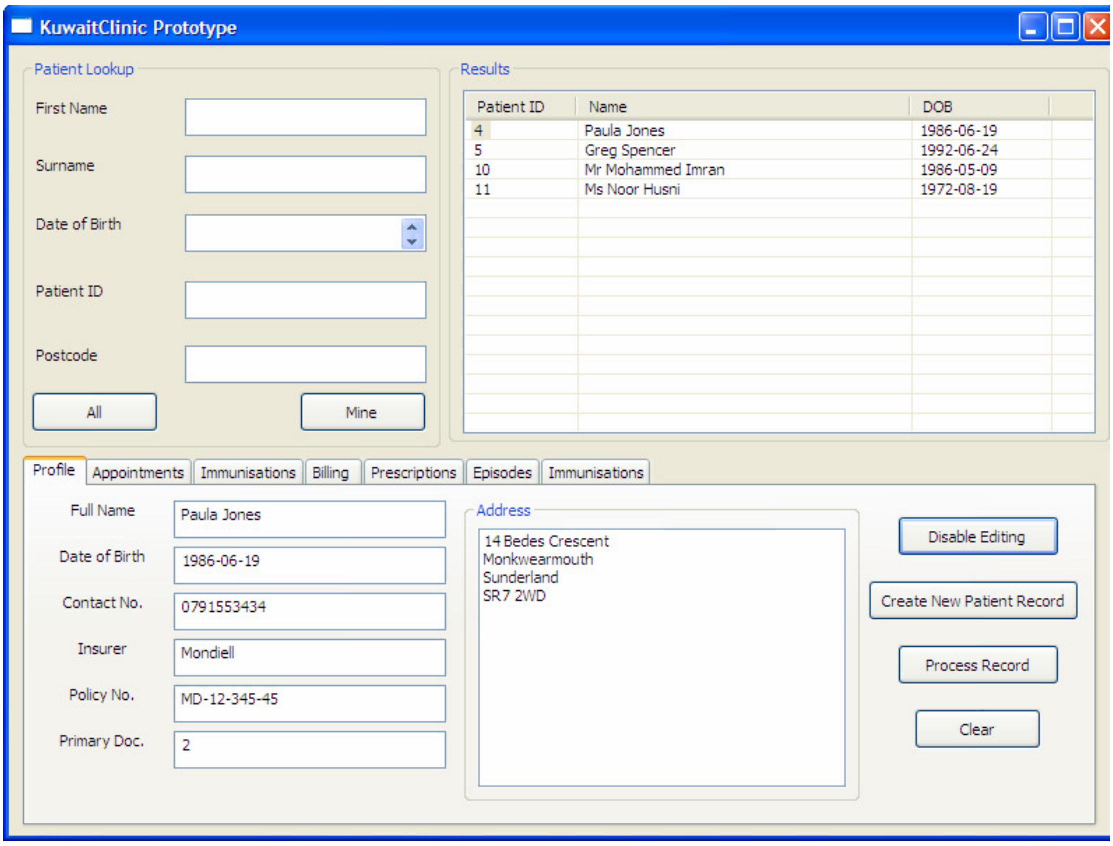
Two records, correctly identifying the patients for whom the primary doctor is DoctorID #3

Screenshots for the GUI for different users:



Patient ID	Name	DOB
4	Paula Jones	1986-06-19
5	Greg Spencer	1992-06-24
10	Mr Mohammed Imran	1986-05-09
11	Ms Noor Husni	1972-08-19

Figure 60: The GUI for Admin after a successful access-check section 7.7-part f.



Patient ID	Name	DOB
4	Paula Jones	1986-06-19
5	Greg Spencer	1992-06-24
10	Mr Mohammed Imran	1986-05-09
11	Ms Noor Husni	1972-08-19

