Lillian Røstad

Access Control in Healthcare Information Systems

Thesis for the degree of Philosophiae Doctor (PhD)

Trondheim, June 2008

Norwegian University of Science and Technology
Faculty of Information Technology,
Mathematics and Electrical Engineering
Department of Computer and Information Science

NTNU
Norwegian University of Science and Technology

Thesis for the degree of ....

Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Computer and Information Science

 ISBN ... (printed ver.)
ISBN ... (electronic ver.)
ISSN ...

Theses at NTNU, 2008:...

Printed by Tapir Uttrykk

Logic will get you from A to B. Imagination will take you everywhere.

(Albert Einstein)

Abstract

Access control is a key feature of healthcare information systems. Access control is about enforcing rules to ensure that only authorized users get access to resources in a system. In healthcare systems this means protecting patient privacy. However, the top priority is always to provide the best possible care for a patient. This depends on the clinicians having access to the information they need to make the best, most informed care decisions. Care processes are often unpredictable and hard to map to strict access control rules. As a result, in emergency or otherwise unexpected situations, clinicians need to be able to bypass access control. In a crisis, availability of information takes precedence over privacy concerns. This duality of concerns is what makes access control in healthcare systems so challenging and interesting as a research subject.

To create access control models for healthcare we need to understand how healthcare works. Before creating a model we need to understand the requirements the model should fulfill. Though many access control models have been proposed and argued to be suitable for healthcare, little work has been published on access control requirements for healthcare. This thesis work has focused on exploring these requirements.

The process of trying to better understand the requirements for access control in healthcare has led to a number of smaller, distinct, but related projects being conducted within the context of this thesis. The main focus areas and contributions can be summarized as:

- Requirements: Studies performed on audit data, in workshops, by observation and interviews have helped discover requirements. Results from this work include methods for access control requirements elicitation in addition to the actual requirements discovered.
- Process-based access control: The main conclusion from the requirements work is that access control should be tailored to care processes. Care processes are highly dynamic and often unpredictable, and access control needs to adapt to this. This thesis suggests how existing sources of process information may be used for this purpose.
- Patient-controlled health records (PCHR): In a PCHR the patient is the administrator of access control. This thesis explores the consequences of making the patient the administrator and proposes a model for access control in a PCHR. A usability study has been performed to investigate how visualization can help keeping the patients informed of the consequences of their actions when they are in charge of access control.

Contents

Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) in partial fulfillment of the requirements for the degree *Philosophiae Doctor (PhD)*. The work has been conducted at the Department of Computer and Information Science (IDI), NTNU, Trondheim, Norway, under supervision of Associate Professor Øystein Nytrø.

Acknowledgements

I would like to thank my advisor Associate Professor Øystein Nytrø and co-advisor Professor Svein Johan Knapskog for all their help. I would also like to thank my colleagues at SINTEF ICT for their support and help. A special thanks to my co-authors Per Håkon Meland, Inger Anne Tøndel and Ole Edsberg for their contributions. I would also like to thank Ole Andreas Alsos for help with the usability study and Thomas Brox Røst for help with reading and reviewing. Thanks are also due to Associate Professor Kenneth Mandl of the Children's Hospital Informatics Program and Professor Pete Szolovits at MIT for allowing me to spend six months working with their team.

The work presented in this thesis has been financed by The Research Council of Norway.

Lillian Røstad
August 18, 2008

Part I

Introduction

Chapter 1

Introduction

It's a job that's never started that takes the longest to finish.
J. R. R. Tolkien

1.1 Background and motivation

Healthcare is information-intensive – vast amounts of information is created in the course of a treatment process and access to this information is important in the continuing care for the patient. Over the last decade, Electronic Health Records (EHR[1]) have become increasingly common and widespread. However, the EHR is not the only clinical information system in use. In any single hospital one can often count more than a hundred separate clinical information system, ranging from common systems like the EHR to specific X-ray or lab systems. In addition to these systems, much information is still only available on paper. Even if healthcare is very technologically advanced in areas such as surgical equipment and patient monitoring, in many aspects health care information systems are still in their infancy.

Most healthcare information systems are local to a hospital or a doctor's office, but patients are not. Patients move, become ill while traveling or simply choose to use a different healthcare provider. Patients with long-term illnesses and/or a complex diagnosis may receive services from many providers. There is a disconnect between the way information is managed and the needs for access to that information. Information remains in one place but patients do not. The explanation for this disconnect is not simply the technological challenges of integrating systems and information, but can be found in the legal frameworks regulating how sensitive

[1] Also often referred to as Electronic Patient Records (EPR)

clinical information may, or may not, be shared. There is a strong, current focus on resolving these issues, based on the fact that availability of correct, up-to-date and complete information is crucial to make the best, most informed care decisions.

In 1997, the Norwegian government published a report named *The patient first!* (nou [1997]) outlining visions for a more patient-centric healthcare. "Openness", "availability" and a more "coherent care process" are among the ten goals listed in the report. To fulfill these goals, availability of information needs to be improved for healthcare personnel as well as for the patient and next of kin. The goals put forth in this report are consistent with the current focus in healthcare worldwide. Since the publication of the report in 1997 many projects have focused on improved availability through information and systems integration, while other projects have chosen the path of improved availability through new systems where the patient manage access to information.

Healthcare systems may be categorized as *security-critical systems*. Other systems often described as being security-critical include railroad signaling systems, nuclear plant control systems, air traffic control systems as well as financial systems such as the online banking systems most of us use. Access control is an important feature of all of these systems, but healthcare systems are different in one important aspect. Access control is about making sure information is accessible only to authorized users. Whereas in most other security-critical systems the default access control rule is "when in doubt – deny", for healthcare it will always be "when in doubt – allow". Protecting patient privacy is important, but the most important goal is to provide the best possible care for patients, which depends on the clinicians having access to information. Hence, access control is a balance between confidentiality and availability. This is what makes access control in healthcare systems so challenging – and interesting as a research subject.

To ensure availability in emergencies or otherwise unexpected situations, mechanisms that allow a user to override the access control is included in many healthcare systems. Allowing access control to be overridden implies including functionality in your system that may be misused. To minimize security risk, retrospective controls such as extensive auditing are usually employed. In Povey [2000] this is described as "optimistic security".

