Role models in healthcare

Specialization assignment
TDT4700 Healthcare informatics
Autumn 2004

Bjørn-Erik Stenbakk and Gunnar René Øie

NTNU
Department of computer and information science

Supervisor: Øystein Nytrø

Abstract

This report examines the possibility of developing a role-model that is capable of dealing with both access control and information ranking.

The motivation for this project is trying to improve efficiency in healthcare by using IT-based solutions. The healthcare sector is very information intensive. Also, healthcare information is required by law to be strictly secured. Thus an access control policy is needed. Using roles, instead of just users or clearance levels, is necessary to enforce this policy. When a healthcare employee is granted access to information, only the relevant information should be presented by the system, providing better overview and highlighting critical information.

After a study of relevant laws, standards and other publications, this report presents some necessary pre-requisites for a role-based model. Two test cases were developed in order to test the role-model. Hierarchies of roles and information classes were developed based on these cases.

A role model was developed and formalized with set statements and functions. The role model was then tested with the cases.

The results indicate that using the same role-model for access control and information ranking is possible. It was also concluded that realizing patients' individual objectives seems to be easier by using an access control list, than by using global roles alone. Although only used as an extension to the test cases, the development of a role-hierarchy, a information-hierarchy, and ranking functions demonstrates that hierarchies makes it easy to administrate access and ranking in the same model.

The model can not express guidelines, and has a limited ability to express time. In its present state, the role model can not be legally used as the sole access control method in a record system, because pure implementations would not comply with Norwegian law or standards for access to healthcare information.

The report has been influenced by the fact that it acts as a pre-study to the authors' master's thesis.

Preface

This report is a result of the work done in the subject TDT700 Healthcare Informatics, Specialization, autumn 2004. The assignment gives study credit equivalent to half a semester. The assignment was given by the Department of Computer and Information Science (IDI) at the Norwegian University of Science and Technology (NTNU), and the assignment was formulated by Associate professor Øystein Nytrø. Nytrø has also been our supervisor during the work on this assignment. This assignment will lead into our master's thesis for the spring of 2005.

For follow-up, contacting the authors, and further work, see
http://gunnarre.nvg.org/studies/rmhc/.

We wish to thank the department for providing the facilities, and the following for support and inspiration:

- Øystein Nytrø, our supervisor
- Students sitting in room itv060
- Torbjørn Nystadnes at KITH
- Hans Jørgen Varfjell at KITH
- Trond Hanken Urke at the Norwegian Board of Health (Helsetilsynet)

....
Bjørn-Erik Stenbakk

....
Gunnar Rene Øie

Trondheim
26. November 2004

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"Information is the oxygen of the modern age.
It seeps through the walls topped by barbed wire,
it wafts across the electrified borders."
Ronald Reagan

"Hyperventilation is rapid or deep breathing, usually caused by anxiety or panic.
This overbreathing, as it is sometimes called, actually leaves you feeling breathless."[1]
MedlinePlus.

"I have a theory about the human mind. A brain is a lot like a computer.
It will only take so many facts, and then it will go on overload and blow up."
Erma Bombeck

Information overload blowing up a human brain is about as rare as information overload blowing up a computer.[2] But, if you don't have some way of structuring or picking out the information that you want, having all the information in the world is about as worthless as having no information at all. The task of finding relevant information becomes even harder if you don't even know *what* information you want. At worst, life-critical and useful information could drown in irrelevant information.

Your chance of survival as a patient could improve when freeing your doctor and other health care professionals from this information overload. Could the access control rules for healthcare information, existing for your privacy, be implemented with the same method as relevance ranking? That is what our project seeks knowledge about.

This chapter contains some context and motivation for our project, and concludes with an outline of this report.

[1] Actually, hyperventilation symptoms are caused by a lack of CO_2, not by a surplus of oxygen.
[2] Similes could be an epileptic attack and a distributed denial of service-attack.

1.1 Context and motivation

The healthcare sector is very information intensive. All encounters with and treatment of a patient needs to be documented. Data needs to be acquired from a multitude of equipment and persons, and then shared among and interpreted by healthcare workers. Healthcare professionals also spend significant amounts of time recording and searching for information.

There is a lot of work going on to provide decision support in healthcare, structuring and standardizing it. One could ask wether it would be more cost-effective to just provide healthcare workers with a search engine and letting them write free text in the record. An argument against this is that medical information has to be shared between a wide variety of people, from medical doctors, to administration, to the patients themselves. Also, having critiquing guideline systems requires the record information to be structured in a way that a computer can make sense of.

An issue related to traditional decision support is the emphasizing of information. If decision support is not strong enough to make a diagnosis on its own or noticing a mistake, it could still be strong enough to emphasize information that is likely to be important.

All information about patients must be protected to ensure its confidentiality and correctness. On the social and legal level, information is protected by law and the morals of healthcare workers. However, as availability is increased by the use of information technology, there is an increase in both the opportunity for malicious activities and their magnitude. Because of this heightened risk, information technology must take part in managing security. Just morals will not be enough to keep information safe.

In this project, we wanted to find out wether or not a combined model for roles in ranking and access control makes sense, and looking at how one would go about in designing it.

1.2 Report outline

The first chapters will provide background information and theory needed in the project. The following chapters will include prerequisites, case description, design and results. At the end this report we include two chapters for discussion and conclusion.

This report proceeds, after this introduction, with chapter 2 containing the assignment description and a set of research goals formulated to help us concretize our work. Chapter 3 will give an overview of the Norwegian healthcare system, the laws and regulations and the electronic health-record standard (EHR-standard). In chapter 4 we describe the theory behind role-based access control(RBAC) and we look at the use of RBAC in the healthcare sector. We also describe some theory behind role-algebra. Chapter 5 will provide some theory around user interfaces and information ranking.

In chapter 6 we define some prerequisites in the form of a set of hypotheses, requirements from RBAC and the EHR-standard and our concepts of role-hierarchy, information-hierarchy and information ranking. In chapter 7 we introduce the reader to a case which our role-model is being tested against, the case description also includes a constructed role-hierarchy, information hierarchy and a table with information ranking. In chapter 8 we present our design choices and present the language and the beginning of a formal role-model.

Chapter 9 contains results from applying our formal model to the cases.

Chapter 10 and 11 contains the discussion and conclusion respectively. These two chapters will be a summary of the choices we have made and the effects of these, and what we have and have not accomplished with our work.

Chapter 2

Assignment

Our assignment was originally given in Norwegian and the original assignment text can be found in the appendix. During the work the assignment text has been revised with approval from our teaching supervisor. We have in addition developed nine research goals.

2.1 Role-models in healthcare service

The healthcare service is very information intensive. IT-based solutions may help professionals by displaying and emphasizing important information. A condition for IT-based solutions is to have satisfying control of access for updating, making, transmitting, reading and signing sensitive information. In healthcare service it is practical to regulate access in relations with roles and not individual persons, professions or organizations. These roles should also be used to emphasize information, improving efficiency and quality of care.

The assignment consist of studying current and future standards for health information and propose a role-based model for access to and emphasizing of information.

2.2 Research goals

To guide our work and to help evaluating our results, we have defined the following research goals:

1. Provide some information on organization of healthcare in Norway, the laws and regulations and the EHR-standard to give a basic idea of the complex environment we are working in.

2. Review the state of the art of role-based access control.

3. Create a set of hypotheses to makes sure that the model has qualities other than merely being possible to implement.
4. List a set of prerequisites for a role-model
5. Find or create a useful formalism or modelling language for describing the model.
6. Develop the role-model according using the prerequisites and the formalism.
7. Validate the model using the case of medical regimens.
8. We will make suggestions for further work.

Chapter 3

Healthcare in Norway

In this chapter we will give an introduction to the Norwegian healthcare system and the laws that regulates this system. We will also introduce some of the IT-services in which will be or is established in the healthcare sector. At the end of this chapter we will review the Norwegian electronic health record standard and in particular the part of the standard that presents access control.

3.1 Healthcare providers

The Department of Health [Hd04a] controls the regional health authority through laws, regulations, general meeting and steering document. The Department of Health also appoints the leader of the regional health authority. The Norwegian healthcare system can be divided into primary care and specialist care.

3.1.1 Primary care

The local municipal authority is responsible for providing primary care. Some of the services may be provided in cooperation with the special healthcare service.

3.1.1.1 General Practitioner

The General Practitioner (GP) is the patients most important, and often the earliest encounter, with the health service. Everybody who ask for a regular general practitioner will have one appointed. The general practitioner has the general medical responsibility for his/her list of patients during day time, including emergency care. The GP also cooperates with the other services in primary care and social services when needed.
If emergency care is needed during the evening or the night it is the casualty clinics responsibility.

If the patient needs a more comprehensive type of care the patient is referred to a specialist; private or public.

3.1.1.2 Other services provided by Primary care

Primary care also includes physiotherapy, nursing home, midwife services, and nursing services. These services are usually performed in cooperation with the general practitioner.

3.1.2 Special healthcare

Special healthcare services includes public owned hospitals, psychiatric institutions, ambulance services, emergency dispatch, hospital pharmacy, laboratories and some institutions for addicts. Five regional health authorities are each responsible for the specialist health services in a geographical area. These areas are North, Middle, West, East and South. The hospitals are also organized in health authorities which are controlled by the regional health authorities.

3.2 IT services

The new Norwegian Health-Network provides the infrastructure for electronic interaction between disparate institutions. The Healthcare Personnel Registry and the Healthcare Unit Registry are today rather simple services, but could possibly be built further into a national role and addressing database for health information.

3.2.1 Health network

Norsk helsenett (NHN) [hA04] is a closed network for electronic communication and interaction in health- and social sector in Norway. NHN is owned by the regional health authorities, each with an equal share. It was established on the Sept. 27th. 2004. The reason for establishing NHN, described in Si@ [oh01], is a good foundation for electronic interaction between health personnel, and between health personnel and patient based on two elements. Firstly a physical infrastructure with a satisfactorily capacity and coverage. Secondly a set of basic services to arrange for the interaction. A national health network shall ensure data quality, information security and privacy when exchanging sensitive information. The entire health sector will be tied together by this network.

3.2.2 HPR

The Norwegian Healthcare Personnel Registry ”Helsepersonellregisteret (HPR)” is a publicly availably registry of living persons authorized or licensed to provide healthcare or animal care. Authorization is mandatory for healthcare workers having protected titles, like ”MD”, ”dentist”, ”veterinarian”, ”nurse”, ”nurse's assistant”, ”health secretary”, etc. Those who do not need authorization may anyway apply for a voluntary license. If this license is granted, this voluntary license will also be registered in the HPR.

3.2.3 HER

The Norwegian Healthcare Unit Registry, ”Helseenhetsregisteret (HER)”, is being developed as a national registry of health institutions, wards, units and personnel that can receive electronic messages. The information in this registry is to be held in a central database, and used by various information systems. The HER provides health records and other information systems with address information.

3.3 Laws and regulations

This section describes in short the different laws and regulation which controls the activity in a healthcare authority. Special attention is given to how the laws regulate the access to healthcare information. The short version is that access should be given only in the extent necessary for performing the given task or access is only to be given to the person accountable for data processing, people who by agreement manage the healthcare information on behalf of the owner, or who work under their authority.

3.3.1 The patient rights act

The purpose of the patient rights act [Hd03e] is to ensure the population equal access to healthcare of good quality. This is done by stating the patient's rights towards the health service. The act regulates among other things the patient rights regarding consen to the use of a medical record. The act gives patients the right to view their own medical records. Exceptions are made where viewing the record is not advisable due to the risk of life or serious health damage. The act provides for a better opportunity to correct or delete errors or incriminating informationm and it requires that the patient gives approval to use the healthcare information.

3.3.2 The healthcare personnel act

The purpose of the healthcare personnel act [Hd03b] is to contribute to security for patients, to the quality of healthcare, and to trust in healthcare personnel and health services. The act states that all healthcare personnel which provides healthcare is obligated to keep a medical record. Exception are made when the personnel are guided through the procedure. The act also defines healthcare personnel and authorization needs. The act stipulates that the medical record shall contain relevant and necessary information about the provided healthcare and show who has recorded the information. Erroneous, insufficient or improper information has to be corrected. Information that is erroneous or misleading and feels incriminating has to be deleted if the information is not clearly needed for providing healthcare. The main principle in this act is patient confidentiality, but information can be passed on if the patient approves. The medical record or the information in it can be passed on to other healthcare providers that need it unless the patient opposes this. The record has to clearly state to whom access was given. There are numerous of exceptions to the patient confidentiality, for instance when information is anonymized, personnel assist in electronic editing of information and access when service and maintenance of equipment are required. Worth noticing is that rules regarding an electronic medical record can be given in a regulation to this act.

3.3.3 The personal data act

The purpose of the personal data act [opd00] is to prevent the privacy of the individual person to be violated during processing of private information. The act contains all the fundamental regulations for all processing of private information. The act and its regulations are complementary to the healthcare registration act.

3.3.4 The healthcare register act

The purpose of the healthcare register act [Hd03c] is to give the healthcare service and healthcare administration information and knowledge without violations of the right to privacy. This means that this act needs to be seen in relation with the personal data act. The act regulates both complete and partial electronic processing of healthcare information. The act stipulates that every processing of healthcare information havs to have a defined purpose. The act also says that healthcare information in a medical record and other treatment adjusted health registers about the same patient can be compared.

3.3.5 The regulation regarding patient medical records

The regulation regarding patient medical records [Hd03a] gives further instructions regarding the duty to keep a medical record, the authorities' obligation to set up and organize the medical record, and your right to view your own medical record. The regulation stipulates that each medical record must have a person with superior responsibility for it. However, it is not specified who this person might be.

3.3.6 Other acts and regulations

These acts and regulations also give direction for the healthcare services and the use of a electronic health record:

- The special healthcare service act [Hd03f].
- The psychic healthcare act [Hd03d].
- The purpose of the electronic signature act [Hd02].
- The act of healthcare enterprise [Hd04b].
- The archive act [okd01]

3.4 Norwegian EHR-standard

The Norwegian EHR-standard [Nys01] – Elektronisk pasientjournal standard (sic) - was developed by KITH as part of standardization program run by the Norwegian Ministry of Social Affairs and Health.

The standard contains requirements and recommendations, and general information models. The requirements stated are based on Norwegian law and regulations, while recommendations based on best practice are also given. The information models are very general with regard to specific medical cases.

The most important requirement met by the standard is that its specifications and recommendations, when implemented, produce systems and procedures that comply with Norwegian laws and regulations. Among the influences from the law is the principle that the patients own their own records, that information should only be available on a need-to-know basis, and that the records should enable healthcare workers to give the best possible care. The CEN ENV 13606 pre-standard for exchange of record information has been used as guide in developing the Norwegian standard, it does not follow the CEN pre-standard down to every detail.

The main issues addressed by the standard are:

- A data format for final archiving of electronic health records.
- Access control mechanisms
- General information structure
- Handling coding and classification

The standard has not been detailed with respect to specific medical procedures.

We'll now look into how the Norwegian EHR-standard addresses access control.

3.4.1 Requirements

Giving all healthcare workers full access to all patient records all the time is considered too prone to abuse, and would violate Norwegian law. On the other hand, healthcare workers can not spend their day asking their supervisors to approve each and every access to an electronic health record. The record system must in some way allow access that is neither too open, nor too restrictive. This implies that the access control model should have a high granularity. It must also support the many ways in which access may be granted.

3.4.1.1 Actions

All access to the record must be part of a of a necessary action or event. The actions might be part of the treatment of a patient, aggregated data collection for epidemiological studies, or any other endeavor that either uses or produces medical information, i.e. reading or writing in the record. The action must have an expressed and valid goal. The standard specifies that the EHR must have several action templates, with each of these action templates clearly specifying what information the action will require and what information the action will produce. Thus, when a healthcare worker is tasked with performing an action, the action is entered into the EHR system, and the system will automatically grant the necessary access, according to the action template. The action template must specify what categories of information are needed, and wether the access includes writing, changing and deleting information. If information is changed or deleted, the reason for the editing must be stored in the record.

There is a minimum number, required by the standard, of action templates that must exist. One of these is the permanent action that allows emergency access to the record, and another is an action that gives some roles the ability to order new actions.

Actions may be specified to be performed by a specific person or role, or the action may be specified to be performed by any person. The action template will specify which roles and authorizations are needed to perform the action and thus access the record components. The standard also supports healthcare workers performing actions on behalf of another. E.g. a front room secretary may update the record on behalf of the doctor. The action may also be limited to patients who are treated at the same healthcare unit.

3.4.1.2 Access to global information

Access to read and update guidelines and coding is also handled by the model. This information is not privileged patient information, but it must be protected from unauthorized tampering and there may be licensing agreements in place limiting who gets to read the information. Special role templates grant this access.

3.4.1.3 Initiating records

Role templates also grant the right to create a new patient record, specify who is responsible for the record, restricting access to it on behalf of the patient, granting access on behalf of the patient, and changing personal details. The person responsible for the record of the patient usually has full access to the contents of the record, except when access to parts of it has been restricted by the patient.

3.4.1.4 Administrator access

Access for system and maintenance work must also be granted through the access control model, without sacrificing security for expedience. System and maintenance work should rarely require access to the records of real patients. System changes should be verified on test data, before affecting real patient data. Special role templates grant this access. Some role templates also give access to change the organizational structure, and create and change role templates, action templates and roles, and assigning healthcare workers to roles.

3.4.1.5 Multiple roles

A user/healthcare worker who has multiple roles must able to switch between these roles. When a role has several actions to perform, the worker must be able to choose which action to perform first. And when an action has been completed, this must be registered in the system. The worker should have reading access to information that was written into the record by her/him-self.

3.4.1.6 Time controls and suspicious activity

Actions may be given time limits for beginning and ending, thus reducing the timeframe for potential misuse, and reminding users/healthcare workers of actions that have been forgotten. For larger organizations, the standard requires the EHR system to produce reports about actions that have yet to be started, and about those actions that are yet to be completed. The system must also report all emergency access, and self-approved actions possibly used to extract information illegally.

3.4.1.7 Patient access and approval

The record must reflect a patient's request to see the record, wether or not this request was approved, or wether the record was seen by the patient or somebody acting on the patient's behalf.