Full Name: Paula Jones

Date of Birth: 1986-06-19

Contact No.: 0791553434

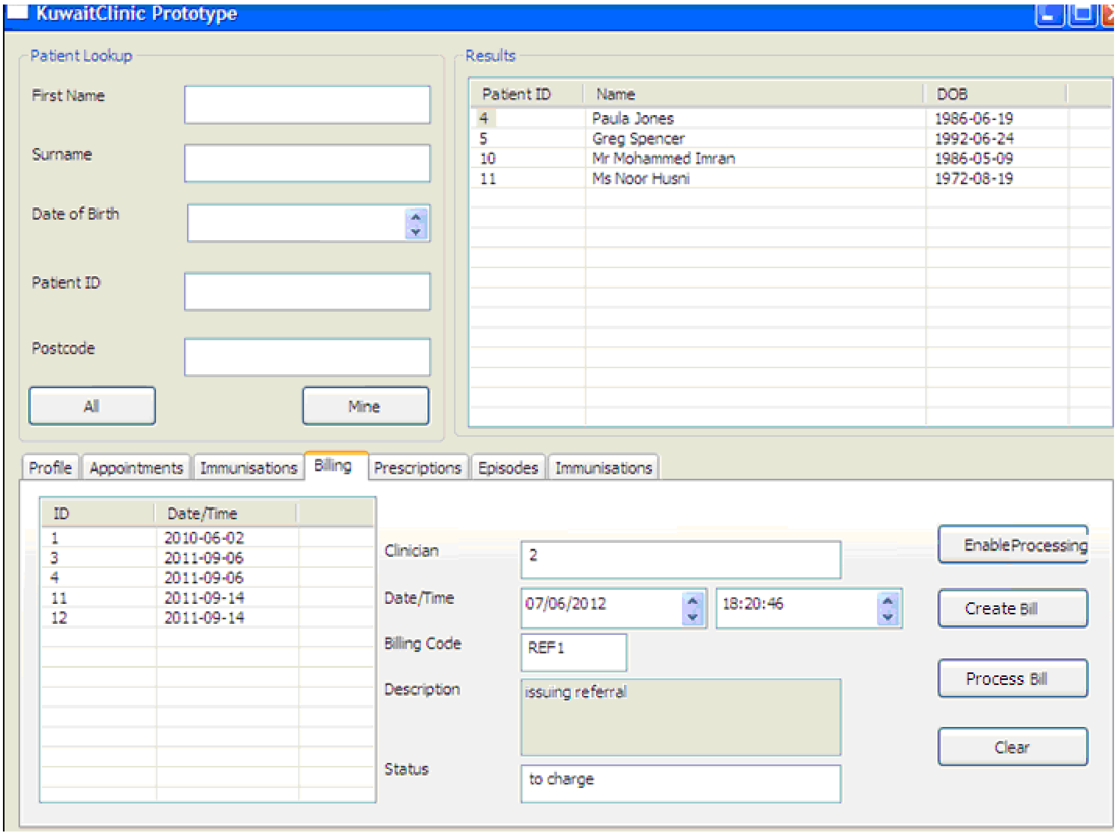
Insurer: Mondiell

Policy No.: MD-12-345-45

Primary Doc.: 2

Address:
14 Bedes Crescent
Monkwearmouth
Sunderland
SR7 2WD

Figure 61: Editing the information of existing patient.



Patient ID	Name	DOB
4	Paula Jones	1986-06-19
5	Greg Spencer	1992-06-24
10	Mr Mohammed Imran	1986-05-09
11	Ms Noor Husni	1972-08-19