An example illustrates what may happen, from a patient perspective:

> JD has been suffering from abdominal pains for a while. As the pains seem to become more frequent and severe he makes an appointment to see his primary physician. The physician orders a number of tests but finds nothing wrong. He decides that JD should be admitted to the gastro[2] ward at the hospital for further testing. When JD arrives at the hospital the staff queries him about his medical history, what tests his

[2]Gastroenterology is the medical specialty devoted to the study, diagnosis and treatment of disorders of the digestive system (www.medterms.com).

> doctor performed and what the results were. They take blood samples from JD to send to the lab for further testing. When the tests are done the results are entered into the EHR via the lab system. The clinicians treating JD at the gastro ward then logs onto the EHR system and reviews the test results. The tests reveal a very high white blood cell count. They suspect that JD may have leukemia and decides to transfer him to the cancer ward immediately for further testing. The clinicians can transfer JD physically but cannot change his "admitted to" status in the system and JDs record is only accessible to people working at the gastro ward where he is currently admitted. The clinicians at the cancer ward are aware of this, so rather than using the EHR to prepare for JDs arrival they ask the gastro ward to print a summary of their findings so far and send it with JD. After having performed several more tests they log onto the EHR by overriding the access control to enter their findings (...)

This examples illustrates what frequently happens to patients: they are given an initial diagnosis that may change several times before the real problem is identified. As the diagnosis, tests and treatment change they are transferred back and forth and often asked to recount their medical history and treatment so far. The override mechanisms are often used to handle common, recurring events that are not allowed by the access control rules.

The existence of override mechanisms, and even the invention of the term "optimistic security", indicates that there is room for improvement on access control in healthcare. It is also clear from the previous discussion that access control for healthcare systems faces challenges that are unique and still unresolved.

1.2 Research focus

Current research on access control largely tends toward a theoretical approach. A quick search on google, or skimming the bibliography of this thesis, reveals a large number of scientific papers presenting diverse access control models. Many of them use healthcare as a motivating example, but very few are based on empirical studies that support the selection of model properties or explain in more detail why the models are suitable for a healthcare setting. Research on access control may be viewed as a continuum from theoretical via implementation-centric to user- and problem focused. Research so far tends toward the former while little has been done on the latter. Motivated by this fact, this PhD project has taken a practical approach to access control in healthcare. The focus has been on increasing our understanding of information and access needs to be able to create access control models that are better tailored to reality. Only when we have a better understanding of the everyday reality of healthcare workers and their patients

does it make sense to propose solutions in the form of access control models for healthcare information systems. The main goal is to create access control models that minimizes the need for use of override mechanisms and "optimistic security".

1.3 Publications

The research performed as part of this thesis has resulted in the publication of a number of scientific papers that are listed here. At the time of writing, 9 of the papers have been accepted for publication, while the last one has been submitted to a conference and is currently in the review process. The papers are listed here in the order they appear in part 2 of the thesis. Rather than using chronological ordering by publication date, the papers have been grouped according to topic.

Requirements for access control in healthcare

- Paper A: Lillian Røstad, *Access Control in Healthcare Applications*, NOKOBIT05, Bergen 22.-23. November 2005.
- Paper B: Lillian Røstad and Ole Edsberg, *A Study of Access Control Requirements for Healthcare Systems Based on Audit Trails from Access Logs*, Annual Computer Security Applications Conference (ACSAC), Miami, December 11-15 2006.
- Paper C: Lillian Røstad, *An extended misuse case notation: Including vulnerabilities and the insider threat*, The Twelfth Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'06), Luxembourg, June 5-6 2006.
- Paper D: Per Håkon Meland, Lillian Røstad, Inger Anne Tøndel, Øystein Nytrø, *The iAccess Handbook: A Methodology for Access Control Integration*, MedInfo 2007, Brisbane, August 20-24 2007.
- Paper E: Lillian Røstad, Per Håkon Meland, Inger Anne Tøndel and Øystein Nytrø, *Access Control and Integration of Health Care Systems: An Experience Report and Future Challenges*, Second International Workshop "Dependability Aspects on Data WArehousing and Mining applications" DAWAM 2007, in conjunction with The International Conference on Availability, Reliability and Security (ARES 2007), Vienna, April 10-13, 2007.

Process-based access control

- Paper F: Lillian Røstad, *MG-RBAC: Using Medical Guidelines as a Source of Contextual Information to Activate and Deactivate Roles and Permissions*, MedInfo 2007, Brisbane, August 20-24 2007.

- Paper G: Lillian Røstad and Øystein Nytrø, *Towards Dynamic Access Control for Healthcare Information Systems*, Medical Informatics Europe (MIE 2008), Gøteborg, Sweden, 26-28 May 2008. Extended edition invited to be published in the journal *Methods of information in medicine*.

Patient controlled health records

- Paper H: Lillian Røstad, *An Initial Model and a Discussion of Access Control in Patient Controlled Health Records*, The International Workshop on Privacy and Assurance (WPA-2008), in conjunction with The International Conference on Availability, Reliability and Security (ARES 2008), Barcelona, March 4-7, 2008.
- Paper I: Lillian Røstad and Øystein Nytrø, *Personalized Access Control for Personally Controlled Health Records*, accepted for publication at the 2nd Computer Security Architecture Workshop (in conjunction with 15th ACM Conference on Computers and Communication Security), George Mason University, Fairfax, Virginia, October 31st 2008.
- Paper J: Lillian Røstad, *Visualization for Patient-Administered Access Control: a Usability Study*, submitted to: the 24th Annual Computer Security Applications Conference (ACSAC2008), decision due: August 18th 2008.

1.4 Thesis outline

This thesis is divided into two parts: the introduction and the published papers.

The first part includes motivation and background for the project, a summary of the research process, summaries of the papers in part two, and concluding remarks including suggestions for future work.

The second part of the thesis contains the papers that constitute the results of this project.

At the end of the thesis a bibliography of all references used in the papers and this thesis is included.

Chapter 2

Research Process

"If you keep at it, one day something which at first appeared impossible will become merely something very difficult indeed."
Danny Paradise (American Yoga instructor, b.1943)

This chapter describes the research process and how the results of this PhD project came to be what they are.

2.1 A shift of focus - from solving to understanding

Albert Einstein is frequently quoted to have said:

> "If we knew what it was we were doing, it would not be called research, would it?"