The patient has a right to limit access to certain parts of the record, and may specify for which service provider groups, roles, or specific individuals the information is to be available or unavailable. This may apply to the entire record, or just to certain topics. If the patient has given no instruction regarding access, the general rule applies: Access is granted to the role performing an action.

3.4.1.8 Other concerns

All access, both reading and entering of information, must be logged. The log must reflect who performed the access, at what time, and because of what purpose/action the record was accessed.

If the patient requests deletion or correction of information in the record, and the request is denied, this fact should be stored in the record.

If access to certain components is barred, the user must be informed of the reason why access is denied without giving away clues to the nature of the information.

Access groups Healthcare workers The patient Administration Quality assurance and investigative authorities

Thus, the access control model in the Norwegian EHR-standard takes into consideration

- Which patient the access concerns.
- Who is attempting access (role, abilities, authorizations, and unique identity).
- Which unit the patient is being treated at, and wether it's the same unit from which the access is attempted.
- Why the access should be granted (actions, action templates, goals).
- What data is accessed (which components are accessed)
- What mode of access is attempted (reading, writing or changing information).
- Wether or not the patient has granted special access to, or restricted certain parts of the record.

Some of the information components (guidelines, codes) are not related to a patient, and are thus subject to less stringent controls.

3.4.2 Information model

Part two of the standards document is an information model for archiving records, but not for live use in record systems. The information model is given as several UML composition and aggregation diagrams, with varying levels of detail; and as tables describing the field contents of each component. The access control specific part of the standard is centered around the components action (”tiltak”), action template (”tiltaksmal”) the patient’s consent (”samtykke”) and role (”rolle”). (Norwegian terms in parentheses.)

Chapter 4

Access control

In this chapter we will give a short introduction to different access control models before concentrating on role-based access control. The review of RBAC contains a description of characteristics and constraints, a review of the proposed NIST-standard and the use of RBAC in healthcare. The chapter ends with a overview of role-constraints languages and role-algebra.

4.1 Introduction

Access control is the process of granting subjects (users or processes) access to perform operations on objects (files, processes or data fields) [Gol99, pp. 30–45]. Role-based access control is an alternative to the more traditional form of access. There are three basic access control models [Fra03],[PHS03, pp. 565–589]:

- Discretionary access control (DAC), usually identity based. An access control matrix gives full control over what operations any subject can perform on any object; but a full matrix may be too complex to be practical for security management. To make access grants simpler, permissions are often given per object, as an Access Control List, or per subject, as a Capability. The Access Control List is used by the owner of the object to give other subjects access to the object. Capabilities are defined by the system administrator and give subjects access to objects based on the needs of subjects.
- Mandatory access control (MAC), based on levels, and found primarily in the military or other highly sensitive systems. This is based on classifying objects according to the sensitivity of the data and giving subjects a clearance, or authorization level. A user is granted access only when the user and object have corresponding clearance levels. The access method may permit reading from lower levels, but not reading from higher levels. It may permit writing to a higher level, but not writing to a lower level.
- Non-discretionary access control, usually role-based, centrally administrated with

authorization decisions based on the roles individuals have within the organization. The system administrator grants and revokes system privilege based on the user's role.

The basic models may be used in pure form, or be combined with each other to provide more fine-grained access control. On the flip-side, one could reduce granularity in other ways, e.g. by defining a privilege as a ordered set where a higher privilege includes all lower privileges.

4.2 Role-based access control

The preferred information system of use for RBAC would exhibit the following characteristic [RC99]: For users a large number of users, few security administrators, and frequent change of job responsibility. For data and applications there are large numbers of data and sharing objects based on job functions. For enterprises the data is owned by the enterprise, controlled by security administrators, before and after the fact audit, and periodic assessment of access control policy enforcement necessary. Ferraiolo, Cugini and Kuhn [FCK95] believe the principal motivation behind RBAC is the ability to express and enforce enterprise-specific security policies and streamline the typical burdensome process of security management.

The essence of role-based access control is that system permissions are assigned to defined roles rather than to individual users. And therefore the basis of RBAC is the concept of a role. A role is a type grouping that categorizes subjects based on various properties. These properties pertain to the functional responsibilities of the user in the organization.

4.2.1 RBAC Characteristics and policies

RBAC policies are described in terms of users, subjects, roles, role hierarchies, operations and protected object. We have listed some of the characteristics stated by Ferraiolo, Cugini and Kuhn[FCK95]:

- Role-hierarchy defines roles that have unique attributes and that may contain other roles.
- Role authorization can be subject to the following
  1. The user can be given no more privilege than is necessary to perform his/her job (principle of least privilege).
  2. The role in which the user is gaining membership is not mutually exclusive with another role for which the user already possesses membership (static separation of duty)
  3. The numerical limitation that exists for role membership cannot be exceeded (cardinality property).

- Role activation involves the mapping of a user to one or possibly many roles. A particular role for a user can be activated if:
    1. the user is authorized for the role being proposed for activation.
    2. the activation of the proposed role is not mutually exclusive with any other active role(s) for the user.
    3. The proposed operation is authorized for the role that is being proposed for activation.
    4. The operation being proposed is consistent within a mandatory sequence operation.
- Role execution of an operation can take place only if the subject is acting within an active role.
- Dynamic separation of duty can be provided in RBAC if a subject can become active in a new role only if the proposed role is not mutually exclusive with any of the roles in which the subject is currently active.
- Operation authorization can only be granted to a subject if the operation is authorized for the subjects proposed active role.
- Operational separation of duty requires that for all the operations associated with a particular business function, no single user can be allowed to perform all of these operations.
- Object access authorization requires subject access to RBAC objects to be controlled. A subject can access an object only if:
    1. The role is part of the subjects current active role set.
    2. The role is allowed to perform the operation.
    3. The operation to access the object is authorized.

4.2.2 RBAC constraints

RBAC also enables administrators to place constraints on role authorization, role activation and operation execution. In RBAC separation of duty is a well known control principle in management. Separation of duty can be seen as mission critical combination of tasks required to be performed by different people and it prevents accidental or malicious violation of business requirements [Cra03]. Separation of duty can be divided into static, dynamic and historical separation of duty. A more detailed description of static and dynamic separation of duty is given in the description of the NIST-standard. Historical separation of duty is, as the name implies, based on historical or logged states and events in the system. Historical constraints are more difficult to enforce than static and dynamic constraints, but a proposed method of enforcing this would be to create a blacklist of request that would cause a constraint to be violated. This method however raises some new matters to the prior, for instance a poorly specified set of constraints may lead to situations where no user can invoke a particular method on a particular object.

4.3 The proposed NIST-standard

To give a more technical view of RBAC we have included a review of the NIST-standard [FSG+01] , [FKC03]. The NIST-standard is a proposed standard for RBAC. The proposed standard tries to resolve a situation where no single authoritative definition of RBAC exist. It does so by unifying ideas from a base of frequently referenced RBAC models, commercial products and research prototypes. It does not try to standardize RBAC features beyond those that have achieved acceptance in the commercial marketplace and research community. The concept of RBAC embodies notions, like groups in operating systems, privilege groupings in DBMS and separation of duty, in a single access control model in terms of roles and role hierarchies, role activation, and constraints on user/role membership and role set activation.

4.3.1 Component overview

The NIST-standard is organized in two main parts: the RBAC reference model and the RBAC functional specification. These two main parts are in turn organized into four RBAC components. The basic concept of RBAC is that users are assigned to roles, permissions are assigned to roles and users acquire permissions by being members of roles. Core RBAC embodies the essential aspects of RBAC and is required for any RBAC system. Core RBAC includes requirements that user-role and permission-role assignment can be many-to-many. Core RBAC also includes requirements for user-role review whereby the roles assigned to a specific user can be determined, as well as users assigned to a specific role. The concept of user sessions allows selective activation and deactivation of roles. Core RBAC also requires that users be able to simultaneously exercise permissions of multiple roles. Hierarchical RBAC adds requirements for supporting role hierarchies. The NIST-standard recognizes two types of role hierarchies: general hierarchical RBAC and limited hierarchical RBAC. Static separation of duty (SSD) relations are used to enforce conflict of interests policies. Conflict of interest can occur in a role-based system because of a user gaining authorization for permissions associated with conflicting roles. Preventing this can be done by SSD, that is, to enforce constraints on the assignment of users to roles. An example of this is the requirement that two roles be mutual exclusive. SSD is, in the NIST-standard, defined both in the presence and absence of role hierarchies. Further, SSD is defined as a binary relation $(role\_set, n)$ where no user is assigned to n or more roles from the role set. Dynamic separation of duty (DSD) relations, like SSD relations, limit the permissions that are available to a user. DSD differ from SDD by the context in which these limitations are imposed. DSD requirements limit the availability of the permissions by placing constraints on the roles that can be activated within or across a user's sessions. DSD relations also define constraints as a binary relation $(role\_set, n)$ where n is a natural number $n >= 2$, with the property that no user session may activate n or more roles from the role set.

4.3.2 The RBAC Reference model

The reference model provides a rigorous definition of RBAC sets and relations. It has two primary objectives: to define a common vocabulary of terms for use in consistently specifying requirements, and to set the scope of the RBAC features included in the standard. There is a reference model for each of the four RBAC components.

> Each model component is defined by the sub components:
>
> - a set of basic elements sets;
> - a set of RBAC relations involving those element sets [...]; and
> - a set of Mapping Functions which yield instances of members from one element set for a given instance from another element set.

[FSG+01, p. 232]

All figures in the following paragraph is from the proposed NIST-standard [FSG+01]: The core RBAC reference model is shown in figure 4.1. This includes sets of five basic data elements called users, roles, objects, operations, and permissions. Figure 4.1 illustrates the user assignments and permission assignments relations, the arrows indicate a many-to-many relationship. This arrangement provides great flexibility and granularity of assignment of permissions to roles and users to roles. This strengthens the applications of the principle of the least privilege. Figure 4.2 shows the core RBAC model expanded to hierarchical RBAC. This standard include, as described earlier, both general and limited role hierarchies. Figure 4.3 adds static separation of duty to the core RBAC model and figure 4.4 adds dynamic separation of duty to the core RBAC model.

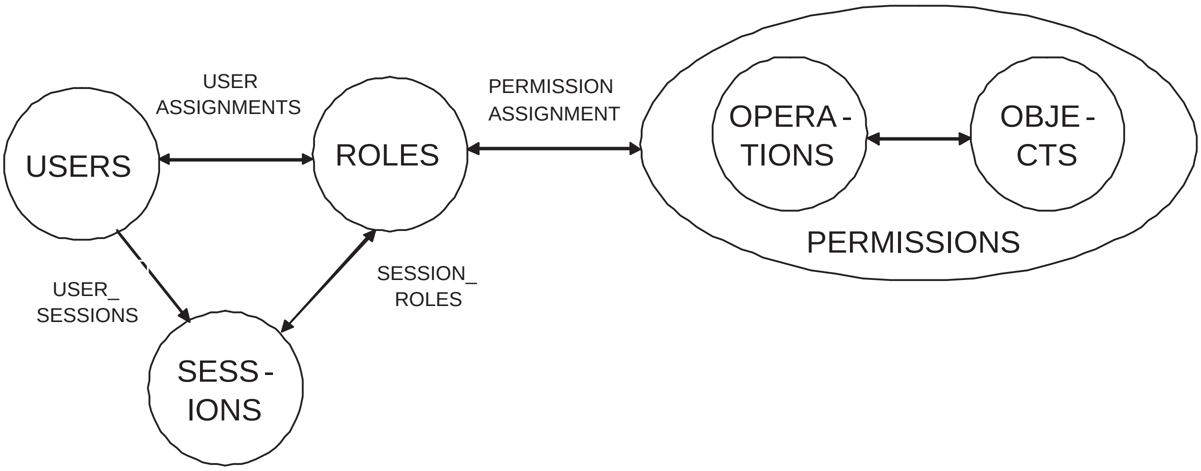


Figure 4.1: Core RBAC

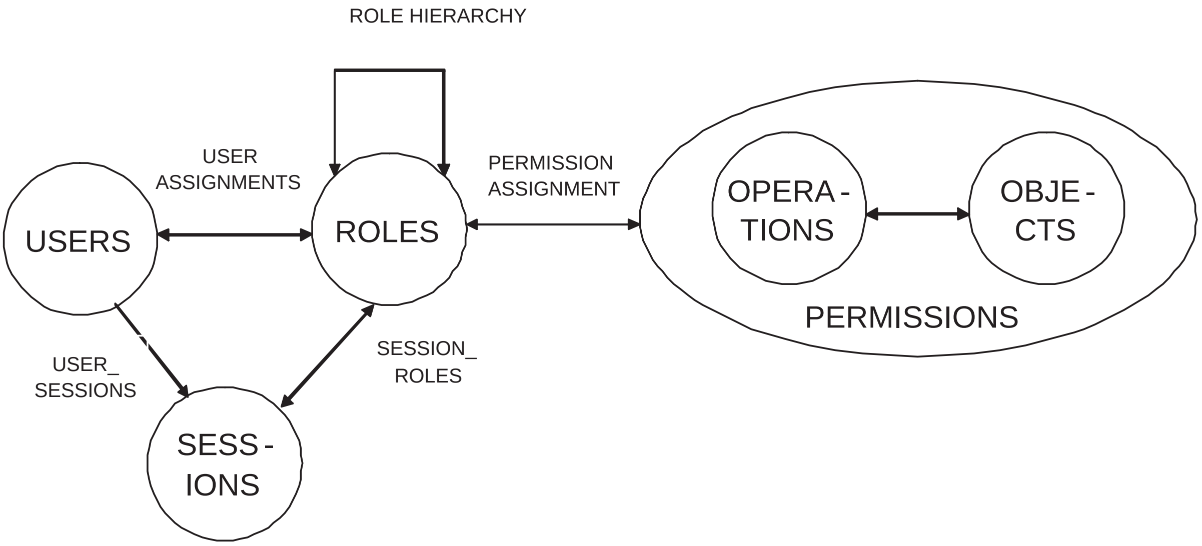


Figure 4.2: Hierarchical RBAC

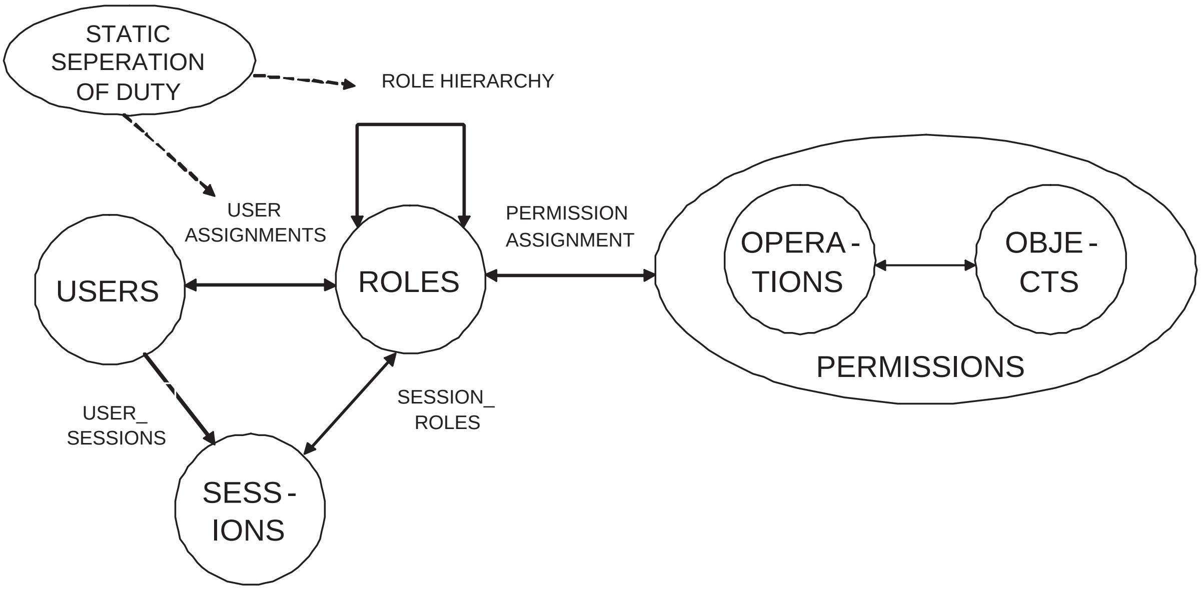


Figure 4.3: SSD RBAC

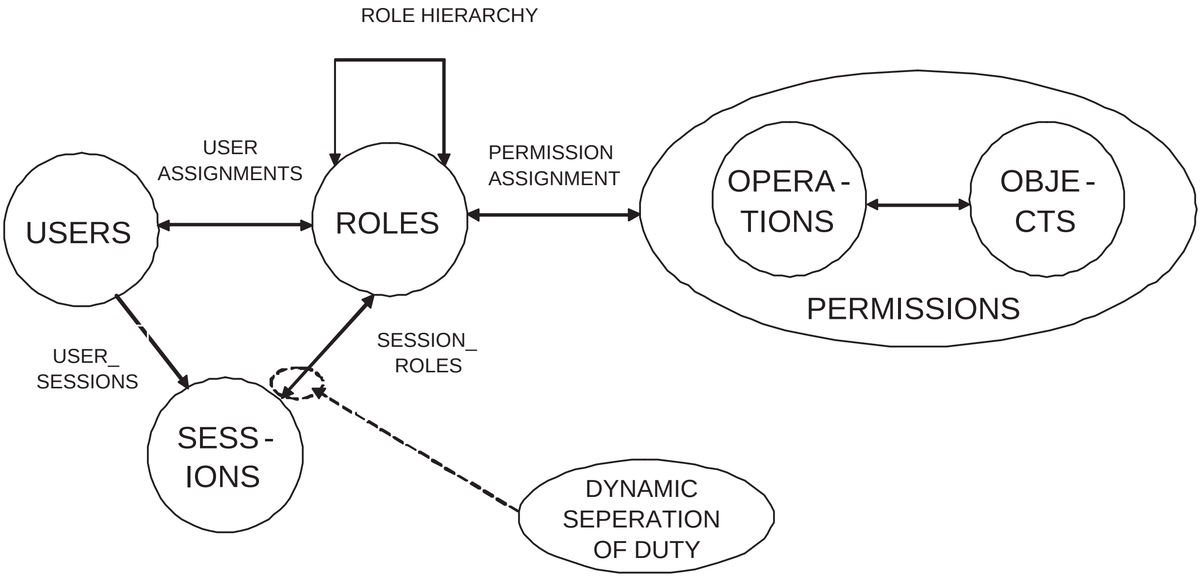


Figure 4.4: DDS RBAC

4.3.3 RBAC functional specification

The functional specification to the NIST-standard of RBAC provides an overview of the functionality in meeting the requirements for each of the components defined earlier.