ID	Date/Time
1	2010-06-02
3	2011-09-06
4	2011-09-06
11	2011-09-14
12	2011-09-14

Clinician: 2

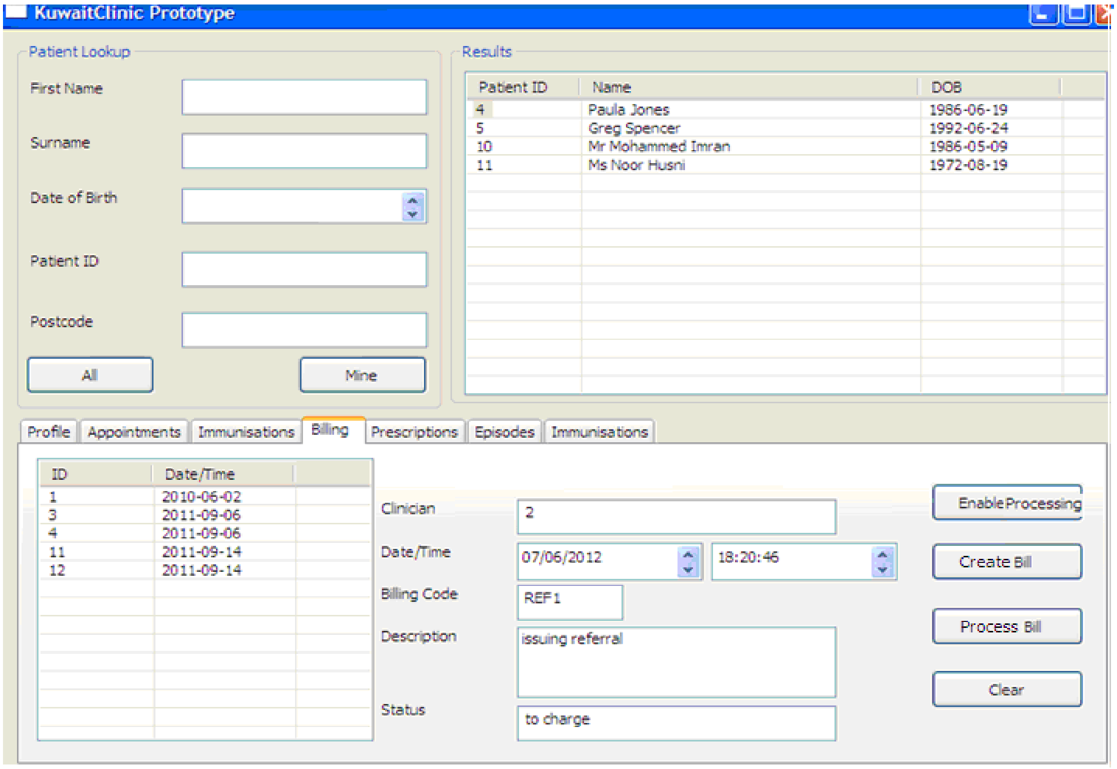
Date/Time: 07/06/2012 18:20:46

Billing Code: REF1

Description: issuing referral

Status: to charge

Figure 62: Checking the billing of a specific patient.



Patient ID	Name	DOB
4	Paula Jones	1986-06-19
5	Greg Spencer	1992-06-24
10	Mr Mohammed Imran	1986-05-09
11	Ms Noor Husni	1972-08-19