This quote illustrates why research projects are hard to plan. Re-planning research projects as new knowledge is discovered is very common and often necessary. This is also true for this PhD project. The project started out with a brief problem description that was refined into a more detailed project plan over the first year of the project.

The original title of this thesis was *Context- and Role-Based Dynamic Access Control in Distributed Healthcare Information Systems* and the intention was to develop access control models that were "context-aware", "more dynamic" and overall better suited to the needs of healthcare. However, it quickly became clear that the information needed to design those models simply did not exist. As a re-

sult, the main focus of this thesis shifted towards eliciting requirements for access control in healthcare and exploring how this knowledge may be used.

2.2 Linking the pieces – publications and contributions

This PhD project has consisted of a number of smaller projects resulting in a total of 10 scientific papers. At the time of writing 8 of these papers have been accepted and published, while 2 are still in the review process. The papers summarize the main results and contributions from this PhD project.

Figure 2.1 shows an overview of the written papers, how they fit together and which papers inspired or led to others. The papers have been grouped by topic. The three main groups are: access control *requirements* (5 papers), *process-based access control* (2 papers) and access control for *personally controlled health records* (3 papers). The group *access control requirements* may be further decomposed into two subgroups: papers that propose methods for requirements elicitation (2) and papers that describe requirements (3). One of the method papers and one of the requirements papers have a special focus on system integration, as depicted in Figure 2.1.

The next few sections present the papers and contributions in more detail.

2.2.1 Access Control Requirements

As illustrated in the figure, and as mentioned in the previous section, a major focus has been on better understanding *requirements* for access control in healthcare and three of the published papers focus mainly on requirements. The first paper "Access Control in Healthcare Applications" was written very early on and published in 2005 at The Norwegian Conference on Organizations' use of Information Technology (NOKOBIT). The paper is based on a case study of electronic health record (EHR) systems in use in Norway and presents a generalized access control model for healthcare. The model comprises common properties of the access control models in the systems that have been studied. This generalized model illustrates how in many systems access control is static (define once - use many) and also explains how access in exceptional circumstances, such as emergencies, is handled by allowing the user to override access control. This is often referred to as "exception access".

Learning about the use of exception access mechanisms inspired the study of audit logs presented in the paper "A Study of Access Control Requirements for Healthcare Systems Based on Audit Trails from Access Logs". The study revealed that

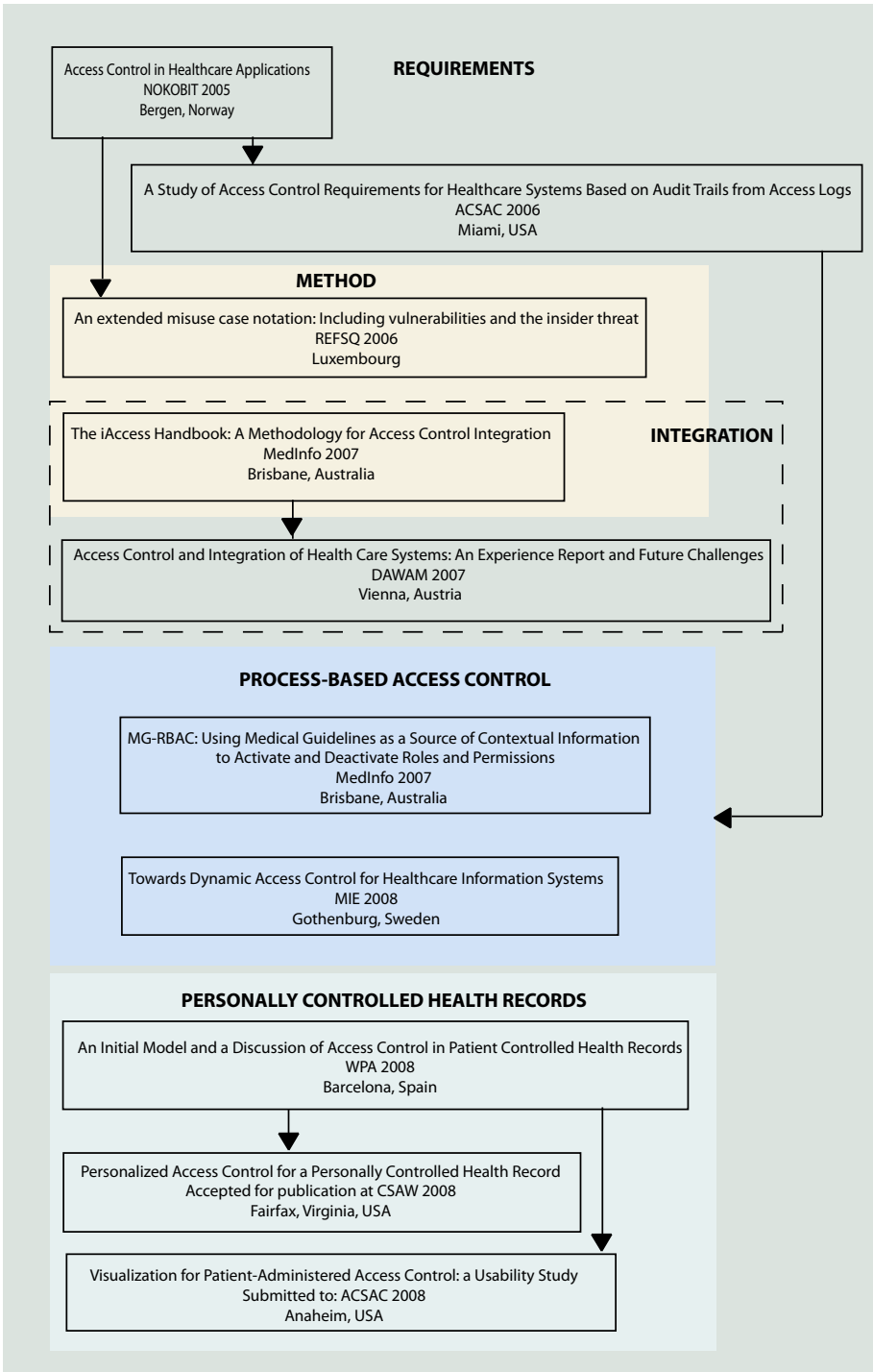

Figure 2.1: Papers grouped by topic

the use of of exception access mechanisms triggers extensive logging of the user's activities from that point on and the user has to provide a reason for using exception access. The idea for this paper was to examine how common the use of these access mechanisms were, and to see if the information in the audit logs could reveal requirements for access control. The assumption was that any frequently occurring event may be a candidate for inclusion as an allowed access control rule. The study revealed that exception access was used to access the EHRs of about 50% of the patients in the study period and 17% of all accesses were performed using exception access. We found that a relatively small number of similar reasons were provided for using exception access, including *out-patient clinic encounters* and *physician referrals*. Clearly these are events that should be handled by the access control.