”The three categories of functions in the RBAC functional specification and their purpose are:

- Administrative Functions: creation and maintenance of element sets and relations for building the various component RBAC models;
- Supporting System Functions: functions that are required by the RBAC implementation to support the RBAC model constructs[...]; and
- Review Functions: review the results of the actions created by administrative functions.

”[FSG+01, pp. 241–242]

4.4 RBAC in health care

The description given of the preferred information system to use RBAC in suites the healthcare industry very well. The characteristics for users, data and enterprises fits almost every description given above.

4.4.1 Requirements

There are several important reasons why RBAC should be used or is in fact needed in healthcare. The RSA security [Inc02] lists these reasons [1] for using RBAC in healthcare:

- Risk management for cost avoidance.
- Operational efficiencies for improved patient care and cost reduction - which means implementing efficient ways of securely accessing information.
- New services and new customer relationship management for revenue generation.
- Compliance with acts and regulations for cost avoidance

4.4.2 Using RBAC in healthcare

When implementing RBAC in healthcare all the roles of the healthcare organization must be defined and RBAC must support as many roles and instances of roles as necessary. For effectiveness this can be done with the use of inheritance, this implies that changes will be made to subclasses when its made to the general classes. Many of the users will also fill many different roles and therefore the system must allow for assignment of multiple roles. Other aspects of RBAC in healthcare are control of activities different roles have i.e. read, write and execute, generating queries to view and edit permissions granted to different roles and the ability for logging of activity. In addition data may be classified into various types i.e. clinical, administrative and according to sensitivity (e.g. records containing information about HIV, abortion and mental health). The following two sections describes in short two projects/products which use RBAC in healthcare.

4.4.3 The RSA ClearTrust Web Access Management solution

The RSA ClearTrust Web Access Management [Inc02] solution delivers many of the critical features that are necessary for role-based access control in healthcare. It was designed to:

- Provide salability for supporting very large user populations found in HCOs,
- Allow unlimited numbers of roles to match the large number of job functions in HCOs,
- Support inheritance so that if a change is made to the authorizations for a particular role - all users and subclasses within this role are automatically changed,

[1] The Health Insurance Portability and Accountability Act requires implementation of RBAC, but [Inc02] lists these other reasons for implementing it

- Support the implementation of a HCOś specific policies using Basic Entitlements and Smart Rules,
- Allow the creation of different entitlements for individual users within a role who have unique privileges or restrictions,
- Control access to resources and the actions performed including viewing, creating, editing, signing, releasing, amending, copying and archiving a file,
- Provide the ability to generate queries in order to view and edit permissions,
- Provide delegated user administration for distributing administration tasks to various business units found in large HCOs,
- Support extensive logging capabilities for auditing purposes,
- Support multiple authentication methods used in HCOs such as password, tokens, smart cards, LDAP authentication, and biometrics,
- Allow the use of different authentication methods for different roles accessing the same portal (e.g., patients could use passwords, while physicians would require RSA SecurID tokens),
- Easily integrate with the diverse environments found in HCOs based on open architecture, certified interoperability with best of breed products, and full support for industry standards such as Security Assertions Markup Language (SAML),
- Provide transparent Web single sign-on access to the many applications found in a HCO, for increased efficiency, as users do not have to continually sign on as they move from one application to another, and enhanced security, as users do not have to remember multiple passwords, and
- Provide a complete identity management solution through strategic partnerships with provisioning and data store solutions.

4.5 Role constraints languages and role algebra

Roles may be described in informal language, or in semi-formal modelling languages. However, if they are to be validated and used, roles and operations on roles must be defined in a formal language. This section provides a brief look at some such formal constraint languages and algebra for roles.

4.5.1 RCL2000

Ahn and Sandhu describe [AS00] a formal role constraints language, called RCL 2000 (Rickle two-thousand). Building on the basic elements and functions of RBAC96 (a precursor to the NIST RBAC standard), RCL 2000 is a language for specifying SoD between roles, users and even between permissions. RCL2000 is limited to representing to static SoD or sessions where there is no dynamic activation of roles. RCL2000 does not express time and state constraints. However, it is able to express obligation constraints in addition to the usual prohibition constraints.

Defining constraints in RCL 2000 is simpler than defining them in first order predicate logic. A definition written in RCL 2000 may be converted to a restricted first-order predicate logic with only universal quantifiers. This conversion is two-way. A reduction algorithm converts from RCL, while a construction algorithm converts in the other direction.

4.5.2 Role algebra for agents

Karageorgos , Thomson and Mehandijev [KTM03] use role-models as part of a knowledge base for agent system design. They demonstrate a simple algorithm that combines roles into a minimal number of agents without violating rules for separation of duty and performance rules. The algorithm works by moving roles that violate constraints to other agents, or to new agents initialized by the algorithm. A simple algebra for roles and agents is defined and used in the algorithm.

Chapter 5

User interfaces

This chapter briefly puts relevance ranking in the context of user interfaces and processes.

A dynamic relevance ranking of record information could be used in the design of intelligent user interfaces. Having information about what pieces of information are most important, the user interface can show just the most important information at a glance. The interface may allow the user the option to also view the less important information. A key feature of intelligent user interfaces is that may they show a different ”face” to the user depending on in which part of the working process the user is.

5.1 Business and user modelling

Analyzing business processes and formally representing them [KP00], is helpful not only in re-designing and improving business practices, but also in developing workflow systems and their user interfaces. Roles and goals are important concepts in business modelling.

User modelling has traditionally analyses the goals of an individual user, and identified what operations the user needs to perform in the accomplishment of that goal [Kie99], with the purpose of improving user interfaces.

5.2 Relevance Ranking

Bayegan [Bay02], with Nytrø and Grimsmo [BØNG01], propose a framework to be used in ranking record information. ”Decision frames” are the selections of what information should be shown at each time. Bayegan proposes classifying record information in a content ontology called ”CareActType”, and that each decision frame would be a selection of classes from this ontology. Thus, after selecting a medical problem, each information object related to that problem would be selected for display depending on to which class it belongs. The top super-classes in the CareActType information classification are actually not information types in themselves, but are named by phases in the care process. By relating information classes to simple phases, the content ontology contains a choice of what information is important in what phase.

To know which particular decision frame to show, the record system needs to know the phase that the user is in. Bayegan proposes using ”traces” to establish process knowledge without disturbing the user. The record system could analyze traces, action patterns, in the information that has already been entered into the system. Trace recognition could become more advanced depending on the quality of artificial intelligence, becoming able to recognize not just what phase of care the user is in, but wether treatment is working and the patient is getting better. This is a difficult task, but one that becomes simpler if we satisfy ourselves with a small number of defined phases of care.

Another way that the system can keep up with phases, is having the user explicitly change phase. An common example of this is configuration ”wizards” used when installing software applications.

Chapter 6

Prerequisites for our model

In this chapter we define some prerequisites in order to clearly identify what our model is supposed to demonstrate and how it is supposed to demonstrate it. The idea of creating a model is to clearly explain an underlying concept. The goal of our model is to demonstrate that the role is the central concept for handling the complexities in both relevance ranking and access control. This implies that if a dynamic role has been defined by the combination of relevant dynamic and static characteristics, the dynamic role contains all the information needed to rank and control the access to information objects. By providing a set of hypotheses we hope to show what specific qualities we want our model to demonstrate, other than it being possible to implement. Identifying requirements from RBAC and the EHR-standard helps us fit our model with the context of RBAC and the healthcare sector.

In addition we will clarify a few concepts regarding roles, information classification and information ranking.

6.1 Hypotheses

We will create a model that has qualities other than simply being possible to implement. To show this we have created a set of hypotheses that we will test our model against. This implies that an important aspect is that we will be able to prove or reject the hypotheses; thus we have described the method for testing or the testability of the hypotheses.

6.1.1 Ranking of information

Problem: The ranking of information has to be done without any extra input from the user, only relying on information already entered into the EHR system.

Hypothesis: The role-model is sufficient to provide information ranking based on information in the record.

Why: It is essential that the information ranking gives the information that is preferred by the role without any other input from the person being that role.

Testability: The role-model is build up by logic and can be tested with paper to se what information that appears and disappears from our test case.

6.1.2 RBAC and information ranking

Problem: If access control and information ranking can't be combined different models will have to be created.

Hypothesis: It is possible to use the same model for both access control and information ranking and this reuse will be a benefit both for system developers, administrators and users.

Why: Ranking and access control share the activity of selecting information for view dependent on who the user is.

Testability: By creating a model for access control and use this model for information ranking we will prove that this is possible.

6.1.3 Role constraints

Problem: There are rules for separation of duty, temporal constraints, and other constraints in how roles are defined.

Hypothesis: It is possible to build roles without violating the rules.

Why: Without constraints properly implemented, the model would become useless for access control.

Testability: Introduce a restrictive rule, and see that as a result users get more limited access.

6.1.4 Rules changes

Problem: Rules for access may change because of new regulations, hiring and firing, and changing patient consent.

Hypothesis: Global changes to the RBAC rules will control individual users' access.

Why: In a dynamic environment with many changing variables influencing access, the model must facilitate change.

Testability: Introduce a restrictive rule, and see that as a result users get more limited access.

6.1.5 Process and time

Problem: Adding a process dimension could complicate the model, but it's necessary to give different access and ranking depending on the care process.

Hypothesis: Simply defining states in the process as a role is a valid way of giving process-dependent access and ranking.

Why: Users have different information needs in different parts of the care process. If we were to define the information needs of each actor in every process state, the model would become more complicated.

Testability: For role being the same except for its time component, a different information ranking should be shown.

6.2 Requirements for role composition

To better scope the framework we are working in, it is useful to identify some requirements from RBAC and the EHR-standard. These requirements are not to be seen as requirements for the role-model itself, neither are they to be compared to system requirements which are prioritized and measurable. Rather the requirements are to be seen as constraints on the surroundings and on what the model has to represent.

6.2.1 RBAC

The RBAC requirements are presented for two reasons. Firstly they narrow the scope of the framework and secondly they emphasize important aspects of RBAC that need to be addressed. A further description of the requirements are found in table 6.1, 6.2, 6.3, 6.4 and 6.5.

Name	Core elements
Description	Using the NIST-standard as a starting point, we have included the core RBAC elements in order to comply with RBAC.[FSG+01]. These are users, roles, operations, objects and permissions.
Goal	Our model will be compatible with a known RBAC standard.

Table 6.1: RBAC1: Core RBAC

Name	role-hierarchy
Description	The healthcare sector has multiple roles and many of them have similarities, this can be used for constructing a hierarchy.
Goal	Privilege inheritance.

Table 6.2: RBAC2: role-hierarchy

Name	Dynamic separation of duty
Description	The healthcare sector is a very dynamic environment and constraints on roles can be put in the context of the situation. Dynamic separation of duty is therefore a needed property.
Goal	Setting constraints dynamically and avoid breaking them.

Table 6.3: RBAC3: Dynamic separation of duty

Name	Least privilege
Description	The principle of least privilege is supported in RBAC and is one of the reasons for choosing RBAC in healthcare.
Goal	Our model should follow the principle of least privilege.

Table 6.4: RBAC4: Least privilege

Name	Many to many relations
Description	Between user and role, and between permission and role there needs to be many-to- many relationships.
Goal	A role is built up by other many roles and a role can have many permissions.

Table 6.5: RBAC5: Many to many relations

6.2.2 EHR

The EHR requirements are presented for the reason of making the model comply with the Norwegian laws and regulations that the EHR standard is based on. A further description of the requirements are found in table 6.6, 6.7, 6.8 and 6.9.

Name	Emergency access
Description	The opportunity of emergency access must be available in healthcare. If there are conflicting roles, but the emergency ”flag” are set the role conflict have to be neglected.
Goal	Constraints can be broken due to emergency, and the information shown will be adequate if the information exists.

Table 6.6: EHR1: Emergency access

Name	Type of authorization
Description	The type of access have to vary depending on the role accessing the record.
Goal	The model needs to differentiate the type of authorization given, i.e. read, write etc.

Table 6.7: EHR2: Type of authorization

Name	Performing actions on behalf of someone
Description	Various tasks in healthcare can be delegate and these tasks are therefore carried out on behalf of someone.
Goal	The model must give the opportunity for performing task on behalf of another user, and authorizing another user to perform tasks on one’s behalf.

Table 6.8: EHR3: Performing actions on behalf of someone

Name	Patient approval
Description	The patient may deny access to the whole or part of the record for one or several healthcare employees, by identity, profession, or other criteria.
Goal	If the patient deny access to someone this will overrule everything else.

Table 6.9: EHR4: Patient approval

6.3 Roles

We will distinguish between two types of roles, these are construction roles and functional roles.

6.3.1 Construction roles

Construction roles are the building blocks for a functional role. Figure 6.1 shows how the construction roles are built up in a hierarchy. There are several root nodes called super roles. Each of the super roles can have several children, if a node does not have any children we call it a fundamental role. Figure 6.1 shows two super roles with its children. In the hierarchy the children inherit the properties of their parent, i.e. access rights and information ranking.

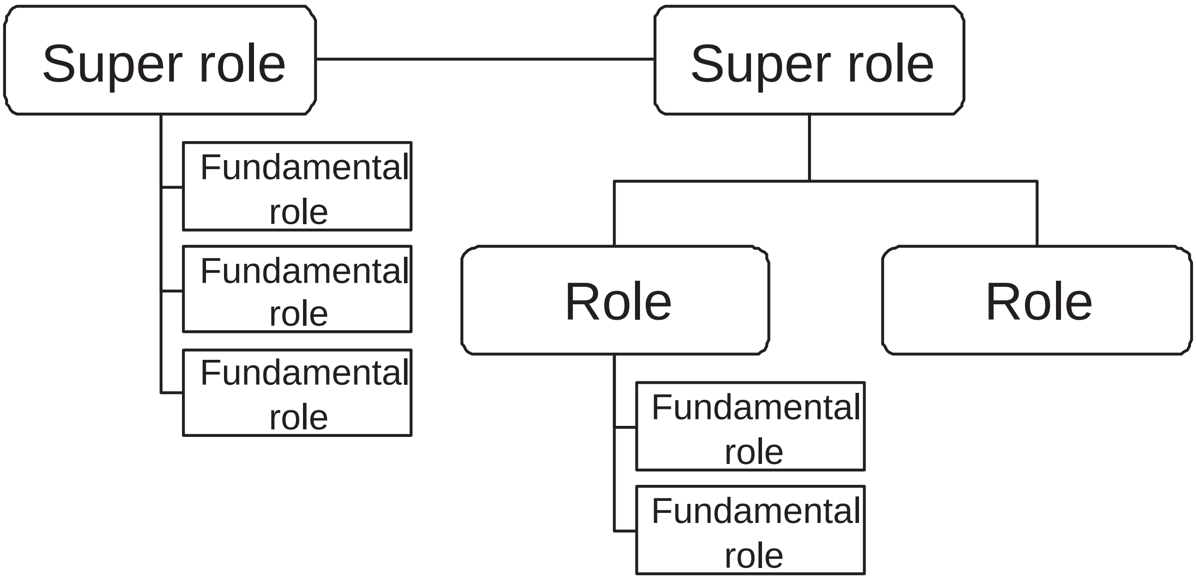


Figure 6.1: role-hierarchy for construction roles

6.3.2 Functional roles

The functional role is the role that the user acquires in a session. The functional role contains the permissions and information ranking that the user has available when a session is established. The functional role is built up by one or several construction roles. Figure 6.2 shows an example of how a functional role is built.

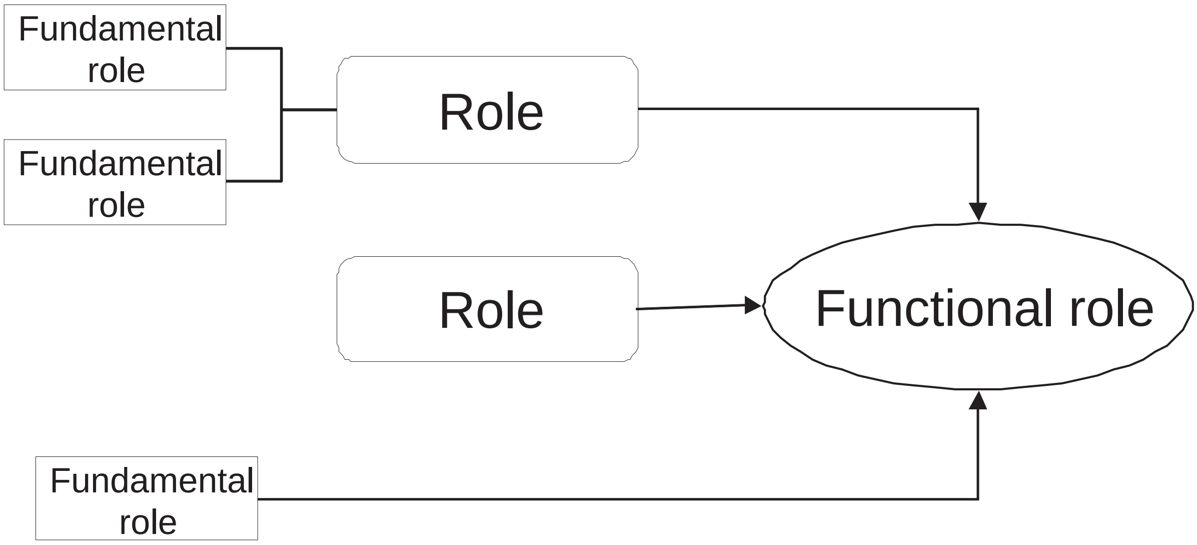


Figure 6.2: Information classification

6.3.3 Limitations of the role set

To present a complete set of roles would take a considerable amount time and the need for in depth interviews with employees at different medical facilities. Therefore we will present a set of roles we believe is sufficient to describe and test our hypotheses.