ID	Date/Time
1	2010-06-02
3	2011-09-06
4	2011-09-06
11	2011-09-14
12	2011-09-14

Clinician: 2

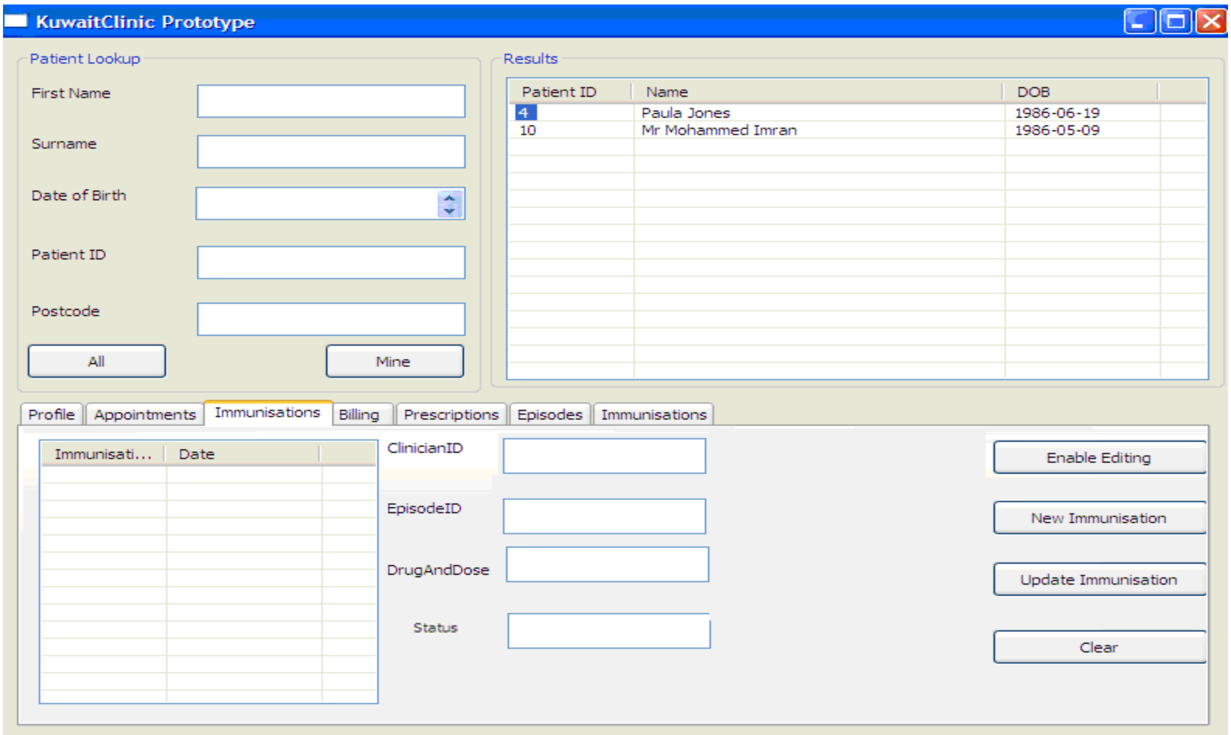
Date/Time: 07/06/2012 18:20:46

Billing Code: REF1

Description: issuing referral

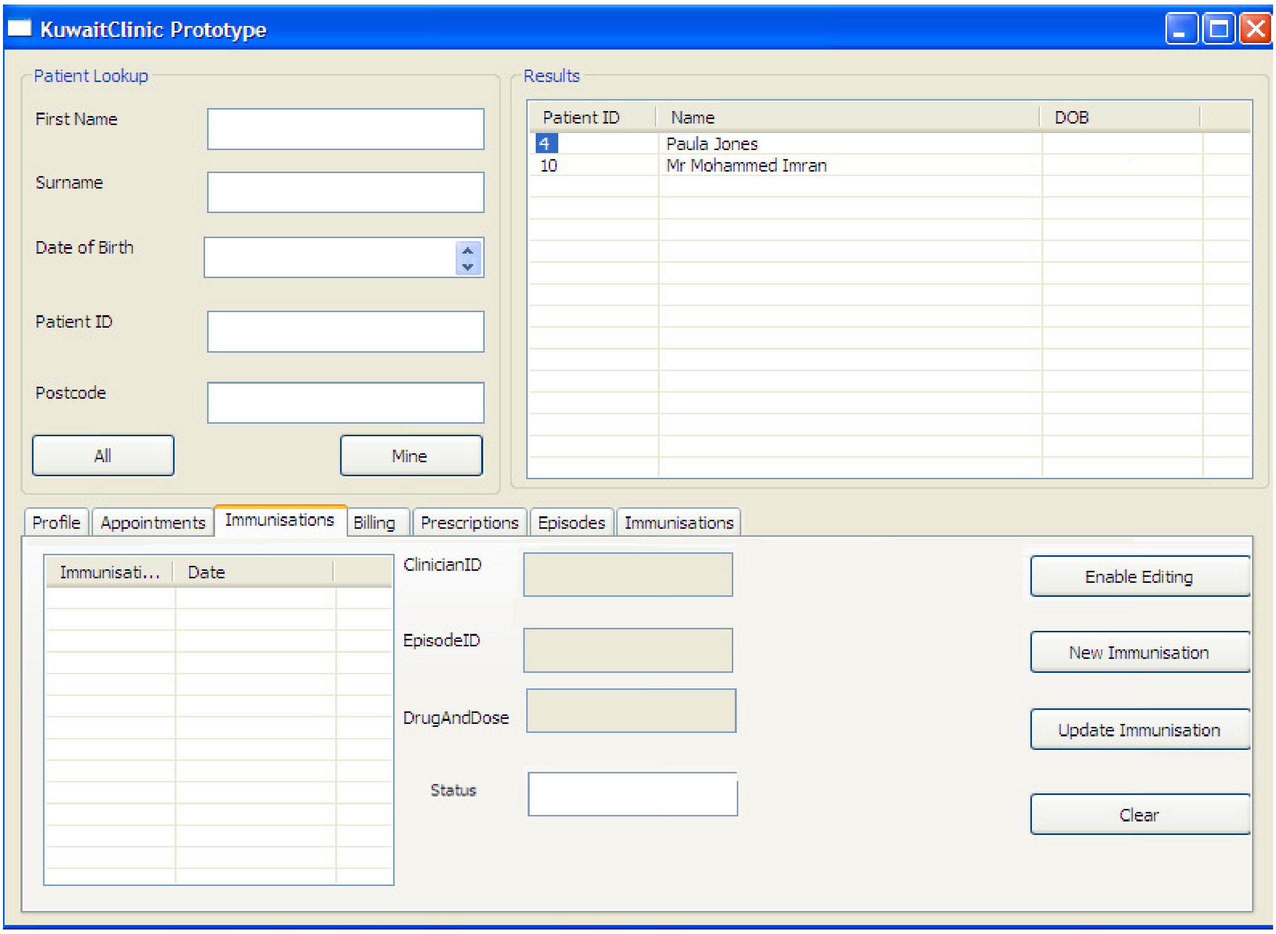
Status: to charge

Figure 63: GUI for admin for system A and C.



Patient ID	Name	DOB
4	Paula Jones	1986-06-19
10	Mr Mohammed Imran	1986-05-09

Figure 64: Immunisation tab as shown for a user type Doctor for system B and it's the same for systems A and C also .



Patient ID	Name	DOB
4	Paula Jones	
10	Mr Mohammed Imran	

Figure 65: Immunisation tab as shown for a user type Nurse for system B.