2.2.2 Methods for access control requirement elicitation

Requirements elicitation requires the use of appropriate methods, which sometimes leads to creation of new methods or adaption of existing ones. Two papers presenting new or adapted methods for requirements elicitation have been published as part of this project.

The paper "An extended misuse case notation: Including vulnerabilities and the insider threat" presents an extension of the misuse case notation. The misuse case notation is itself an extension of UML use cases. Use cases described desired functionality in a system and misuse cases extend the UML use case notation by adding negative use cases and negative actors (colored black) to illustrate threats and attackers respectively. The main contribution of the extended misuse case notation is on adding notations to represents threats from insiders and to represent vulnerable system functionality. That is desirable system functionality that may be exploited by an attacker. This notation was created to be able to represent and illustrate the threat posed by exception access mechanisms in healthcare systems.

Integration

Information and system integration is among the most important challenges faced by healthcare today. Currently the reality is that there are hundreds of disconnected clinical systems in use at any hospital, and most systems are local to a ward, clinic or hospital. Access control is tightly coupled with information and depends on knowledge of information structure. This means that when integrating systems and information, access control integration is a major challenge.

One of the papers on requirement elicitation methods has a special focus on integration of healthcare information systems and access control. The paper, "The iAccess Handbook: A Methodology for Access Control Integration", presents a

handbook consisting of two parts: relevant reference information and a set of methods to collect, analyse, structure and represent relevant information for access control integration. The handbook focuses on the legal, technical and organizational aspects of integration and should serve as an aid in the access control integration process. At the time of writing, the handbook can be found online at http://iaccess.idi.ntnu.no.

Another paper that is part of this thesis, "Access Control and Integration of Health Care Systems: An Experience Report and Future Challenges", reports findings from a study of one such integration effort with a focus on access control. It turns out that in the system under study access control is not integrated at all. A portal has been created to offer a common access point to several subsystems. An access control mechanism implemented in the portal decides what subsystems a user can access through the portal, but deciding what access the user has to information is left up to the different subsystems by forwarding the user's credentials from the portal. This study illustrates a practical approach taken on the path towards integration and provides insight into the difficulties of access control integration.

2.2.3 Process-based access control

From the requirements work it became clear that many access control mechanisms implemented in existing systems are too static, in the sense that access rules rarely change or are updated after a patient is admitted. Access usually depends on what ward a patient is admitted to, where a user is working and the profession (doctor, nurse etc.) of the user. However, the care process for a patient is dynamic and individual. To improve access control, and minimize the use of exception access mechanisms, we suggest that access control should be process-based – tailored to workflows in healthcare and dynamic in the sense that access rules change or are updated as the treatment process changes.

Medical guidelines are idealized representations of the treatment process for a patient with a specific diagnosis. The paper "MG-RBAC: Using Medical Guidelines as a Source of Contextual Information to Activate and Deactivate Roles and Permissions" presents how the information in medical guidelines may be utilized in process-based access control. This is illustrated with an example medical guideline on treatment of Gestational Diabetes Mellitus (GDM)[1]. The guideline include instructions on how often the patient should see her doctor. This could be used to open the EHR for access for the doctor around the next scheduled appointment, rather than it being accessible all the time. According to the GDM guideline the patient is to monitor her glucose level and if it exceeds a certain limit she needs to see her doctor. If access to the EHR was linked to the GDM guideline the EHR could be made accessible to the patient's doctor in the event of a too high glucose level measurement.

[1] A form of diabetes found in pregnant women.

The study of audit logs led to a realization that the logs in healthcare systems contain information that could be used as a valuable source of work process knowledge. Data mining techniques could be applied to historical log data to extract usage patterns. These usage patterns could form the basis for access control rules. Continuous monitoring and mining of the logs would allow access rules to adapt as the care process evolves. The paper "Towards Dynamic Access Control for Healthcare Information Systems" suggests a model where audit data, in combination with medical guidelines and knowledge gathered by observation, may be used for more dynamic, process-based, access control in healthcare. Using the audit data in combination with observational data and guidelines enables us to tap into knowledge from three different viewpoints: the user, the system and the idealized version. Combining several viewpoints hopefully leads to a more complete understanding of information access needs.

2.2.4 Access control for Personally Controlled Health Records

Three of the papers that are part of this thesis focus on access control issues related to Personally Controlled Health Records (PCHRs). PCHRs may contribute to solving the challenges of information sharing and exchange because the patient is able to take his clinical information with him and decide who gets access. From an access control viewpoint, PCHRs are interesting mainly because the patient is made the administrator of access control.

The work presented in these papers was initiated when I spent six months in 2007 (February – July) as a visiting researcher at the The Harvard-MIT Division of Health Sciences and Technology (HST) working with the team at the Children's Hospital Informatics Program (CHIP) that are developing the Indivo[2] PCHR.

The paper "An Initial Model and a Discussion of Access Control in Patient Controlled Health Records" presents an overview of issues related to access control in PCHRs. The paper highlights transparency as one of the main concerns for access control in a PCHR. When the patient is administrator it is important that the system is transparent in the sense that the consequences of granting access are obvious and immediate to the patient. The goal is to make sure the patient is informed when sharing. The paper presents a sketch of a model for access control in a PCHR with a focus on important issues that need further investigation.

The paper "Personalized Access Control for a Personally Controlled Health Record" presents a more detailed model suggesting how these issues may be handled in an access control model for a PCHR. The model is semi-formally defined. Core properties of the model are the two sets of access policies (common and personal) and the definition of policy adaption hierarchies stating how policies may be combined.