6.4 Information classification

By classification of information we mean that information is classified into groups, or classes, for instance all information regarding blood samples can qualify as one class. This concept of class is not the same as MAC classification level (see section 4.1). We believe that classifying information into classes will be a way of providing information ranking based on roles. Our general idea of information classification is pictured in figure 6.3. All information is is related to a patient, a specific time, and an incident of contact. This information can be divided into information classes. These information classes can be further divided into several information classes. This means that information classes can be both parent and child nodes. In principle the child inherits the rating from its parent node, but if the child node is given some other ranking this ranking will be applied. This reduces the number of information classes to be ranked for each role. We emphasize that an information group is not information objects, but information objects can be classified as belonging to a particular information group and are ranked thereafter.

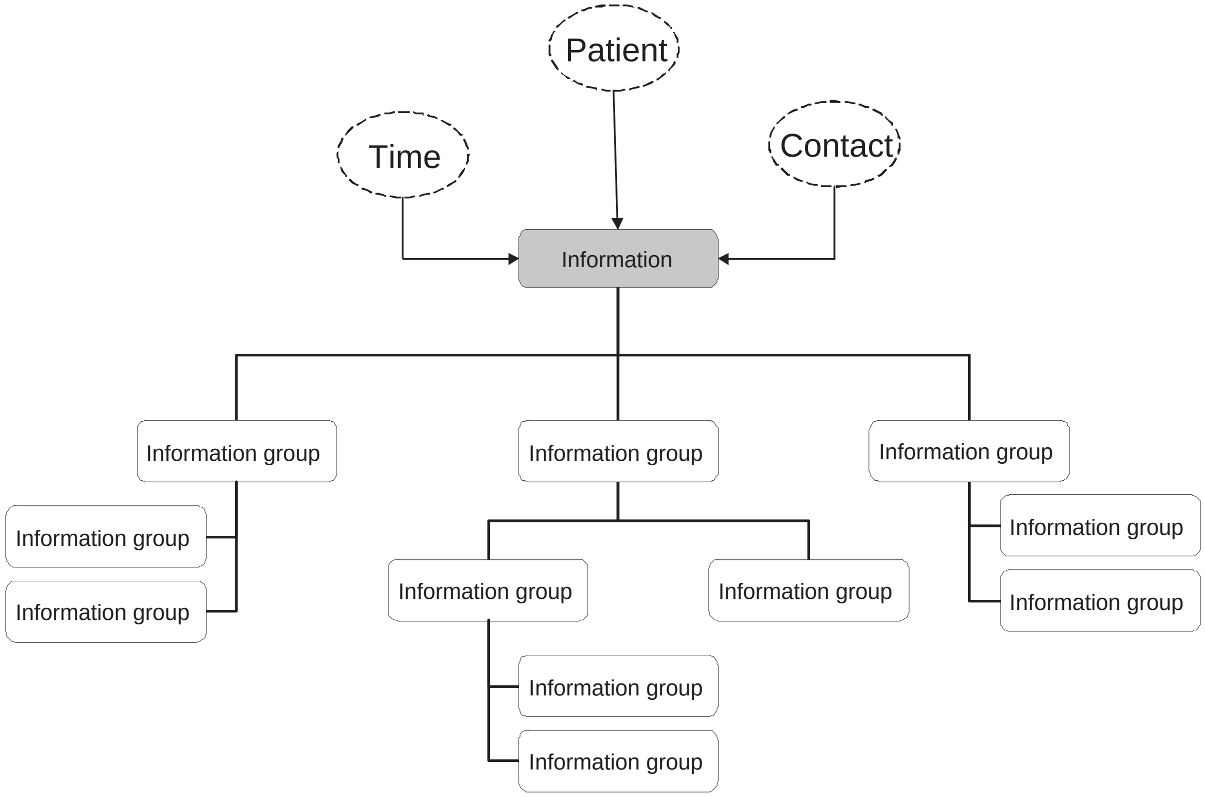


Figure 6.3: Information classification

6.4.1 Limitations

To make use of information classification we define a hierarchy of some information classes. These classes will be designed to be adequate for our case, and thus contains far from every information group needed in everyday healthcare. A likely necessity for information classification is a structured, problem based, medical record; but we think that particular discussion is outside of our scope.

6.5 Ranking of information

Information ranking is supposed to tell us something about the importance the information holds for a user. But what is important for one healthcare employee may not be important to another healthcare employee, thus we need to rank importance of particular information somehow. We will do this by ranking information groups for every construction role. This will give the functional role activated by the user information ranking based on which construction roles it is composed of. We will separate between two different types of importance. The reason for this is to tell how relevant the information is to the role and also tell at which level of detail the role needs the information. This means ranking two different parameters; relevance and detail, with a integer value.

6.5.1 Relevance

The relevance axis will tell us both how crucial the information is for the care provider and also what information the care provider is most likely to be needed to perform the best possible care. The higher the value of the integer signify that it is the more likely that the role needs to see it.

6.5.2 Detail

The detail axis will tell us on what level of detail a role needs to have sufficient information and perform the best possible care. By doing this we hope to reduce the amount of information while still retaining important information. A higher integer value signifies that the information gets more detailed. An example of this will be the detail level of the test result from a blood sample. One care provider will perhaps just need to know that the result is in the range of normal values, but another care provider will need to see the exact test result for every test done from the blood sample.

6.5.3 Relevance and detail for a role

Figure 6.4 shows how information groups can be ranked for a role. The figure 6.4 shows three information groups rated for one role and its subordinates if it has any. In this example the information group C has the most important information and the detailed information in this group is also in particular interest for this role.

6.5.4 Limitations

We have not studied what information would be important to different roles so the relevance is based on qualified guessing. The ranking of detail is also made on assumption, however we will provide cases where some information will be ″not existing″ for a particular role.

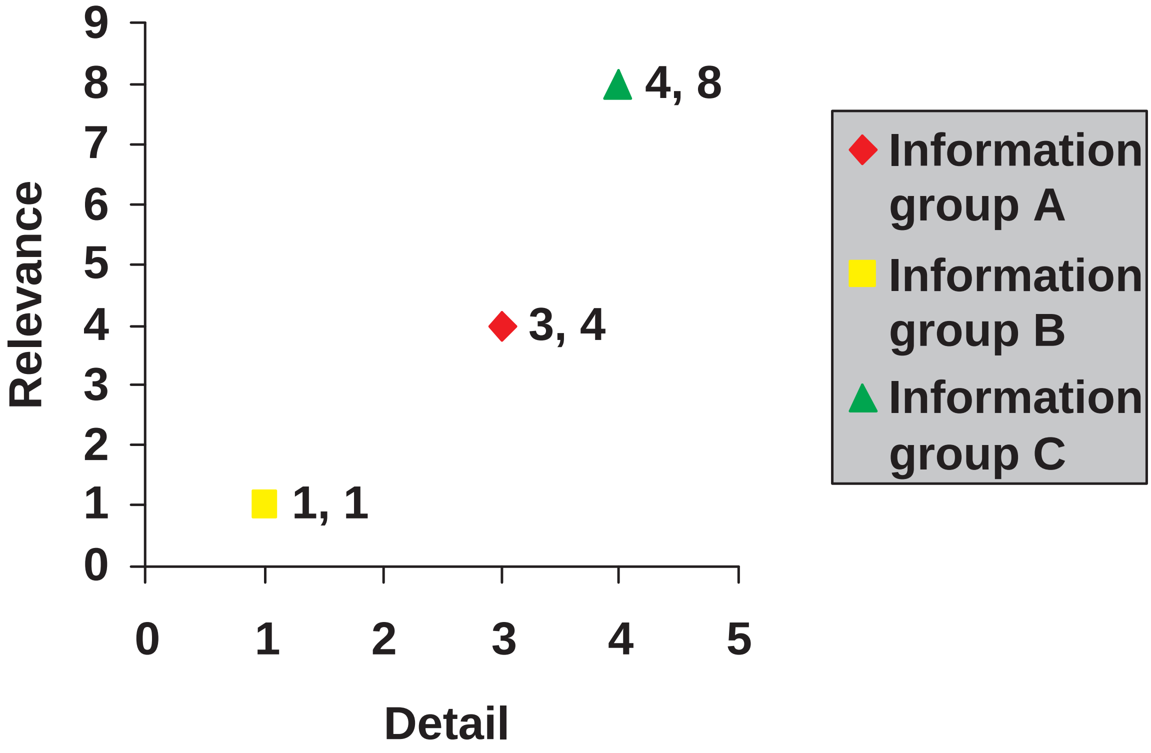


Figure 6.4: Relevance and detail for a role

Chapter 7

Cases

This chapter describes two cases that we will use to test our model. A central aspect in the two cases is medication regimens from discharges. The reasons for this is that the application of medication regimens often involve many different people (actors) with different roles in the medication process, with different goals and information needs. We will describe what information the different actors need according to our own beliefs and not from interviews of healthcare personnel. These actors are specific persons for these cases, and do not equal roles. To reflect this, we have given them names. The most important medical terms in the case descriptions will be explained in the Glossary on page 83. Drug information is based on information in a Norwegian medication catalogue [Tør04], but does not follow it totally. The central points are what information is most important to whom, an what information is to be kept confidential. After giving the case description we will make examples for use from the tree concepts explained in chapter 6, these examples will not be complete and may very well be irrelevant to a healthcare employee, but they will be sufficient to demonstrate the general idea of ranking and access control.

7.1 Case 1: Insomnia and birth control with acute bacterial infection

We start off with a case that is not very complex, but which does include some information that needs to be emphasized.

7.1.1 Story

This patient is a 28 year old female. The patient uses regular medication, drugs A and B. The patient buys these drugs for herself in a drugstore. The patient gets an acute infection in the weekend and seeks out the doctor on call for help. It is outside the office hours of the patient's regular GP. The doctor on call needs to be informed about a possible interaction between the drug that the doctor on call wishes to prescribe, and of the drugs that the patient is using. The GP needs to know about the prescription, but it is not urgent.

7.1.2 Medicines and interactions

The drugs involved in this case are listed in table 7.1.

Drug	Description
Drug A	Sleep inducer, 20 mg, taken when needed but never more than 1 tablet per day.
Drug B	Anti-conception pills, 100 μg of substance B.1 and 20 μg of substance B.2, starting on day 1 of the menstrual cycle and continued for 28 days. (The last 7 days of each package are inactive).
Drug C	The antibiotic that the doctor on call has as the first choice. Interacts with drug B with severity class B (both drugs may be taken provided certain precautions are taken.)

Table 7.1: Medication in case 1

7.1.3 Information in case

The specific information elements are described in ”C.1 Information in case 1” on page 85.

7.1.4 Actors

The minimum information needs and access denials of the actors in this case follows. Figure 7.1 illustrates access needs for a subset of the information, namely medication information. Thicker lines indicate higher importance. Continuos lines indicate more detail, while broken lines indicate less detail. Look to the list below for other information classes.

Regular GP (Faith): Has access to information about medication and diagnoses, but does not need detailed information of medications on first glance. New information entered by other healthcare workers could have a higher priority when first seen.
Minimum: List of the existence of all drugs and diagnoses.

Patient (Jane): Needs to know about the interaction between drug B and drug C. In a setting where the patient doesn’t have direct access to the patient record, this information must be relayed by one of the actors with access.

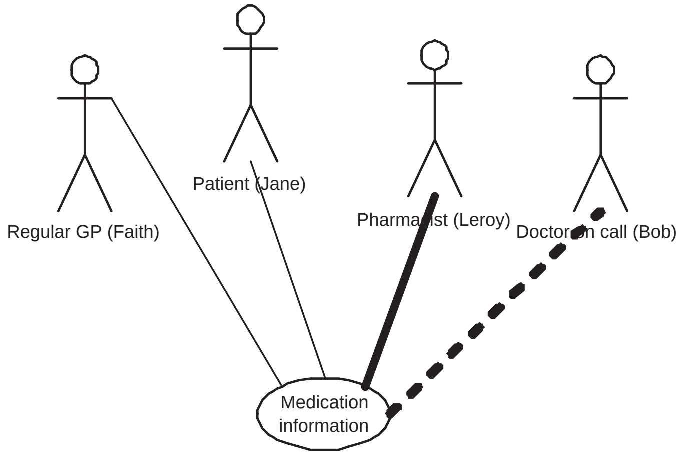


Figure 7.1: Case 1

Pharmacist (Leroy): Minimum: Needs full information about the medications, to check for any interactions, pick out the drugs, and advising the patient.
Deny: All other information except name, address, number.

Doctor on call (Bob): Minimum: Needs to know about the interaction between drug B and drug C.
Deny: All unrelated information except name, address, number.

7.2 Case 2: Endocarditis, diabetes mellitus and manic depression

We take a fictional discharge letter from an official example [Ree02, pp. 25–31], and expand the case with a psychiatric diagnosis chosen by ourselves. Combining psychiatric (mental) and somatic (bodily) information, introduces a conflict between hiding psychiatric information from those who do not need it and ensuring that medicine interaction and other critical information is available to those who need it.

7.2.1 Story

This patient is a 60 year old male, with a history of manic depression and uncomplicated diabetes. Doing non-emergency follow-up on a heart attack (problem 1), a cardiologist giving new medication needs to know about interactions with the medication that the patient takes.

The regular GP receives information about the discharge.

After discharge, the patient returns home. A home nurse visits once a week to prepare the medication doses.

Later, the patient sees a physician at a clinic with symptoms of diabetes complications (problem 2). The patient is referred to a hospital. The internal medicine specialist at this hospital finds that the patient doesn't have diabetes complications, but has an infection caused by a dental problem. A dentist fixes the problem, and the internal medicine specialist prescribes an antibiotic.

The regular GP and the referring physician receive information about the discharge.

After discharge, the patient returns home. A home nurse visits once a week to prepare the medication doses.

After four weeks, the patient is scheduled for a check-up by the regular GP. The GP secretary books the appointment.

After six months, the patient is scheduled for a dental check-up with another dentist.

7.2.2 Medicines and interactions

The drugs involved in this case are listed in table 7.2.

Drug	Description
Drug A	A high-ceiling diuretic. 1 tablet, 20 mg, in the evening.
Drug B	An anti-thrombotic drug. 1 tablet, 160 mg, in the morning.
Drug C	An anti-biotic. 1 tablet, 1 g, twice a day for four weeks after discharge
Drug D	A urea production inhibitor. 1 tablet, 100 mg, in the morning.
Drug E	An anti-psychotic drug. 1 tablet, 25 mg, twice a day.
Drug F	A non-specific beta-blocker. 1 tablet, 50 mg, in the morning. Interacts with drug E with severity class B (both drugs may be taken provided certain precautions are taken).
Drug G	A specific beta-blocker. 1 tablet, 50 mg, in the morning. Does not interact with the other drugs.

Table 7.2: Medication in case 2

7.2.3 Other constraints

The psychiatric record is not to be available to non-psychiatric healthcare personnel, except in emergencies.

CAVE (allergies and other life-critical information): Allergic to sulfa

7.2.4 Information in case

The specific information elements are described in ″C.2 Information in case 2″ on page 86.

7.2.5 Actors

The minimum information needs and access denials of the actors in this case follows. Figure 7.2 illustrates access needs for a subset of the information, namely somatic and psychiatric medication information, and CAVE information. Thicker lines indicate higher importance. Continuos lines indicate more detail, while broken lines indicate less detail. Look to the list below for other information classes.

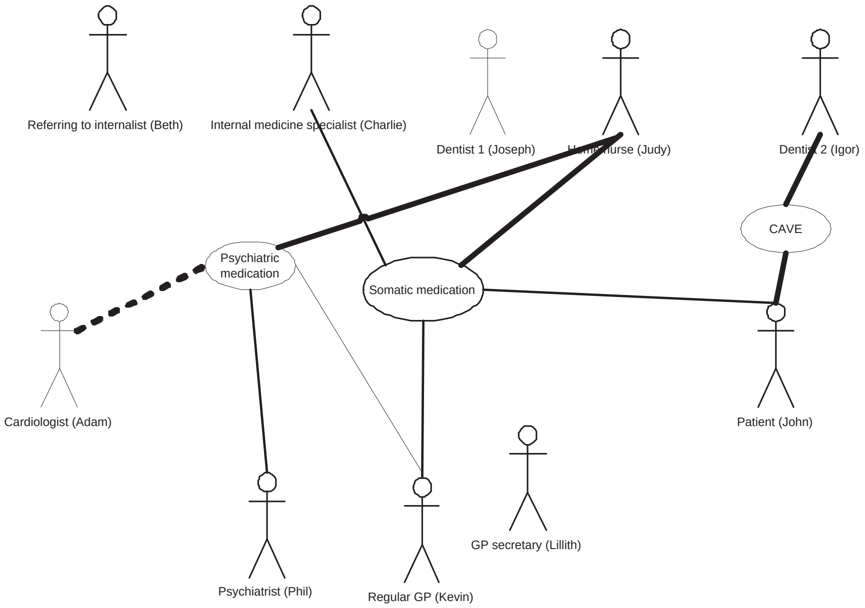


Figure 7.2: Case 2

Cardiologist (Adam): Needs access to past history, except for psychiatric conditions.
Minimum: Interaction between Drugs F and E
Deny: The reason for using drug E, psychiatric diagnosis unless the patient gives his permission.

Psychiatrist (Phil): Needs information about psychiatric problems, and needs to know about prescriptions that have interactions with psychiatric medication.
Minimum: Psychiatric diagnoses. Drug E.

Physician referring problem 2 to the internal medicine specialist (Beth): Minimum: Diagnosis found by the specialist to compare with her own diagnosis.
Deny: Psychiatric information.

Discharging internal medicine specialist (Charlie): Needs to be able to write all the relevant information in the discharge.

First dentist (Joseph): Minimum: Infection diagnosis.
Deny: All other information except name, address, number.

Home nurse (Judy): Minimum: Full dosages and usage information for all medication.
Deny: All other information except name, address, number.

Regular GP (Kevin): The patient has given his regular GP full access to all information.
Minimum at discharge: Diagnosis. What medications (particularly new ones).
Minimum at check-up: Diagnosis. What medications. Previous test results to compare.