[2] http://www.indivohealth.org/

As a patient could be anybody, very little can be assumed about the user in terms of technical skills, computer literacy etc. Therefore the usability of the sharing interface (where access rights are granted) becomes crucial to keep the patient informed about consequences of sharing. In the last paper, "Visualization for Patient-Administered Access Control: a Usability Study", findings from a usability study where three different demos for visualization in the sharing user interface for a PCHR were evaluated and compared is reported. The demos were built using a mock-up of the Indivo user interface and the people participating in the testing were familiar with the Indivo system and the PCHR concept. All the demos allowed the user to either select a predefined access policy when sharing or create a new policy, based on an existing one or from scratch. The contents (permissions on information) of the policies were visualized in the interface using different styles in the different demos: list, cube and rainbow. The study resulted in some valuable insights. One interesting thing was that once the patients realized that they could change the policies to fit what they wanted exactly, all of the users almost always did make some changes. Another interesting point that came up is the fact that system-defined access policy templates have a lot of authority. If the users trust the system then they may implicitly trust that the suggested policies and assume that they are correct. Therefore creating this policies is a major responsibility.

Chapter 3

Paper Summaries

There is no spoon.
(The Matrix)

This chapter presents the papers that have been published as part of this thesis project. For each paper a short summary is given including the original abstract, publication details, author contributions and remarks including any errors discovered after publication.

3.1 Paper A: Access Control in Healthcare Applications

3.1.1 Abstract

Healthcare personnel are dependant on access to relevant information to be able to provide the best possible health care for their patients. Designing access control for healthcare information systems is tricky, because of the dynamic nature of the organizations and the tasks performed. Most existing implementations solve this need through the use of access control exception mechanisms: if the normal access control mechanism won't grant a user legitimate access it is possible to use some exception mechanism to gain access to required information - for example in the case of an emergency. This paper discusses the special needs of access control in healthcare information systems and presents how these needs have been solved, or attempted solved, based on a study of a selection of clinical healthcare systems in use that has resulted in a generalized access control model. Role-based access control (RBAC) is the common principle for designing these access control mechanisms, and this article concludes with a discussion on how these implementations deviate from the RBAC principle and discusses if RBAC is indeed sufficient to fulfill the requirements of healthcare systems or if we need extensions to RBAC or an entirely new access control principle.

3.1.2 Publication details

This paper was published in the proceedings of NOKOBIT 2005 (The Norwegian Conference on Organizations' use of Information Technology) and presented at the NOKOBIT conference in Bergen, Norway in November 2005.

3.1.3 Author contributions

This paper was written entirely by Lillian Røstad.

3.1.4 Remarks

This paper was written very early on. It presents a summary of the author's understanding of access control in healthcare based on a limited study of existing systems.

3.2 Paper B: A Study of Access Control Requirements for Healthcare Systems Based on Audit Trails from Access Logs

3.2.1 Abstract

In healthcare, role-based access control systems are often extended with exception mechanisms to ensure access to needed information even when the needs don't follow the expected patterns. Exception mechanisms increase the threats to patient privacy, and therefore their use should be limited and subject to auditing. We have studied access logs from a hospital EPR system with extensive use of exception-based access control. We found that the uses of the exception mechanisms were too frequent and widespread to be considered exceptions. The huge size of the log and the use of predefined or uninformative reasons for access make it infeasible to audit the log for misuse. The informative reasons that were given provided starting points for requirements on how the usage needs should be accomplished without exception-based access. With more structured and fine-grained logging, analysis of access logs could be a very useful tool for learning how to reduce the need for exception-based access.

3.2.2 Publication details

This paper was published in the proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC 06), IEEE Computer Society, ISBN 0-7695-2716-7. The paper was presented at ACSAC in Miami, Florida, December 2006.

3.2.3 Author contributions

This paper was written by Lillian Røstad and Ole Edsberg. Røstad wrote the main parts of the paper. Edsberg contributed on data analysis.

3.2.4 Remarks

Errata

Some minor errors in the numbers were discovered after publication. They do not affect any of the conclusions, but are presented here for completeness purposes:

- In Table 2 the number of users with actualization permission should be 12289, not 12298.

- In Table 3 the number of EPRs accessed using emergency access should be 48 not 67. 67 was the total number of emergency accesses. 48 was the number of patients whose records were accessed using emergency access.
- In Table 4 the total number of accesses using actualization should be 275762, not 297742. This is due to an error in the query. This means that the average number of accesses of one EPR within one actualization period should be 2.06 and not 2.31.

3.3 Paper C: An extended misuse case notation: Including vulnerabilities and the insider threat

3.3.1 Abstract

Misuse cases are a useful technique for eliciting and modeling security requirements and threats. In addition they may be very useful in a risk analysis process, particularly as part of the system development process. The original misuse case notation adds inverted use cases to model threats and inverted actors to represent attackers. However, an attack is usually performed by exploiting a vulnerability in a system and it would be useful to be able to represent vulnerable functions in a model. In addition, it should be possible to discern between insiders and outside attackers in a model, as they have very different abilities and potential for attacking a system. This paper therefore proposes an extended misuse case notation that includes the ability to represent vulnerabilities and the insider threat, and discusses the use of this extended notation in the system development and risk analysis processes.

3.3.2 Publication details

This paper was published in the proceedings of REFSQ 2006 (The Twelfth International Working Conference on Requirements Engineering: Foundation for Software Quality) and presented at REFSQ in Luxembourg, June 2006.

3.3.3 Author contributions

This paper was written entirely by Lillian Røstad.

3.3.4 Remarks

The creation of the notation presented in this paper was motivated by the fact that access control mechanisms in healthcare often includes break-the-glass mechanisms that clearly can be misused. We needed a notation to be able to express the risk represented by these mechanisms. However, the notation has proved useful on a more general basis. Creating misuse cases is an important activity when performing risk analysis of software systems. This extended notation helps highlight what functionality in a system may be misused and makes it easier to link countermeasures directly to the vulnerable functionality.

3.4 Paper D: The iAccess Handbook: A Methodology for Access Control Integration

3.4.1 Abstract

Health care information about a patient is usually scattered among several clinical systems - potentially more than a hundred separate systems just within one hospital. System integration and interoperability is difficult to achieve, and various strategies for integration exist. However, one topic that has not received much attention is how to integrate system specific security mechanisms such as access control. This paper presents the iAccess handbook, which is a tool to aid this process. It consists of a repository of reference information and a set of methods for collecting information and presenting results, and concerns the legal, organizational and technological aspects of integrated access control for health information systems. The methods have been applied on two separate integration efforts in Norway, which affect ten hospitals in total.

3.4.2 Publication details

This paper was accepted for poster presentation at MedInfo 2007 - 12th World Congress on Health (Medical) Informatics, Brisbane, Australia, August 2007.

3.4.3 Author contributions

This paper is based on work performed in the iAccess-project. iAccess was funded by the The Research Council of Norway and project participants included the Norwegian University of Science and Technology and the independent research organisation SINTEF. This paper was co-authored with Per Håkon Meland and Inger Anne Tøndel from SINTEF and Øystein Nytrø. All authors contributed equally to the work presented in the paper. The paper was written mainly by Meland, Røstad and Tøndel. Nytrø contributed with comments and refinement of the paper.

3.4.4 Remarks

The original 5-page paper is included in this thesis. After acceptance the paper had to be reduced to a 2-page summary and resubmitted. The original version is included here because it provides more detail.

3.5 Paper E: Access Control and Integration of Health Care Systems: An Experience Report and Future Challenges

3.5.1 Abstract

Health information about a patient is usually scattered among several clinical systems, which limits the availability of the information. Integration of the most central systems is a possible solution to this problem. In this paper we present one such integration effort, with a focus on how access control is handled in the integrated system. Although this effort has not yet solved all the issues of access control integration, it demonstrates a practical approach for creating something that works today and serves as input to the discussion on future challenges for access control when integrating multiple systems.

3.5.2 Publication details

This paper was published in the proceedings of The Second International Conference on Availability, Reliability and Security (ARES 2007), IEEE Computer Society, ISBN 0-7695-2775-2. The paper was presented at the Second International Workshop "Dependability Aspects on Data WArehousing and Mining applications" (DAWAM 2007) in conjunction with ARES in Vienna, Austria, April 2007.

3.5.3 Author contributions

This paper is based on work performed in the iAccess-project. iAccess was funded by the The Research Council of Norway and project participants included the Norwegian University of Science and Technology and the independent research organisation SINTEF. This paper was co-authored with Per Håkon Meland and Inger Anne Tøndel from SINTEF and Øystein Nytrø. The paper was written mainly by Røstad, Meland and Tøndel. Nytrø contributed with comments and refinement of the paper.

3.5.4 Remarks

This paper illustrates how hard integration of access control is. The effort described here is far from a solution, but serves well as an illustration of a practical approach and to highlight challenges.

3.6 Paper F: MG-RBAC: Using Medical Guidelines as a Source of Contextual Information to Activate and Deactivate Roles and Permissions

3.6.1 Abstract

Controlling access to information is a key concern in healthcare systems. Some form of Role-Based Access Control (RBAC) is implemented in most healthcare systems. A problem with existing RBAC models used in healthcare is their static nature which doesn't capture the dynamic needs of healthcare providers. In this paper we propose an enhanced access control mode combining RBAC with the use of Medical Guidelines, MG-RBAC. Medical guidelines contain temporal and contextual information that may be used to make more informed, dynamic access control decisions.

3.6.2 Publication details

This paper was accepted for poster presentation at MedInfo 2007 - 12th World Congress on Health (Medical) Informatics, Brisbane, Australia, August 2007.

3.6.3 Author contributions

This paper was written entirely by Lillian Røstad.

3.6.4 Remarks

The original 4-page paper is included in this thesis. After acceptance the paper had to be reduced to a 2-page summary and resubmitted. The original version is included here because it provides more detail.

3.7 Paper G: Towards Dynamic Access Control for Healthcare Information Systems

3.7.1 Abstract

Access control is a key feature of healthcare information systems to protect the privacy of patients and to ensure access to information as required by healthcare professionals. A problem with many existing access control mechanisms is their static nature. In this paper we propose combining workflow information from medical guidelines, observations and audit logs to create dynamic access rules that are adapted to the actual workings of a hospital. Our aim is to help minimize the use of "break the glass" access.

3.7.2 Publication details

This paper was published in the proceedings of The XXIst International Congress of the European Federation for Medical Informatics (MIE), IOS Press 2008, ISBN 978-1-58603-864-9. The paper was presented at MIE 2008 in Gothenburg, Sweden, May 2008.

3.7.3 Author contributions

This paper was written by Lillian Røstad and Østein Nytrø. The main parts of the paper were written by Røstad. Nytrø contributed to the discussion and provided feedback.

3.7.4 Remarks

This paper is important for the thesis in that it proposes future work. Unavailability of the required audit data unfortunately made it impossible to investigate the proposed approach in further detail within the scope of this thesis.

After the MIE conference, we have been invited to submit an extended edition of this paper to be published in the journal *Methods of information in Medicine*.

3.8 Paper H: An Initial Model and a Discussion of Access Control in Patient Controlled Health Records

3.8.1 Abstract

Health information about a patient is usually kept local to the hospital or clinic where the patient was treated. Patient Controlled Health Records (PCHR) has been proposed as a means to collect all this information and make it available to the patient. In a PCHR the patient is in control and determines who gets access to his health information. In this paper we present a set of usage scenarios to explore the concept of a PCHR. From the scenarios we deduce a set of concerns of relevance when designing an access control model for a PCHR. Finally we outline an initial access control model for a PCHR.

3.8.2 Publication details

This paper was published in the proceedings of The Third International Conference on Availability, Reliability and Security (ARES 2008), IEEE Computer Society, ISBN 0-7695-3102-4. The paper was presented at The International Workshop on Privacy and Assurance (WPA-2008) in conjunction with ARES in Barcelona, Spain, March 2008.

3.8.3 Author contributions

This paper was written entirely by Lillian Røstad.

3.8.4 Remarks

This paper is primarily a discussion paper and serves as background for the next two papers on personally controlled health records.