GP secretary (Lillith): Minimum: Name, address social security number, reason for booking
Deny: All other information

Second dentist (Igor): Minimum: Need for pre-emptive antibiotics.
Deny: All other information except name, address, number.

Patient (John): Minimum: Interactions. Medication.

7.3 Representing the case with our concepts

In chapter 6 we introduce three concepts we want to use. These were information groups, role-hierarchy and information ranking. Using our case descriptions we come up with an information-hierarchy shown in figure 7.3 and a role-hierarchy shown in figure 7.4. The suggestions made here are just made for use in our case and is not meant to be complete nor is it necessarily the right structure.

7.3.1 Information groups

We have come up with the information groups shown in figure 7.3 based on the information found in the case descriptions. The information ranking will be done for each information group for each role. However if nothing else is stated a default value will be used. In addition will every child node inherit its rating from its parent unless nothing else is stated.

7.3.2 Role-hierarchy

Figure 7.4 shows the role-hierarchy we have made. The role-hierarchy consists of three different super roles with different number of construction roles. The roles that par-

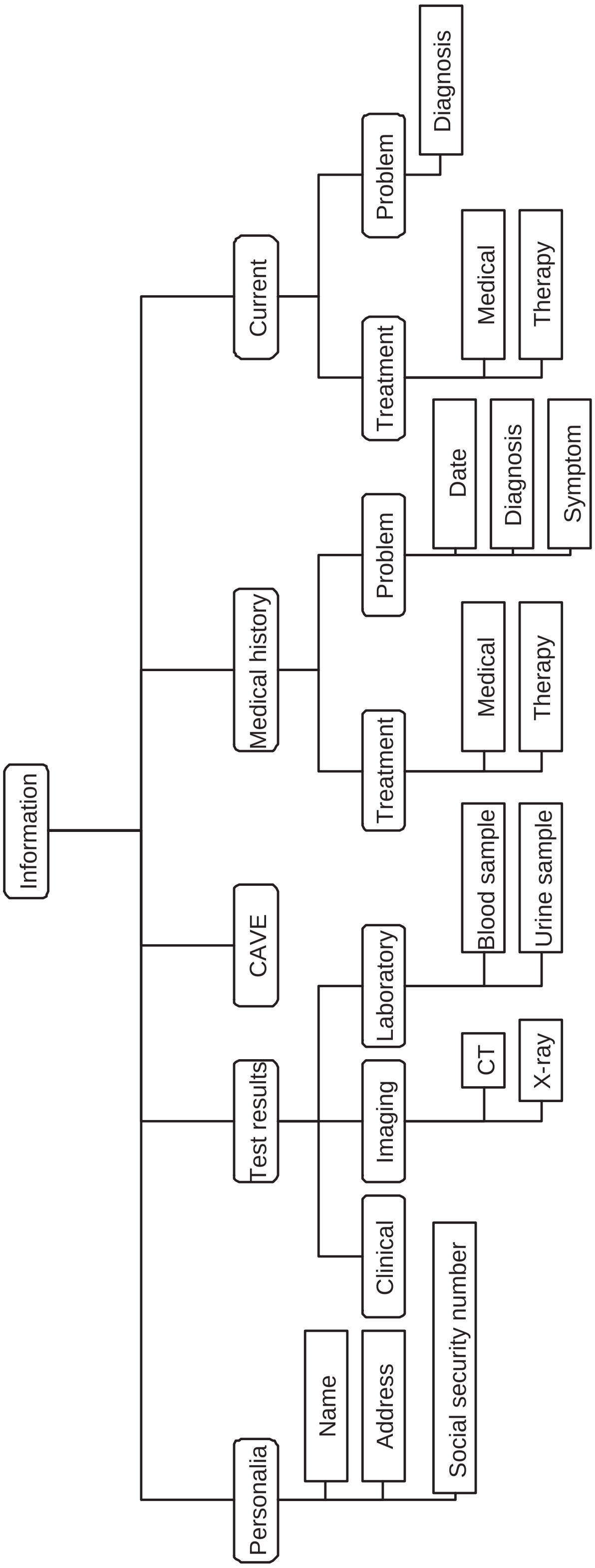


Figure 7.3: Information groups used in the case

ticipate in our case are all included in this hierarchy. The idea of having a hierarchy of roles is inheritance. The role will inherit the rating of its parent if nothing else is stated.

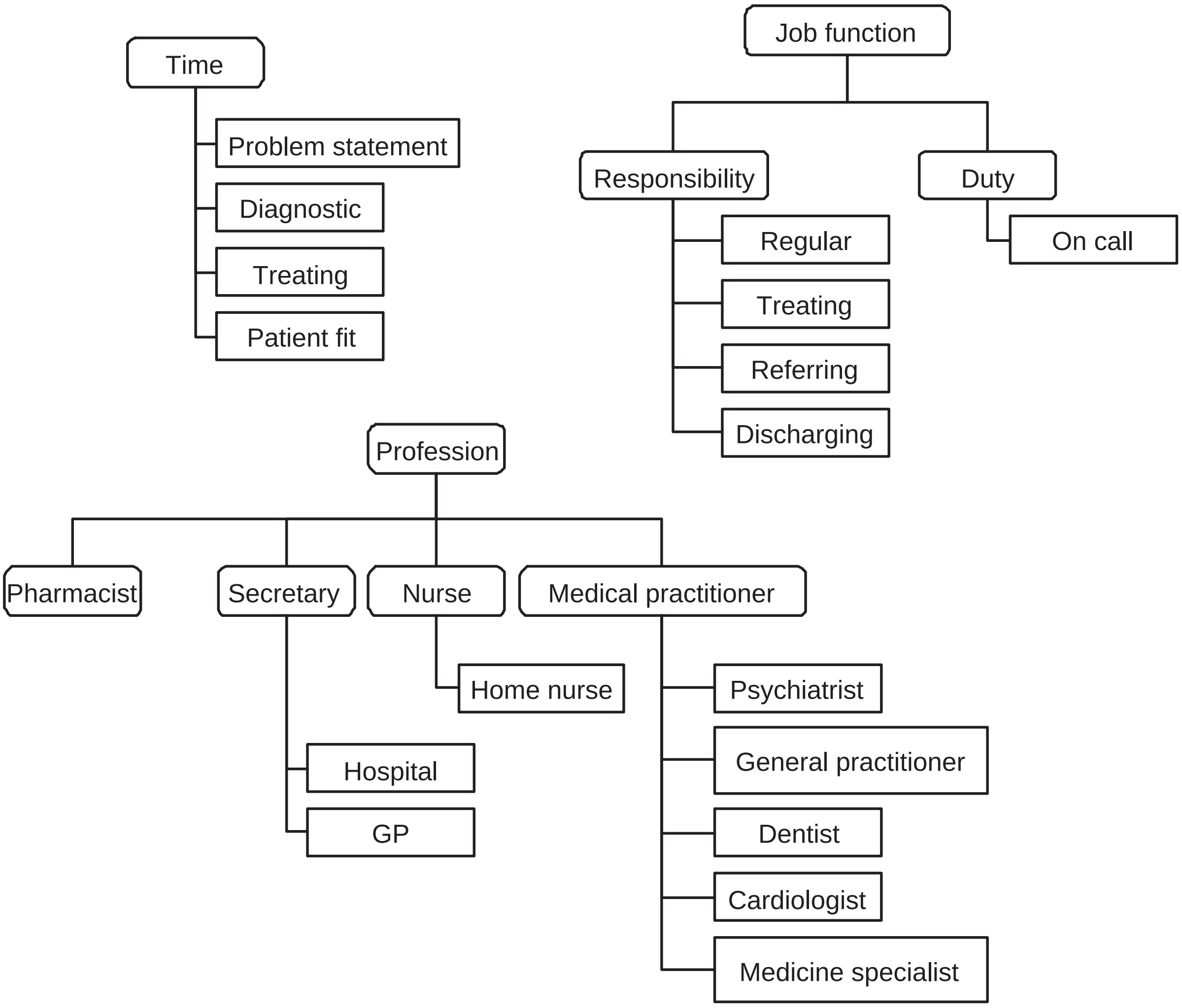


Figure 7.4: Role-hierarchy used in our assignment

7.3.3 Rating relevance and detail

For use in the case we have ranked the information groups for every role. The detailed information can be found in appendix D.

Chapter 8

Design

In this chapter we present our role-model. We start out by framing the model in a framework based on the work done in the two previous chapters. Next we discuss design decisions for combining RBAC and information ranking and how to accomplish individual patient requests. The motivation for discussing these questions is to demonstrate that there are several solutions to the problem. We then show a the beginnings of a formal definition of our model. A short description of additive methods ends this chapter.

8.1 Framework

The previous chapters have motivated creating a model, or framework, to illustrate how the concepts and ideas we are working with are related. Our information ranking and access model is an amalgam of six main models, each looking at different aspects. The six different parts are, as shown in Figure 8.1, ″Information classification″, ″role-hierarchy″, ″Access Rules″, ″Role combination″ and ″Ranking and granting″, and ″Additive methods″. While not a model on their own, access rules are an important part of the information needed to provide access. Additive methods are methods that simply add (or subtract) relevance to the results of ranking and granting.

An illustration and description of the two parts information classification and role-hierarchy is given in the two previous chapters and not given much more attention here. Again, these components as presented in this paper should not be considered authorative. They may be exchanged with other information classification and role-models, like the CareActType ontology (see 5.2, page 27), or information models from standards orgnisations. Access rules are similarly interchangeable.

The two other parts, ″Role combination″ and ″Ranking and access control″, marked with darker backround in the figure, are the central components of our model. These will be further described in this chapter.

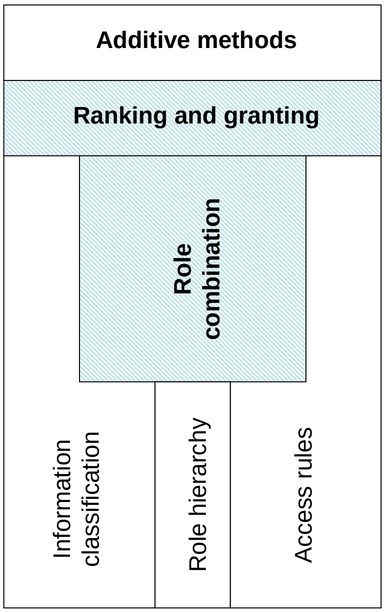


Figure 8.1: Metamodel

8.2 Ranking information and access control

Figure 8.2 and figure 8.3 shows two reference models which illustrate different approaches toward a role-model with information ranking. Both models are in reality an extension of figure 4.4. The extension consist of explicitly showing how the constructions roles are used to build a functional role and how this functional role is given privileges and information ranking. In section 6.5 we argued for the use of two different rankings in order to accomplish a satisfying information ranking. Both were needed in order to tell which information was of a particular relevance and to reduce the amount of information given while still retaining important information. Access control, however, is meant to control the access to and the type of access given to the same information.

Relating both relevance, detail and access control to the role-based model can be done with at least two different approaches. The first is to have all three parameters as independent axes, as shown in figure 8.2, with relevance, detail, and privileges assigned to the object as three different values. The second approach is to combine access control with the detail ranking, as shown in figure 8.3, by assigning operations to detail values during design or configuration.

A rationale for the second approach could be that having a low detail ranking means knowing that an information object exists, but not seeing its contents; and this is similar to having only the privilege to refer to an information object. However, placing access operations or privileges on the detail axis means that the users's ability to ”travel” along the axis and be shown more information must be limited. When reaching a certain point, the user must be stopped from showing more information. Such a limit may be bypassed by using emergency overrides, the use of which will be recorded, and investigated if there was no true emergency.

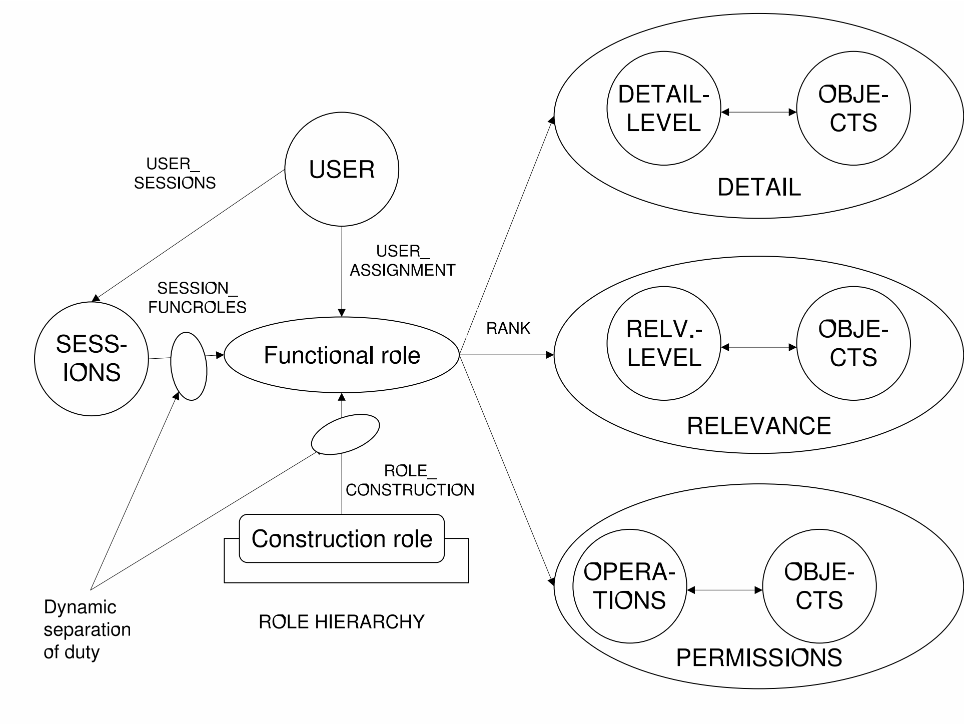


Figure 8.2: Role-based ranking and access control with three parameters

8.2.1 Placing operations on an axis

There are several ways in which operations may be assigned. We could define privileges as an ordered set, as mentioned in section 4.1. This solution needs only one operation value to be stored for each object, but has very low granularity. Figure 8.4(a) shows how a subject or role granted access to ″Move″ object a must also have access to ″Append content″ etc. By allowing each object to be given multiple operations on the access axis, as shown in figure 8.4(b), access control is more fine grained and expressive. Figure 8.5(a) illustrates an approach where each point on the access axis points to a unique privilege (a set of operations).

Figure 8.5(b) revisits combining detail and access, showing how this approach further complicates figure 8.5(b) with detail levels.

Because combining detail and operations into a single axis seems to offer too little granularity of access control, we have chosen to keep those axes separate in our model.

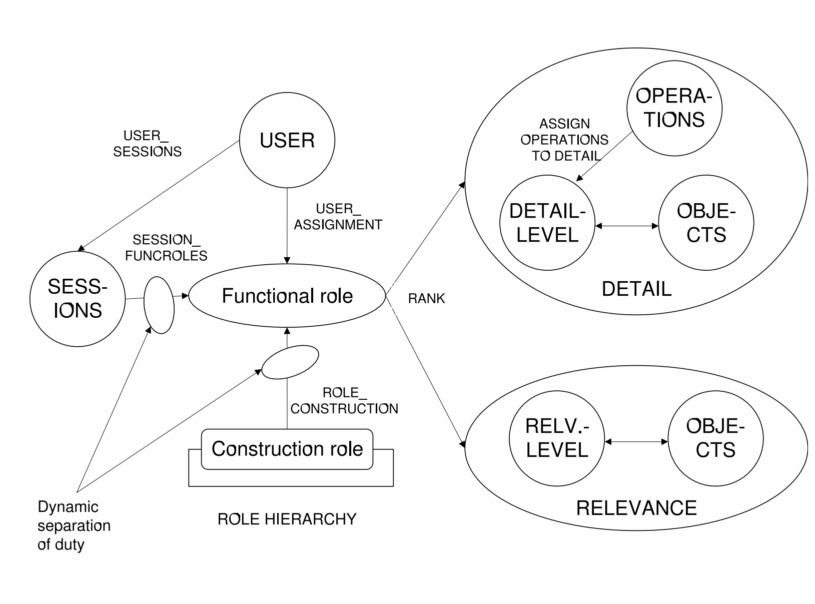
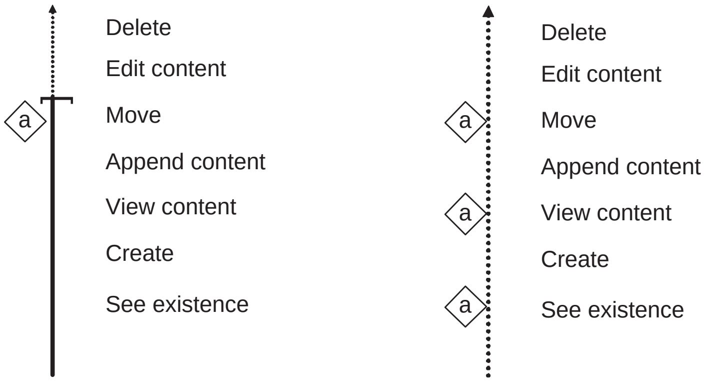


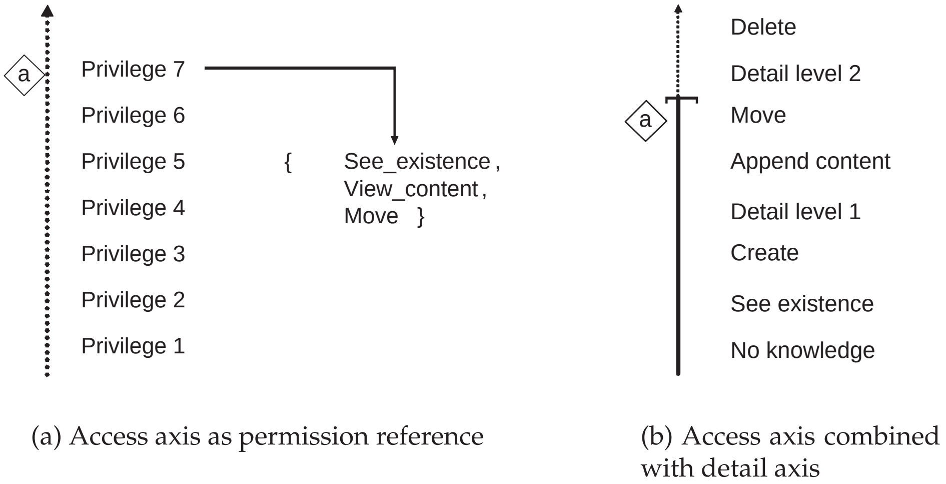
Figure 8.3: Role-based ranking and access control with two parameters



(a) Access axis as a range from zero

(b) Access axis as multiple values

Figure 8.4: Access axis I



(a) Access axis as permission reference

(b) Access axis combined with detail axis

Figure 8.5: Access axis II

8.3 Pure RBAC vs. a hybrid model

So far we have only discussed a role-based approach to access control and information ranking. We will not go into detail of other access control mechanisms, but we find it necessary to explore the possibility and the necessity of using a role-based approach in combination with an access control list (ACL) in order to fulfill all the different access rules required by the Norwegian law, one being patient consent which is stated in table 6.9. This implies that access to information is controlled both by general rules for access – the minimum amount of information to conduct a task/action – and the patient's own objectives about who should have access. In the subsequent sections we look at how general rules and patient wishes, consent and reservations, may be encoded in a purely role-based model, and in a hybrid model using both roles and patient ACLs.

8.3.1 A role-based model

We believe that RBAC is the most appropriate model for access and ranking of information in the medical record. On the other hand a problem occurs when we want to express individual patient wishes. If using a role-based model with no interference from other access control models the administrator ends up writing individual patient consent into the construction role as a part of the role definition. We believe this is feasible, but will most likely complicate the role management and at the end be inappropriate.

8.3.2 Role-based/ACL hybrid model

As mentioned in section 4.1(page 18), the basic access control models may be used in pure form or combined. An access control list is a table belonging to an information object that tells which access rights a particular user, groups or roles has to that object. Using ACL means introducing security attributes to each information object that identifies this object's ACL. The list has an entry for each system user with access privileges. This means that we can use a role-based model as the main access control and information ranking, and implement individual patient whishes as ACLs. The ACLs will in this case overrule the access control given by RBAC.

As shown in figure 8.6, the ACL is in the Ranking and granting part of our metamodel.

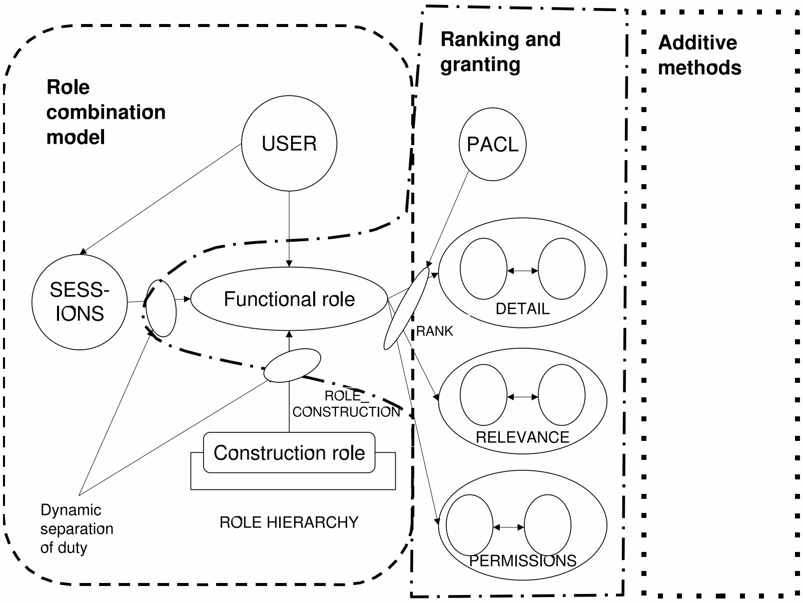


Figure 8.6: Patient-ACL in Ranking and granting

8.4 Formal model

This section contains a more formal specification of our model. We made this formal specification in order to test our hypotheses, and to evaluate our design choices. Some of the formal definition statements in this chapter are equal or similar to the formal definitions in the NIST standard [FSG$^+$01], and readers familiar with that standard will find it easier to understand our formal specification.

8.4.1 Basic definitions

First, we define some basic sets similar to the standard.

- U = a set of users, $\{u_1, \ldots, u_i\}$
- OBJ = a set of information objects, $\{obj_1, \ldots, obj_j\}$
- OP = a set of operations, $\{op_1, \ldots, op_k\}$
- REL = a set of relevance levels, $\{rel_1, \ldots, rel_l\}$
- DEL = a set of detail levels, $\{del_1, \ldots, del_m\}$
- $PE = OP \times OBJ$, the set of possible permissions
- $RE = REL \times OBJ$, the set of possible relevances
- $DE = DEL \times OBJ$, the set of possible details
- $PRIV$ = a set of privileges, $\{priv_1, \ldots, priv_n\}$, where $\forall priv \in PRIV, priv \subseteq OP$
- $ACCRANKS \subseteq REL \times DEL \times PRIV$, a set of access and rankings, each accrank consists of one privilege, one relevance level and one detail level.

8.4.2 Information classification

Each information object is a member of a an information class. The classes form a hierarchy.

- $CLASSES$ = a set of information classes for the objects, $\{class_1, \ldots, class_o\}$
- $CH \subseteq CLASSES \times CLASSES$, is a partial order on classes called the class inheritance relation, called a information classification hierarchy or class hierachy, written as $\succeq_{CH}$, where $class_1 \succeq_{CH} class_2$ means that $class_1$ is either one of the ancestors of $class_2$, or that $class_1 = class_2$
- $CA \subseteq OBJ \times CLASSES$, a many-to-one mapping object-to-class assignment relation. An object has only one class.
- $instanceof(obj \in OBJ) \rightarrow CLASSES$, the mapping of object obj onto a single class. Formally: $instanceof(obj) = \{class \in CLASSES \mid (obj, class) \in CA\}$

8.4.3 Construction roles and construction role hierarchy

The construction roles consist of an identifier and some construction rules. Like the information classes, the roles are organized in a hierarchy.

- $CRULES \subseteq ACCRANKS \times CLASSES$, is a set of rules for how an object belonging to a particular information class should be given an ACCRANK
- $const\_rule\_about(crule \in CRULES) \rightarrow CLASSES$, which class this rule concerns.
- $CORIDS$ = a set of construction role identifiers , $\{corid_1, \ldots, corid_p\}$

- $COR \subseteq CORID \times CRULES$ a set of construction roles, where $\forall cor \in COR, cor = (corid, crules), corid \in CORIDS, crules \subseteq CRULES$
- $corid(cor \in COR) \to CORIDS$, the identifier of construction role cor.
- $CORH \subseteq COR \times COR$, a partial order on COR called the construction role inheritance relation, written as,$\succeq_{CORH}$, where $cor_1 \succeq_{CORH} cor_2$ means that cor_1 is either one of the ancestors of cor_2, or that $cor_1 = cor_2$. NOTE: Permissions and ranks are inherited from parent to child. This is the opposite direction to the permission direction of inheritance in standard RBAC.

8.4.4 Functional roles

The functional roles are constructed by the role-construction function.

- $FUR(constlist, furules) \subseteq CORIDS \times FURULE$ a set of functional roles, $\{fur_1, \ldots, fur_q\}$,
- $FURULE \subseteq ACCRANKS \times CLASSES$, rules for the functional roles, built with he role-construction function
- $func\_rule\_about(furule \in FURULE) \to CLASSES$, which class this rule concerns.

8.4.5 Sessions and dynamic separation of duty

To enforce separation of duty rules, we need to use sessions and DSD rules. Compared to the NIST standard (see 4.3.1, page 20) we set n=2, meaning that DSD is broken if more than one construction role from the DSD set is activated in the same session. Also, DSD applies to both session_funcroles and role-construction.

- $SESSIONS =$ a set of sessions, $\{s_1, \ldots, s_r\}$
- $session\_furoles(s \in SESSIONS) \to 2^{FUR}$, set of functional roles activated in a session
- $DSD \subseteq COR \times COR$, Dynamic Separation of Duty is a set of construction roles that can not be part of the same functional roles or be activated in the same session. Formally:
  $\forall dsd \in 2^{COR}$,$dsd \in DSD \Rightarrow |dsd| \leq 2$, and
  - (DSD1) : $\forall fur \in FUR$,$\forall cors \in COR$,$\forall dsd \in 2^{COR}$,$\forall role\_subset \in 2^{COR}$,$dsd \in DSD$,$role\_subset \subseteq dsd$,$fur = roleconstruction(cors)$,$role\_subset \subseteq cors \Rightarrow$ $\mid role\_subset \mid < 2$, and
  - (DSD2) : $\forall s \in SESSIONS$,$\forall cors \in COR$, $\forall dsd \in 2^{COR}$, $\forall role\_subset \in 2^{COR}$, $dsd \in DSD$, $role\_subset \subseteq dsd$,
    $roleconstruction(cors) \subseteq session\_furoles(s)$,
    $role\_subset \subseteq cors \Rightarrow \mid role\_subset \mid < 2$

8.4.6 Patient access control list

The patient access control list is able to express access rules applying to both classes and objects, depending on both construction roles and users.

- $PATIENTS =$ a set of patients, $\{p_1, \ldots, p_s\}$
- $concerns(obj \in OBJ) \rightarrow PATIENTS$, maps each object to the patient whos record it belongs to
- $PACLRULE$ $\subseteq ((U \cup COR) \times (CLASSES \cup OBJ) \times ACCRANKS \times \{grant, revoke\})$, a set of patient-specific access control rules $pacrule = (subject, object, accrank, dircection)$, $subject \in U \cup COR$, $info \in CLASSES \cup OBJ$,$accrank \in ACCRANKS$, $direction \in \{grant, revoke\}$.
- $PACL \subseteq PATIENTS \times PACLRULE$, a set of patient-specific access control lists where the patients can give more access or less access, than specified globally, to users and construction roles.
- $aboutpatient(obj \in OBJ) \rightarrow PATIENTS$, a mapping of all information objects into the patient that they concern

8.4.6.1 General associations

We finish off the set and association definitions with some more associations.

- $parent(a)$, returns the direct parent (predecessor) of element a.
- $rulesof(cor \in COR) \rightarrow 2^{CRULES}$, returns the rules contained in cor.
- $rulesof(fur \in FUR) \rightarrow 2^{FURULES}$, returns the rules contained in fur.
- $ThisAndAncestors(obj \in OBJ) \rightarrow 2^{CLASSES}$, returns the set of classes where $\forall class \in CLASSES$, $class \succeq_{CH} instanceof(obj)$
- $isroot(cor \in COR) \rightarrow \{true, false\}$, is this the root of the construction role hierarchy?
- $isroot(class \in CLASSES) \rightarrow \{true, false\}$, is this the root of the class hierarchy?

8.4.7 Exceptions

We also informally define exceptions that may be triggered.

- Deny: Deny access to the user. If the information element has a relevance above Alert_level, then alert the user that there is important information hidden, who should be contacted to gain access, and wether emergency access is possible.

- Notify_owner: Notify the record-keeper, the patient or a supervisor.

8.4.8 Functions

After defining sets and associations, we here present the two main alorithmic functions used in our model. The functions are one-way, as opposed to two-way associations found in RBAC. Assigning relevance, detail, and privileges is done by relating them to construction roles, while the ranking itself is done by relating functional roles to relevance, detail and privileges.

8.4.8.1 Role construction

The role construction function (Figure 8.7, page 59) is presented in pseudo-code. This function builds the functional role from the available construction roles. It accumulates construction rules into funtional roles. A functional role should only have one rule about each information class.

As long as all the available construction rules all concern different classes, the function is deterministic. However, if there are two or more rules concerning the exact same class, Exception point 1, a policy choice will have to be made. It could be handled as an SoD-violation. It could be ignored, leading to a non-deterministic role combination. Or the functional role could be given the maximum relevance and detail of the conflicting rules, while it gets the intersection of the privileges. Formally: With $accrank_{resolved}$ being the accrank in the new functional role, $conflicting\_accranks = \{car_1, \ldots, car_t\}$, $car_x = (rel_x, del_x, privs_x)$,
$accrank_{resolved} = (max(rel_1, \ldots, rel_t), max(del_1, rel_t), \bigcap_{priv \in privs_x} priv)$

In Exception point 2, we have a standard SoD-violation that needs to be handeled according to the organization's policy.

8.4.8.2 Rank and grant

The ranking and granting function is presented in a form similar to the previous function. Figure 8.8 on page 61 contains the first part of the function, while figure 8.9 on page 62 contains the second part.

Precedence rules for the ranking and granting function:

1. PACLRULEs take precedence over FURs.
2. PACLRULEs applying to a user takes precedence over a PACLRULE applying to a FUR.
3. PACLRULEs about an OBJ takes precedence to a PACLRULE about a class.
4. Among PACLRULEs about two classes $class_1 \succeq_{CH} class_2$, the rule about $class_2$ takes precedence.

```
function $roleconstruction(cors \subseteq COR) \rightarrow FUR$
{
  if $\forall cors$,$\forall dsd \in 2^{COR}$,$\forall role\_subset \in 2^{COR}$ _
,$dsd \in DSD$,$role\_subset \subseteq dsd$, _
$role\_subset \subseteq cors, \mid role\_subset \mid < 2$

    $newfuncrole = (constlist_{new}, furules_{new})$

    $constlist_{new} = \bigcup_{c \in cors} corid(c)$
    $furules_{new} = \{\}$
    for each $cor_i \in cors$
      $a = cor_i$
      while $\neg isroot(a)$ do
        for each $crule_k \in rulesof(a)$
          $class_b = const\_rule\_about(crule_k)$
          if $\forall f \in furules_{new}, func\_role\_about(f) \neq class_b$ then
            $furules_{new} = crule_k \cup furules_{new}$
          else if the two conflicting rules come from a descendant and ancestor cor
            choose the rule from the lowest/most detailed level
            (this is satisfied by doing nothing in this step
            because we walk "up" the hierarchy)
          else
            Exception point 1: Allow / Deny / Notify_owner depending on policy.
          endif
      next $crule_k$
      $a = parent(a)$
      endwhile
    next $cor_i$
    return $newfuncrole$
  else
    Exception point 2: Deny, or Notify_owner according to policy and rules
  endif
}
```

Figure 8.7: Role construction function

5. Among PACLRULEs about the same class or object, a revokation rule takes precedence over a granting rule.
6. Among FUR_1 and FUR_2 about two classes $class_1 \succeq_{CH} class_2$, the rule about $class_2$ takes prescedence.

When setting these precedence rules, we have chosen to let a partient's access preferences about a user overrule preferences about a single object. If both preferences should be enforced with the same model, the PACLRULE could be given field for rule priority, and the patient to prioritize rules.

To enforce precedence, functional role rules are applied first, and then patient access control lists are applied.

Precondition: obj to be ranked is about a patient p that is to be provided some form of service by the user u. u has activated a set fur of functional roles, built by the role construction function, in a session.

Postcondition: While the user still has the set of functional roles activated in the session, the OBJ is assigned an ACCRANK, i.e. a relevance level, a detail level, and a set of privileges.

8.4.8.3 Directing a ranking

The funcion in figure 8.10 is used to rank an object when several rules concern the exact same class or object. Establishing wether to revoke or grant is made in the calling function, while the directaccrank function does does the actual revoking and granting of patient consent.

8.5 Additive methods

From the perspective of the user, the ranking assigns each information object a relevance level, a detail level, and gives the user a set of operations that the user is allowed to perform on the object. But our ranking does not need to be the end of ranking. As mentioned in section 8.1 on the framework, additive methods add or subtract relevance and detail after the "rank and grant" function is finished with the object. Potential additive rankings could be:

- Information that is new since the last time the user used the system, could be given an added relevance. (A trivial example of this is how new messages are given a different color when viewing an e-mail inbox.)
- Users may have different preferences for what information they need, and may assign a different relevance or detail level to some information classes. Working habits differ, and the same class of information thought too highly ranked by one user could be thought too lowly ranked by another. Permissions should of course not be set in preferences.

```
function $rank(obj \in OBJ, fur \subseteq FUR, u \in U, p \in PATIENTS, pacls \subseteq PACL)\_$
$\rightarrow ACCRANKS$
$\{$
$accrank_{result} = (0, 0, )$ (intial value)
$if aboutpatient(obj) = p$
  $class_b = instanceof(obj)$
  do
    foreach $rfurule_f \in furules_j, (corlist_j, furules_j) \in fur$
      if $class_b = func\_rule\_about(rfurr)$
        $accrank_{result} = accrank_f, where rfurule_f = (accrank_f, class_b)$
        $match = TRUE$
      endif
    next $rfurule_f$
    $class_b = parent(class_b)$
  while $(\neg isroot(class_b)) \wedge (match = FALSE)$
  $\forall rpacl \subseteq pacls, rplacl = (p, rpaclrules)$
    if $\exists paclr_m \in rpaclrules$,$paclr = (u \in U, info_m, accrank_m, direction_m)$
      if $obj \in info_m$
        if $\exists paclr_k \in rpaclrules$ where $direction_k = revoke$ and $info_k = obj$
          if $direction_m = revoke$ then
            $accrank_{result} = directaccrank(accrank_m, revoke, accrank_{result})$
          endif
        else
          if $direction_m = grant$ then
            $accrank_{result} = directaccrank(accrank_m, grant, accrank_{result})$
          endif
        endif
      else if $info \in \{info_1 \succeq_{CH} \ldots \succeq_{CH} info_n\}$and $info = info_n$
        if $\exists paclr_k \in rpaclrules$ where $direction_k = revoke$ and $info_k \in classes$
          if $direction_m = revoke$ then
            $accrank_{result} = directaccrank(accrank_m, revoke, accrank_{result})$
          endif
        else
          if $direction_m = grant$ then
            $accrank_{result} = directaccrank(accrank_m, grant, accrank_{result})$
          endif
        endif
      endif
```

Figure 8.8: Ranking and granting function I

```
    else if $\exists paclr_m \in rpaclrules, cmpaclr = (class \in COR, info, , )$
      $\forall caclr \in rpaclrules$,$caclr = (corlist, info, accrank_m, direction_m)$, -
$cor_m \in COR$ $corid(cor_m) \in corlist$, $fur = (corlist, rpaclrules)$
        if $obj \in info_m$
      if $\exists paclr_k \in rpaclrules$ where $direction_k = revoke$ and $info_k = obj$
        if $direction_m = revoke$ then
        $accrank_{result} = directaccrank(accrank_m, revoke, accrank_{result})$
        endif
      else
        if $direction_m = grant$ then
        $accrank_{result} = directaccrank(accrank_m, grant, accrank_{result})$
        endif
      endif
      else if $info \in \{info_1 \succeq_{CH} \ldots \succeq_{CH} info_n\}$ and $info = info_n$
      $accrank_{result} = directaccrank(accrank_m, direction_m, accrank_{result})$
      if $\exists paclr_k \in rpaclrules$ where $direction_k = revoke$ and $info_k = info_n$
        $if direction_m = revoke$ then
          $accrank_{result} = directaccrank(accrank_m, revoke, accrank_{result})$
        endif
      else
        if $direction_m = grant$ then
          $accrank_{result} = directaccrank(accrank_m, grant, accrank_{result})$
        endif
      endif
      next caclr
    endif
  next rpacl
endif
return $accrank_{result}$ }
```

Figure 8.9: Ranking and granting function II

```
$directaccrank(accrank_a \in ACCRANKS$,$direction_a \in \{grant, revoke\}$,_
$accrank_b \in ACCRANKS) \rightarrow ACCRANKS$
{
  $(rel_b, del_b, priv_b) = accrank_b$
  $(rel_a, del_a, priv_a) = accrank_a$
    $rel_c = max(rel_a, rel_b)$
    $del_c = max(del_a, del_b)$
  if $direction_a = grant$
    $priv_c = priv_a \cup priv_b$
  else if $direction_a = revoke$
    $priv_c = priv_b \setminus priv_a$
  endif
  return $accrank_c(rel_c, del_c, priv_c)$
}
```

Figure 8.10: Applying an accrank from a PACL

The additive preferences could be used by administrators and designers: Whenever a majority of the users agree, through the preferences, that a class of information has been wrongly ranked, role definitions could be updated. A caveat here is that a majority of the users could be wrong, and downgrading critical information, so experts and guidelines should be consulted.