3.9 Paper I: Personalized Access Control for a Personally Controlled Health Record

3.9.1 Abstract

Access control is a key feature of healthcare systems. Up until recently most healthcare information systems have been local to a healthcare facility and accessible only to clinicians. Currently there is a move towards making health information more accessible to patients. One means for achieving this is the Personally Controlled Health Record (PCHR) where the patient is in charge of deciding who gets access to the information. This poses new challenges for access control. In the PCHR the patient is the administrator of access control. While it certainly is possible to create roles representing people most patients would want to share with, like primary physician, it is also likely, and desirable, to afford the patients a high level of control and freedom to be able to create specialized access policies tailored to their personal wishes. We entitle this personalized access control. In this paper we present a semi-formal model for how we believe personalized access control may be realised. The model draws on and combines properties and concepts of both Role-Based Access Control (RBAC) and Discretionary Access Control (DAC) to achieve the desired properties. Throughout the paper we use the PCHR as a motivating example and to explain our reasoning and practical use of the model.

3.9.2 Publication details

This paper has been been accepted for publication at the 2nd Computer Security Architecture Workshop (in conjunction with 15th ACM Conference on Computers and Communication Security) to take place at George Mason University in Fairfax, Virginia on October 31st 2008.

3.9.3 Author contributions

This paper was written by Lillian Røstad and Østein Nytrø. The main parts of the paper were written by Røstad. Nytrø contributed to the discussion and provided feedback.

3.9.4 Remarks

This paper will be refined and a final version published in the proceedings of CSAW.

3.10 Paper J: Visualization for Patient-Administered Access Control: a Usability Study

3.10.1 Abstract

A Patient-Controlled Health Records (PCHRs) allow patients complete control over their health information. They decide who to share their information with, which makes the patient the administrator of access control. While PCHRs have a great potential for patient empowerment, they have an equally great risk for breach of privacy if consequences of sharing are not completely clear to the patient. This paper presents results from a usability study that compares three different visual interfaces for sharing in a PCHR. The goal of this study was to evaluate if a visual interface can help make the effects of sharing more transparent to the patient.

3.10.2 Publication details

This paper has been submitted to the 24th Annual Computer Security Applications Conference (ACSAC 08). Decision is due August 18th.

3.10.3 Author contributions

This paper was written entirely by Lillian Røstad.

3.10.4 Remarks

If accepted, this paper will be refined based on the feedback from the reviewers and a final version to be published will be submitted to ACSAC in September 2008.

Chapter 4

Concluding Remarks

"Security is the art of making sure certain things does not happen. A thankless task, because when something doesn't happen, there will always be someone who claims that the security measures were exaggerated and unnecessary."
(Salman Rushdie)

This thesis consists of a total of 10 scientific papers. The papers range in focus from requirements engineering for access control, to proposed models based on those requirements.

This PhD project started out with a narrow focus on creating dynamic, context-aware access control models for healthcare. Upon realizing that we didn't have the knowledge to create meaningful models the scope was widened and the focus shifted from creating solutions to gaining a better understanding of the problem.

As a result the main contributions of this thesis are on knowledge about how existing access mechanisms function, what works well and what could be improved. Studies performed in this thesis indicate that access control in healthcare should be more dynamic and adaptable to be able to support the unpredictable, dynamic and individual care process. The results suggest a step towards dynamic access control would be to utilize contextual knowledge, such as medical guidelines and audit data, to allow access rules to change as the context changes.

The consequence of focusing on requirements elicitation, and taking a broad approach, is that while some models have been suggested they are not complete and need more work and refinement. Therefore directions for further work is an important contribution from this project.

4.1 Future work

There are two main directions for future work based on the findings presented in this thesis: process based access control and access control for personally controlled health records.

Process based access control

The work on access control requirements for healthcare has resulted in a proposed model that suggests using knowledge in the form of observations, medical guidelines and mining of audit logs in access control. A logical next step would be to create a more detailed model and implement a proof-of-concept that could be used for evaluation purposes. However, before this can be done work remains on how to extract, represent and combine this knowledge. In particular, further work should focus on:

- **Extracting usage patterns**: Data mining of logs to extract usage patterns is a non-trivial task. A precondition for performing this work is the availability of sufficiently large amounts of high-quality log data. High-quality in this context means that the logs contains sufficiently detailed information to construct meaningful usage patterns. Usage patterns may be simple or more complex and may include information on location, responsibility and roles of the present users, time and situation. It is necessary to perform research on a large corpus of data to gain knowledge on the structure and composition of meaningful usage patterns.

- **Rules for creating permissions**: That a usage pattern is common does not necessarily mean that it should be included as an access rule. Likewise, that an event is common does not necessarily imply that it should be disallowed. Studies and experiments should be performed to create rules for when a usage pattern is a candidate for an access control rule.

- **Misuse detection**: Usage patterns may also be used to create application-level Intrusion Detection Systems (IDS) for healthcare. Realizing that mechanisms to override access control will always exist in healthcare systems to handle emergencies, means accepting that there will always be a need for retrospective access control. An IDS is a system that helps automate the process of misuse detection. An IDS is either based on signatures of known attacks, or on learning common usages. Misuse is suspected when there are deviations from normal use, represented by the usage patterns.

Access control for PCHRs

A model for access control in PCHRs has been proposed, but this model requires further refinement. Also, a model is only useful if it gets implemented and evaluated which is the logical next step. But a correctly implemented model is not sufficient for access control in a PCHR. Further studies should focus on:

- **Usability**: As the patient becomes the administrator, the usability of the systems in the way it communicates consequences to the user is important. Further usability studies and tests on the sharing interface and how the users respond to that should be performed to realize the potential for patient empowerment in a PCHR.
- **Common policies**: A very interesting comment made in the usability study performed in this PhD project was on how the user perceived the authority of any suggestions made by the system. If the system suggests an access profile it seems users may be likely to trust the system and accept that suggestion without any closer examination. It would be interesting to confirm if this is true and explore the consequences. Should or should there not be common policies?
- **Creating common policies based on the most common personal policies**: Granting users the power to create their own access profiles means that over time we get a collection of access profiles representing who users share with and what they share. This dataset could be used to create better common profiles that are based on actual use rather than assumptions.