- Guideline-derived rankings could be added. A knowledge base having more specific knowledge about the relationships between information, could add relevance and detail to an object. If high relevance is granted to an object without suficient privileges to see it, a ″Deny″ exception is issued (see section 8.4.7 Exceptions).

Chapter 9

Results

In the beginning of this chapter we present the results of applying our model to the cases. After the application of the cases, we investigate wether or not we met the goals set in section 6.2 Requirements for role composition. Further discussion of these results follow in the next chapter, 10 Discussion.

9.1 Results from case 1

Faith gets a functional role built up from Regular GP, Time.Treating, and Profession.MedicalPractitioner.GP. The scores for information classes in the functional role can be seen in tables 9.1. Functional roles for Leroy and Bob are represented in tables 9.2 and 9.3.

Running the case information through role construction go through Exception point 1 (see 8.4.8.1 Role construction on 58).

9.1.1 New rule in construction role

When investigating the tables, we see that CAVE-information is not shown to the pharmacist, alhough it is critical information. We adapt the case to correct this situation: The role managers now want to change the construction roles so that all users are made aware of CAVE-information.

This is the new construction rule in the construction role ”Profession”: new crule ((7,2,{F,R}), Information.CAVE).

When re-doing role construction, CAVE enters table 9.2 for the functional role of the pharmacist with relevance level 7, detail level 2 and privileges F and R. Information classified as CAVE, is now shown to the pharmacist.

Faith's first functional role
Built from construction roles: JobFunction.Responsibility.Regular Time.Treating Profession.MedicalPractitioner.GP.

Information group	**Accrank**
Information	(0,0,)
Information.Personalia.Name	(7,3,FR)
Information.TestResults	(7,4,FRUCA)
Information.CAVE	(9,2,FRUCA)
Information.MedicalHistory	(5,2,FRUCA)
Information.Current	(5,5,FRUCA)
Information.Current.Treatment.Medical	(7,4,FRUCA)
Information.Current.Treatment.Therapy	(7,4,FRUCA)

Table 9.1: 1st FUR built for Faith

Leroy's functional role
Built from construction roles: Profession.Pharmacist.

Information group	**Accrank**
Information	(0,0,-)
Information.Personalia.Name	(7,3,FR)
Information.Personalia.SSN	(5,3,FR)
Information.Current.Treatment.Medical	(7,4,FRA)
New rule: Information.CAVE	(7,2,FR)

Table 9.2: FUR built for Leroy

Bob's functional role
Built from construction roles: Profession.MedicalPractitioner JobFunction.Duty.On_call

Information group	**Accrank**
Information	(0,0,-)
Information.Personalia.Name	(7,3,FR)
Information.TestResults	(3,2,FRUCA)
Information.CAVE	(9,2,FRUCA)
Information.MedicalHistory	(7,2,FRUCA)
Information.Current	(3,2,FRUCA)
Information.Current.Treatment	(7,2,FRUA)
Information.Current.Treatment.Medical	(8,4,FRUA)
Information.Current.Problem	(7,3,FRUCA)

Table 9.3: FUR built for Bob

9.1.2 Introducing a separation of duty rule

Case 1, as defined in section 7.1, does not stipulate any separation of duty rules. The EHR standard 3.4.1.6 does stipulate a separation of duty rule, but this is a temporal SoD-rule that can't be expressed in our present model. To demonstrate that separation of duty works, we introduce another SoR-rule:

A user may never activate the construction roles pharmacist and medical practitioner at the same time.
Formally: $\{medicalpractitioner, pharmacist\} \in DSD$ and
$\{JobFunction.Responsibility.RegularGP, pharmacist\} \in DSD$

If Faith now does some extra work at the pharmacy, she should not be able to activate her role as a GP and prescribe medication from the pharmacy. The result of evaluating this DSD rule appears in table 9.4. The role combination fails because of the very first rule of the role combination function.
$|\{JobFunction.Responsibility.RegularGP, pharmacist\}| = 2$

<table>
<tr><th colspan="2">Faith's first functional role</th></tr>
<tr><td colspan="2"><b>Construction roles attempted to be activated</b>
JobFunction.Responsibility.Regular
Time.Treating
Profession.MedicalPractitioner.GP.
Profession.Pharmacist
<b>Constraints</b>
DSD = {{medical practitioner, pharmacist},{JobFunction.Responsibility.RegularGP, pharmacist}}</td></tr>
<tr><td><b>Information group</b></td><td><b>Accrank</b></td></tr>
<tr><td>Exception point 2: Deny, or Notify owner according to policy and rules</td><td></td></tr>
</table>

Table 9.4: 2nd FUR built for Faith

9.2 Results from case 2

When testing case 2, we will only concern ourselves with medication information, as we wish to concentrate on other aspects of the model. The reason for making and testing case 2 is that it is more complex, introducing patient access preferences is more dependent on time roles.

9.2.1 Time roles

Now we try to build two functional roles that are equal except for the time component. Figure 9.5, page 68, and figure 9.6. page 68 show that the accrank has changed on the class Current.

Charlie when diagnosing John	
Built from construction rules Profession.MedicineSpecialist Time.Diagnostic JobFunction.Responsobility.Treating	
Information group	**Accrank**
Information.MedicalHistory.Problem	(5,3,FRUA)
Current.Problem	(6,3,FRUCA)
Current.Treatment.Medical	(6,3,FRUCA)
Current	(5,5,FRUCA)
TestResults	(7,4,FRUCA)
Information.Personalia.Name	(7,3,FR)

Table 9.5: 1st FUR built for Charlie

Charlie when treating John	
Built from construction rules Profession.MedicineSpecialist Time.Treating JobFunction.Responsobility.Treating	
Information group	**Accrank**
Information.MedicalHistory.Problem	(5,3,FRUA)
Current.Problem	(6,3,FRUCA)
Current.Treatment.Medical	(6,3,FRUCA)
Current	(7,4,FRUCA)
TestResults	(7,4,FRUCA)
TestResults	(7,4,FRUCA)
Information.Personalia.Name	(7,3,FR)

Table 9.6: 2nd FUR built for Charlie

9.2.2 Patient preferences

If we apply the functional role in table 9.6 to ”John’s” record, the information about the current problem gets a different accrank than medical history, as shown in table 9.7. We just show diagnoses, not all the information about the problems.

The patient wants to control some of the information more closely, and the patient’s access preferences are encoded in a patient access-controll list.

PACL1={ profession, information.medicalHistory.problem.ManicDepression, (-,-,all), revoke
profession, information.medicalHistory.treatment.medication.Drug E, (-,-,all), revoke

Problem	Accrank
Non-insulin-dependent diabetes mellitus	(5,3,FRUA)
Bipolar Affective Disorder	(5,3,FRUA)
Old myocardial infarction	(5,3,FRUA)
Acute nonspecific idiopathic pericarditis	(7,4,FRUCA)

Table 9.7: Problems ranked 1

profession.nurse.HomeNurse, information.medicalHistory.treatment.medication, treatment, (-,-,all), grant
profession.medicalpractitioner.psychiatrist,
information.medicalHistory.
problem.ManicDepression, (-,-,all), grant
Kevin, information, (-,-,all), grant
}

When applying these rules, Kevin, the regular GP, should get full access to the record. The psychiatrist should get acess to the pscychiatric information, while Charles should be denied access to information about Drug E, and the related problem. This

9.3 Key findings

In this section we point out some of the more obvious results. Further discussion of these results follow in the next chapter, 10 Discussion.

Information is ranked by class depending on roles, and the hierarchies are handeled in a formally specified fashion. This confirms the hypotheses ”Ranking of information” and ”RBAC and information ranking” (subsections 6.1.1 and 6.1.2).

Constraints, like patient preferences (consent and reservations), are enforced by the model. And introducing new rules are a matter of adding a separation of duty role subset to DSD, making a new PACL, or changing the rules of the construction roles. However, temporal constraints and temporal-SoD constraints can not be expressed by the model in the current form. Hypothesis ”Role constraints” (subsection 6.1.3) is thus proved only for separation of duty and patient access preferences

The hypothesis ”Rules changes” (subsection 6.1.4) was proved by introducing a new rule.

The hypothesis ”Process and time” (subsection 6.1.5) is satisfied, as shown in 9.2.1 Time roles.

When using our role combination and ranking functions, we discovered that rules compete to control ranking and access to the same class, as predicted in the explanation of ”breakpoint 1” in 8.4.8.1 ”Role construction”. This happended several times just for the rather small case 1. The conflicts were resolved by the method of using the maximal values of relevance and detail from the conflicting rules, and making the privileges an intersection of the privileges of the conflicting rules.

Our model does not rank interacting medication differently than non-interacting medication. In light of the cases that we developed, this is a significant failure.

Chapter 10

Discussion

In this chapter we will discuss the choices that we have made during our work, what the desired effect of these choices was, and wether real effect was according to those desired effects. At the end of the chapter we will make suggestions for further work.

10.1 Choices

The primary aim for our assignment was to create a role-model which combined access control and information ranking. To accomplish this goal we needed to narrow the scope of the project. To do this we made some decisions based on what we thought was correct choices, and made some ad hoc solutions. In this chapter we will discuss the most important decisions and ad hoc solutions, and wheter those choices make our model more or less suitable for representing access control and ranking.

10.1.1 Using the proposed NIST-standard

We have used the proposed NIST-standard as a starting point for the development of the role-model, because the standard was given as part of the curriculum in one of the subjects related to this assignment, and because it is considered an authorative work on role-based access control. Using the NIST-standard we had to decide wether to use hierarchical RBAC or not and if we was going to use static or dynamic separation of duty. From the articles we have read that deals with RBAC in healthcare, [Inc02] in particular, we found it well established that using hierarchical RBAC and dynamic separation of duty was the right choice.

10.1.1.1 Hierarchical RBAC

A lot of empirical work, interviews and analysis needs to be done in order to make a complete hierarchy. This is a task we considered outside the scope of this project, but we still needed a hierarchy. Thus we defined our own hierarchy based on cases, and from this we constructed an ad hoc solution. The roles in the hierarchy was based on the roles in the case and we could therefore expect the hierarchy to be complete with reference to the case. Our role-model would be valid for the real world if we were able to make the role-model in such a way that real, professionally developed, role hierarchies and information classifications can be ”plugged into” our model without significant modification. Beyod interpreting them to see what rules they contain, the ranking model is not concerned with what the roles actually mean; therefore a substitution of the role-hierarchy should work, but this is not tested. Making the roles in the hierarchy we would expect them to retrieve the right information and the results from the test shows that this is a correct assumptions.

10.1.1.2 Dynamic separation of duty

Our role-model does use dynamic separation of duty, and we were able to test that this works. But the kind of DSD that the EHR standard requires (alerts about self-approved actions, 3.4.1.6, page 13), needs to be expressed as a function of both roles and time. Because our model has not been extended with temporal constraints, the model will not properly represent such constraints.

10.1.2 Information ranking

Developing our solution for ranking information included two main choices. First we decided on what parameters we ranked the information by and how to relate this to the role-model. Secondly we created a way of ranking the information according to roles.

10.1.2.1 Two parameters

Consulting with our mentor we decided that it could be necessary to rank importance as several parameters. The two different parameters we wanted to sort the information for were relevance and detail. These two parameters were chosen because we believe they were both complimentary and relatively independent. The outcome we hoped for from the test results was that the information would be given relevance and detail values in accordance with our case description. The test results were positive to both reducing the amount of information and prioritizing relevance, though the result is biased against the fact that information is ranked according to the case.

The principle of least privilege was only partially supported. Because we didn’t want to obscure information when ranking it, all information was still accessible for the role. A way to introduce least privilege would be making responsobility roles as fine grained as the action templates mentioned in the EHR standard (3.4, page 11).

By making the access control, relevance ranking and detail ranking independent of each other in relation to the role we had a lot easier design job. Different options of combinations were discussed in section 8.2 and is not further discussed here.

10.1.2.2 Information ranking for the different roles

Having decided how information was going to be ranked we needed to rank information for the different roles. This is a necessity in order to provide information ranking from the role-model. We wanted to use an information hierarchy based on the same assumptions we had for creating a role-hierarchy. These assumptions were that it would be easier to administrate, easy to substitute because of independence in relation with the role-model, and we would have an easier job ranking information for particular roles.

The ranking of the information was done quite easily for our ad hoc solution and the test results supports the fact that a hierarchy is sufficient to provide the desired outcome.

There are many ways in wich a role may be related to a healthcare information object, mainly

- Role related to class. An information object has a type, and roles may have different interests and needs for information depending on to what class of information the information object belongs. E.g. a pharmacist role may be interested in current medication. This relationship links a role and object by information class, and it is the type of relationship between role and information the we chose to use for ranking and access control.
- Role related to individual object. E.g. the creator of an information object may have the right to update and correct the content of the object for a specified time after creating it. This type of role is not possible to represent inside our core model
- Role related to individual patient. The regular GP has a role related patients on his/her list of regular patients, and the patient may have somebody acting on its behalf.

All three of these role type might be needed to competely define access rules. We see potential for extending our model with these role concepts.

10.1.3 Pure role-based model vs. hybrid model

The combination of default roles to rank information classes, and patient-specific access control lists worked as predicted. When using only the default roles, the patient's consent and reservations did not affect ranking.

10.1.4 Using cases

Along with law and standards, the case was useful for defining what kinds of medical information our model should work with, and what kinds of constraints, relationships and phases form the context for the use of that information. The case also was useful for testing wether or not our hypotheses held for our model.

As the model was designed, the case was represented in a form that was more formal, in preparation for use in the construction and ranking functions. The case was slightly changed for the purpose of making a better demonstration of the abilities of the model, but beyond detailing we changed the case as little as possible. Too frequent and large changes to cases could lead to a situation where the model is perfect for the case, only because the case was changed to fit the case instead of the model being truly improved. We made sure to be aware of this, and tried to avoid this trap.

It could be argued that we should have spent less effort developing and detailing the cases, and spent more effort on improving our formal model, making it more complete. To a certain extent, we agree with this argument; but defining cases and then formalizing them are important phases of understanding and validating. Without cases, our model could have been even less complete than it currently is, all the while we might be deluding ourselves that it was perfect.

10.1.5 A formal definition

The language we have developed in section 8.4 is both a formal and an expressive language suited to present a role-model. Using the formal language helped bring our attention to some design choices that might otherwise remain hidden, particularly about how to resolve conflicts between rules. When using the formal language, we found a mistake that we made when we first defined the ranking function; this mistake has now been corrected, but might not have been found had we defined the function in less formal terms. The formal language also made it possible to test the model using cases.

As mentioned in section on page 65 in section 9.1, allready the first case showed the importance of a choice brought out by the formalism, namely how one should handle two rules about the exact same class.

We belive that the use of some form of formalism is a necessary part of defining a model like ours. First, because one wants to check that one can implement a policy to comply with the requirements of the laws, regulations and the EHR-standard; and second, because the model ultimately must be represented in computer-interpretable form. It may well be that the simple set statements that we used to formalise and disambiguate the model are not an ideal form of representation. A better language, both in expressiveness, validation potential and ease of use, may be available. Temporal deontic logic [Bai95] and temporal RBAC are possible candidates.

10.1.6 Representing time

One of the design choices was how to represent time, i.e. treatment phases, in our model. We tried to solve this by making time another type of construction role. This was a conscious choice. Specifying information needs as a function of only time is a simpler method than specifying information needs for every role in every time. Although the time-role in our test case could be too simple, it will validate this solution in a limited environment. The test-results show that a time-role proves to do the work of putting the information in context of time for our test case.

10.2 Further work

We have not developed a complete role-model and there are several different parts that are subjects for further work. We will depict the future work in two phases. The first phase will be composed of two parts, one is the completion of the role-hierarchy and the information-hierarchy and the second is the completion of the language and the role-model itself. The second phase will be using the role-model in software and conducting a usability study.

10.2.1 Completion of role-hierarchy and information-hierarchy

Our focus when developing the role-model consisted mostly of making a good example design for use in ranking. This did of course effect our work on developing the role-hierarchy and information-hierarchy. First of all the hierarchies needs to be developed in cooperation with healthcare personnel, and second a more formal definition for the building of the hierarchy is needed. Even if they are independent of the role-model itself, it is important to have a complete set of information classes and roles, in order to judge if role-based access control and information ranking give the desired results for the system users. If our main focus should be developing the model and the theory, the hierarchies could be detailed for a subset of the domain.

10.2.2 Completion of the language and role-model

10.2.2.1 Roles

Further development of the model should include more role concepts than just the class-role relationship.

Our role-model has focused on the relationship formed between a role and a piece of information through the information's class. However, this is not the only way in witch a role and an object could be related. A role may be related to the information through a relationship with the patient, such as ”GP”, ”record keeper” or ”next of kin". The role may also be defined in relation to a single information object, such as "writer", "approver".

10.2.2.2 Constraints other than SoD

Temporal constraints: One of the rules in the EHR-standard, was that self-granted access used by nobody else within a certain times should trigger a notification. To the "owners" of the record. To fulfill this requirement inside the role-model, would require that the role definitions and functions would be able to handle temporal (time) constraints.

10.2.2.3 Representing time and relationship between information

Although exploring the result of using time as a construction role this theory is a subject for further study.

Another characteristic in the healthcare sector, that is not at all included in our model, is the relationship between information, i.e. information being more important because it is related to other important information.

10.2.3 System development

In order to both produce complete hierarchies and a complete language it will be practical to develop an EHR-prototype or use pre developed software which implement a role-model. This would make it possible to test and validate this approach in practice and not just in theory.

Chapter 11

Conclusion

In this project we have examined the possibility of developing a role-model that is capable of dealing with both access control and information ranking. The size of this project has not allowed for creating a complete model, but in the process of developing and testing the case we made some important discoveries.

Of our findings, the most important is the indication that using the same role-model for access control and information ranking is possible. We also discovered that realizing patients individual objectives seems to be easier by using an access control list than using a pure role model. Although only used as an extension to the test cases, developing a role-hierarchy and an information-hierarchy, and proposing a way of ranking information in the relationship with roles, we have been able to demonstrate that administration is made easier using hierarchies. We also found that information needs to ranked with several parameters.

Formally defining our model brought out issues that we otherwise may have overlooked.

The model can not express guidelines, and has a limited ability to express time. In its present state, the role model can not be legally used as the sole access control method in a record system, because pure implementations would not comply with Norwegian law or standards for access to healthcare information. However, we see the potential for making a more complete model.

This assignment have worked as a pre-study for our master thesis. We have accomplished our research goals and we feel we have come up with a strategy for further work. It still remains to see if the theory will be functional in the real world, and wether is too complex to carry out in practice.

Appendix A
References

[AS00] Gail-Joon Ahn and Ravi Sandhu. Role-based authorization constraints specification. *ACM Transactions on Information and System Security*, 3(4):207–226, November 2000.

[Bai95] Patrice Bailhache. Canonical models for temporal deontic logic. *Logique & Analyse*, 149:3–21, 1995.

[Bay02] Elisabeth Bayegan. *Knowledge Representation for Relevance Ranking of Patient-Record Contents in Primary-Care Situations*. EngD thesis. Norwegian University of Science and Technology, 2002. This thesis contains several separately published articles, [BØNG01] among them.

[BØNG01] Elisabeth Bayegan, Øystein Nytrø, and Anders Grimsmo. Ranking of information in the computerized problem-oriented patient record. In Vimla L. Patel, Ray Rogers, and Haux Reinhold, editors, *Proceedings of the 10th World Congress on Medical Informatics (MEDINFO2001)*. IOP Press, 2001.

[Cra03] Jason Crampton. Constraints in role-based access control(foil-set). http://www.isg.rhul.ac.uk/ jason, June 2003.

[FCK95] David. Ferraiolo, Janet. Cugini, and Richard. Kuhn. Role-based access control. http://hissa.ncsl.nist.gov, November 1995.

[FKC03] David F Ferraiolo, Richard Kuhn, and Ramaswamy Chandramouli. *Role-based Access Control*. Artech House Publishers, April 2003.

[Fra03] Steve Frame. Role-based access control. http://www.giac.org, November 2003.

[FSG$^+$01] David F. Ferraiolo, Ravi. Sandhu, Serban. Gavrila, Richard. Kuhn, and Ramaswamy. Chandramouli. Proposed nist standard for role-based access control. http://www.nist.org, August 2001.

[Gol99] Dieter Gollmann. *Computer Security*. Worldwide series in computer science. Wiley, Chichester, 1999.

[hA04] Norsk helsenett AS. Norsk helsenett as. http://www.norsk-helsenett.no/, 2004.

APPENDIX A
REFERENCES

[Hd02] Helsedepartementet. Lov om elektronisk signering. http://www.lovdata.no, December 2002.

[Hd03a] Helsedepartementet. Forskrift om pasientjournal. http://www.lovdata.no, February 2003.

[Hd03b] Helsedepartementet. Helsepersonelloven. http://www.lovdata.no, August 2003.

[Hd03c] Helsedepartementet. Helseregisterloven. http://www.lovdata.no, August 2003.

[Hd03d] Helsedepartementet. Lov om psykisk helsevern. http://www.lovdata.no, August 2003.

[Hd03e] Helsedepartementet. Pasientrettighetsloven. http://www.lovdata.no, December 2003.

[Hd03f] Helsedepartementet. Spesialisthelsetjenesteloven. http://www.lovdata.no, December 2003.

[Hd04a] Helsedepartementet. Helsedepartementet. http://odin.dep.no/hod/, October 2004.

[Hd04b] Helsedepartementet. Helseforetaksloven. http://www.lovdata.no, May 2004.

[Inc02] RSA Security Inc. Role-based access control in healthcare. http://www.rsasecurity.com/, 2002.

[Kie99] David Kieras. A guide to goms model usability evaluation using gomsl and glean3, 1999.

[KP00] Manolis Koubarakis and Dimitris Plexousakis. A formal model for business process modeling and design. In *Conference on Advanced Information Systems Engineering*, pages 142–156, 2000.

[KTM03] Anthony Karageorgos, Simon Thompson, and Nikolay Mehandjiev. Specifying reuse concerns in agent system design using a role algebra. In R. Kowalczyk, H. Tianfield J. Müller, and R. Unland, editors, *Agent Technologies, Infrastructures, Tools, and Applications for e-Services*, number 2592 in Lecture Notes in Artificial Intelligence. Springer, 2003.

[Nys01] Torbjørn Nystadnes. Elektronisk pasientjournal standard. http://www.kith.no/epj_undermapper/19547/, June 2001.

[oh01] Sosial og helsedepartementet. Si@ elektronisk samhandling i helse- og sosialsektoren. http://www.shdir.no/, 2001.

[okd01] Kultur og kirke departementet. Arkivloven. http://www.lovdata.no, May 2001.

[opd00] Justis og politi departementet. Personopplysningsloven. http://www.lovdata.no, April 2000.

[PHS03] Josef Pieprzyk, Thomas Hardjono, and Jennifer Seberry. *Fundamentals of computer security*. Springer, Berlin, 2003.

[RC99] Anthony. Rhodes and William. Caelli. A review paper role based access control. http://www.isrc.qut.edu.au, 1999.

[Ree02] Arnt Ole Ree. Medisinsk-faglig innhold i epikriser - 'den gode epikrise'. http://www.kith.no/Epikrise/Den_gode_epikrise.pdf, December 2002. ISBN: 82-7846-159-7.

[Tør04] Holger Moe Tørisen, editor. *Felleskatalog*. Felleskatalogen AS, 2004.

APPENDIX A
REFERENCES

Appendix B

Glossary

ACL Access Control List: Here a patient access preferences

Cardiology: A ranch of medicine studying the heart.

CAVE: a caveat, a warning, from ″cavere″ (Latin).

CPR: Computerized Patient Record

Construction role: A role that preforms as a building block for a functional role

DAC: Discretionary Access Control

Diabetes mellitus: Medical condition where the body is unable to control the level of sugar in the blood.

DSD: Dynamic Separation of Duty

Electronic patient record/Electronic health record (EPR/EHR): A set of information about patients and their treatment

Endocarditis: Inflammation of the endocardium

Epicrisis: Discharge letter

Functional role: The active role a user requires in a session.

GP: General Practitioner

HCO: Healthcare organization

Health care authority: An enterprise in the healthcare sector.

ICD-10: International Classification of Diseases A systematic classification of all diseases. Used primarily in hospitals.

Insomnia: Difficulty to sleep

MAC: Mandatory Access Control

Manic depression: A medical condition where the person is sometimes very excited and sometimes very depressed and is unable to control these feelings.

RBAC: Role-Based Access Control

SSD: Static Separation of Duty

Appendix C

Detailed case information

This appendix contains the detailed information stored in the records. It is presented in an indented tree structure, because that reflects how information elements are organized into classes.

C.1 Information in case 1

- Personal details
  - Name: Jane Sleepy
  - Social security number: 01017400262
  - Address: 34 Stormy heights
- CAVE (allergies and other life-critical information): None
- Medical history
  - ICD-10: G47.0 Insomnia
    - Time: 2 years ago to now
    - Treatment: Medication: Drug A: Sleep inducer, 20 mg, taken when needed but never more than 1 tablet per day.
  - Contraception
    - Medication: Drug B: Anti-conception pills, 100 μg of substance B.1 and 20 μg of substance B.2, starting on day 1 of the menstrual cycle and continued for 28 days. (The last 7 days of each package are inactive).
- Current problem
  - Symptoms
    - Pain when urinating

  - Test results: Biochemical: Urine: ph 8,5 , Protein high, Nitrite high. Indicates bacetriuria.
  - Diagnosis
    - ICD-10: N30.9 Cystitis (infection)
  - Treatment
    - Medication: Drug C: The antibiotic that the doctor on call has as the first choice. Interacts with drug B with severity class B (both drugs may be taken provided certain precautions are taken.)

C.2 Information in case 2

- Personal details
  - Name: John Sick
  - Social security number: 01014400369
  - Address: 54 Lotsofham Forrest
- CAVE (allergies and other life-critical information): Allergic to sulfa
- Medical history
  - ICD-10: E11.9 Non-insulin-dependent diabetes mellitus, Without complications
    - Time: 10 years ago to now
    - Treatment: Medication: Drug A: A high-ceiling diuretic. 1 tablet, 20 mg, in the evening. Drug D: A urea production inhibitor. 1 tablet, 100 mg, in the morning.
  - ICD-10: F31 Bipolar Affective Disorder
    - Time: 10 years ago to now
    - Treatment: Medication: Drug E: An anti-psychotic drug. 1 tablet, 25 mg, twice a day.
- Problem 1 (This is a current problem for the discharging cardiologist, but past history for the other actors)
  - Symptoms: None. Follow-up after infarction. Adjusting medication.
  - Diagnosis
    - ICD-10: I25.0 Old myocardial infarction.
  - Treatment

    - ∗ Medication: Drug B: An anti-trombotic drug. 1 tablet, 160 mg, in the morning. Drug G: A specific beta-blocker. 1 tablet, 50 mg, in the morning. Chosen after finding that drug F interacts with drug E.

- Problem 2 (This is a current problem for the discharging internal medicine specialist, but past history for the dentist)
  - Symptoms
    - ∗ Feels sick and listless
    - ∗ High body temperature
  - Test results
    - ∗ Biochemical: Blood glucose: 7.4-7.9-7.7-7.6-6.1-7.0 Leukocytes: 8.2-6.1-6.7-6.9-8.4 SR: 49-61-19 CRP: 79-52-39-30-20-4-4-6-10 Creatinine: 85-79-76-77-77-92 Urine acidity: 468-304-340
    - ∗ Microbiological: Urine: Mixed culture, probably polluted. Blood: Streptococcus mutans, sensitive to penicillin
    - ∗ Imaging and measurements: Electrocardigram: Sinusrythm, 64 bpm. Computer tomography: Normal liver and spleen. Kidney cysts. Ureteres of normal width.
  - Diagnosis
    - ∗ ICD-10: I33.0 Acute nonspecific idiopathic pericarditis
  - Treatment
    - ∗ Other treatment: Given dental care
    - ∗ Medication: Drug C : An anti-biotic. 1 tablet, 1 g, twice a day for four weeks after discharge
    - ∗ Other treatment: Check-up after 4 weeks: BT, CRP and urine.
  - CAVE
    - ∗ New entry: Needs pre-emtive antibiotics (endocarditis profylaksis) before dental work, surgical intervention or use of instruments through the mouth, the colon or the urethra.

Appendix D

Rating

All roles will be given a rating for every information group. The super role are given its rating and this will be given a rating for the needed information groups, or classes, and this rating will be its children's defaults. This means that when the profession-role is given 7(relevance) and 3(detail) for Name and default is 0,0, every sub-role will have Name rated as 7,3 and the rest 0,0. However the default value may be changed for a role, and then this is the current ranking value. An example of this is the dentist role which inherits its ranking from the medical practitioner. For the dentist, the default value is set to no access, meaning the dentist does not have access to any information except for the information classes specified in the table. There is one table for each super role. Table D.1 rates the profession-role-tree, table D.2 rates the time-role-tree and table D.3 rates the function-role-tree.

In addition, we have given each information group. We have defined a set of privileges for our case. They are F: ″reFer to″, R: ″Read″, U: ″Update″, C:″Create″, and A; ″Append″.

The coloumn headings mean: R:relevace values, D: detail values, P: privileges.

Each rating number is given a discription, and this can be found in table D.5 and D.4 later in this appendix.

Role	Information class	R	D	P
Profession	Name	7	3	FR
	default	0	0	-
Pharmacist	Social security number	5	3	FR
	Medical treatment	7	4	FRA
	default	-	-	-
Secretary	Personalia	6	3	FRU
	Current.symptoms	4	2	FRUCA
	default	-	1	-
Nurse	Treatment	7	4	FRUA
	default	-	2	-
Medical practitioner	CAVE	9	2	FRUCA
	Current	3	2	FRUCA
	Medical history	3	2	FRCA
	Test results	3	2	FRUCA
	default	-	-	-
Hospital Secretary	default	-	-	-
GP Secratary	default	-	-	-
Home nurse	Address	6	3	FR
	default	-	-	-
Psychiatrist	MedicalHistory.Treatment.Therapy	5	2	FRUA
	MedicalHistory.Problem.Diagnosis	5	2	FR
	Current.Problem	5	2	FRUCA
	Current.Treatment.Medical	5	3	FRUCA
	default	-	-	-
General Practitioner	default	-	-	-
Dentist	Current.Treatment.Medical	5	3	FR
	Personalia	5	3	FR
	default	0	0	-
Cardiologist	MedicalHistory.Problem	5	3	FRUA
	Current.Problem	6	3	FRUCA
	Current.Treatment.Medical	6	3	FRUCA
	default	-	-	-
Medicine specialist	MedicalHistory.Problem	5	3	FRUA
	Current.Problem	6	3	FRUCA
	Current.Treatment.Medical	6	3	FRUCA
	default	-	-	-

Table D.1: Rating table for Professions

Role	**Information class**	**R**	**D**	**P**
Time	Name	7	3	FR
	default	0	0	-
Problem statement	Current.Treatment	7	3	FRUA
	MedicalHistory	7	2	FRUA
	default	-	-	-
Diagnostic patient	Test results	7	4	FRCA
	default	-	-	-
Treating patient	Current.Treatment.Therapy	7	4	FRUCA
	Current.Treatment.Medical	7	4	FRUCA
	Test results	7	4	FR
	default	-	-	-
Patient fit	Personalia	6	3	FRU
	MedicalHistory	5	2	FRU
	default	-	-	-

Table D.2: Rating table for time

Role	**Information class**	**R**	**D**	**P**
Job function	Name	7	3	FR
	default	0	0	-
Responsibility	default	-	-	-
Duty	default	-	-	-
Regular GP	Medical history	5	2	FRUA
	Current	5	5	FRUCA
	default	-	-	-
Treating doctor	Current	5	5	FRUCA
	Test results	4	2	FRUCA
	default	-	-	-
Referring doctor	Current	5	3	FRUCA
	Medical history	4	2	FRUA
	Current.Treatment.Medical	7	4	FRUCA
	default	-	-	-
Discharging doctor	Test results	4	3	FRUCA
	Medical history	3	2	FRUCA
	Current.Treatment	5	3	FRUCA
	Current.Problem	5	3	FRUCA
	default	-	-	-
Doctor on call	Current.Problem	7	3	FRUCA
	Current.Treatment.Medical	8	4	FRUCA
	default	-	-	-

Table D.3: Rating table for Job function

Level	Description of relevance level
0	Information is of no relevance at all and not to be shown unless asked for.
1	Information is of little or no relevance and not to be shown unless asked for.
2	Information is of little relevance and not to be shown unless asked for.
3	Inforamtion is of some relevance, but should not to be shown unless asked for.
4	Information is of relevance.
5	Inforamtion is of relevance and should be shown.
6	Information is of major relevance.
7	Inforamtion is of serious relevance.
8	Critical information that is always shown.
9	It is absolutely crucial that this information is given.

Table D.4: Relevance rating

Level	Description of detail level
0	No knowledge of existence
1	Knowledge of the existence
2	Knowledge of overall content
3	Knowledge of content
4	Knowledge of content and information regarding content
5	Knowledge of relation of content

Table D.5: Detail rating

Appendix E

Original text for the assignment

E.1 Oppgavetekst

Rollemodeller for helsepersonell Helsevesenet er meget informasjonsintensiv. En forutsetning for IT-baserte løsninger er å kunne ha tilfredsstillende kontroll over tilgang til å oppdatere, lage, avsende, lese, signere... sensitiv informasjon. I helsevesenet er det praktisk å regulere tilgangen i forhold til roller, og ikke enkeltpersoner, profesjoner eller organisasjoner.

Oppgaven består i å studere kommende standarder for helseinformasjon og foreslå en rollemodell i forbindelse med henvisning/epikrise mellom primærlege og sykehus. Deretter å implementere en LDAP-basert katalogtjeneste.

Oppgaven vil gjøres i samarbeid med prosjekter tilknyttet KITH og SINTEF.