Part II

Papers

Paper A

Access Control in Healthcare Applications

Access Control in Healthcare Applications

Lillian Røstad
Norges teknisk-naturvitenskapelige universitet (NTNU)
Sem Sælands vei 7-9
7491 Trondheim
lilliaro@idi.ntnu.no
http://www.idi.ntnu.no/~lilliaro

Abstract

Healthcare personnel are dependant on access to relevant information to be able to provide the best possible health care for their patients. Designing access control for healthcare information systems is tricky, because of the dynamic nature of the organizations and the tasks performed. Most existing implementations solve this need through the use of access control exception mechanisms: if the normal access control mechanism won't grant a user legitimate access it is possible to use some exception mechanism to gain access to required information - for example in the case of an emergency. This paper discusses the special needs of access control in healthcare information systems and presents how these needs have been solved, or attempted solved, based on a study of a selection of clinical healthcare systems in use that has resulted in a generalized access control model. Role-based access control (RBAC) is the common principle for designing these access control mechanisms, and this article concludes with a discussion on how these implementations deviate from the RBAC principle and discusses if RBAC is indeed sufficient to fulfill the requirements of healthcare systems or if we need extensions to RBAC or an entirely new access control principle.

Introduction

Access control is the process of determining which users are allowed to perform what operations on which objects in a computer system. Healthcare information systems contain sensitive information about patients that is vital in the treatment process. As such access control in the healthcare sector is about protecting the patient's right to privacy, while ensuring that healthcare personnel get access to the right information at the right time in order to be able to provide the best possible treatment for their patients. Healthcare is one of the most information intensive sectors in the society. As more and more of the clinical information about a patient is recorded in information systems, it becomes increasingly important to have sound and sufficient mechanisms for providing and restricting access to this information. The old paper-based record is becoming a thing of the past, and as all this information about a patient is being transferred into digital and networked systems the risk scenario changes. Earlier, in order to gain access to information, one had to physically locate where the information could be found and track down the actual papers. Now this is only a matter of searching through a database of available information. Adding to this the fact that digital information can be easily

multiplied and transferred while papers have to be copied one by one - the potential for privacy breaches has definitely increased. This potential for much easier access, and also replication, has lead to an increased focus on information security in healthcare. Access control is at the heart of this focus as it is the key issue to be able to protect and make efficient use of this vast amount of digitally stored sensitive information.
Access control has two different dimensions that sometimes are in conflict. While the primary objective for applying access control is restricting access to information and functions, usability is an equally important feature. Access control designers need to understand how the organization functions as well as the access requirements of the system users. The result of applying too strict, or simply wrong, access control mechanisms will be users finding other ways of obtaining the information they need. Access control mechanisms implemented in health care information systems today has not proven entirely successful on the usability aspect, namely to support the working procedures of healthcare personnel. As a result, they have had to rely on allowing exceptions from the normal access control mechanisms to be able to satisfy the needs of their users. From an information security point of view exceptions are bad because it results in loss of control over information flow. The goal of the work described here is to study existing mechanisms, and comparing them with the standards the claim to comply with, in order to gain knowledge that may help in designing an improved access control model for healthcare applications.

Access control implementation in healthcare applications

The information about implemented access control mechanisms presented her is based on a study of access control mechanisms in a variety of clinical healthcare information systems. Although the specific details of implementation and chosen technologies may differ, they all have a lot in common. These common traits can be summarized in a *generalized access control model for healthcare* that is visualized in Figure 2. This model, forms the basis for our discussion. Before moving on we will explain the model in detail.

THE GENERALIZED ACCESS CONTROL MODEL FOR HEALTHCARE

All the systems that have been studied have the one thing in common that they claim to support role-based access control (RBAC). A majority of the systems have chosen an approach based on the concepts of roles and responsibilities in combination with contextual information like place of work. The underlying assumption is that any specific healthcare profession has a well-defined set of responsibilities and should be able to perform these responsibilities for patients currently under their supervision - namely patients admitted to the ward that is their primary place of work. In other words role is equal to healthcare profession and each hospital has a list of possible roles and in each separate healthcare information system this role is mapped to the related permissions.

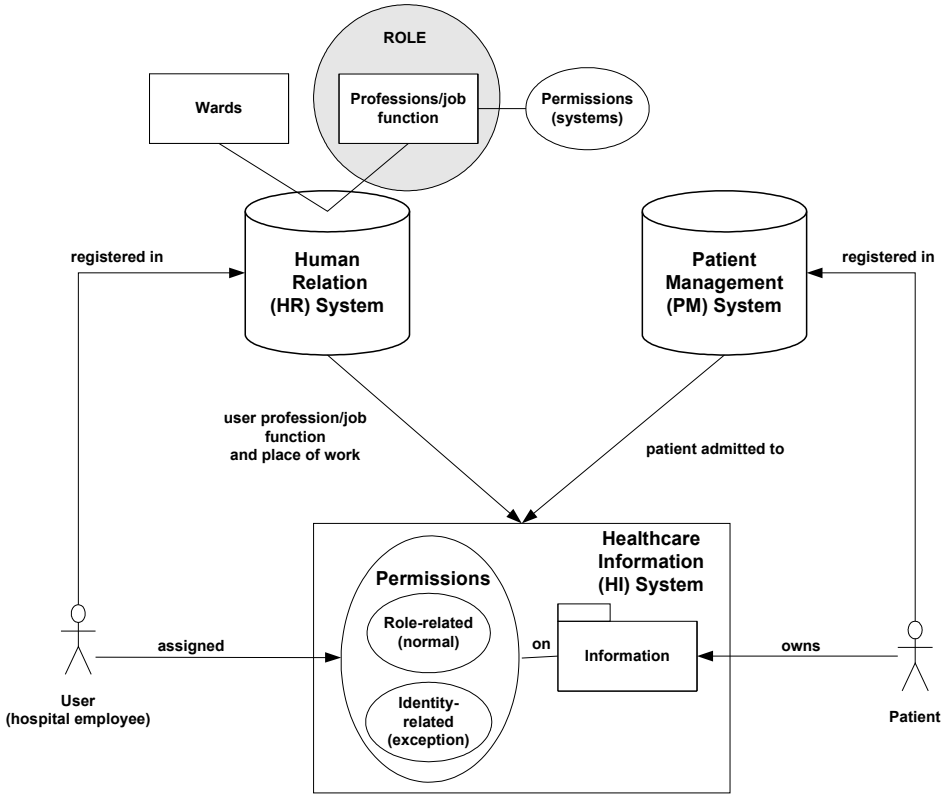

Figure 1 - This figure illustrates the generalized access control model for healthcare information systems.

This list of roles may also include non-healthcare professionals that are employed at a hospital, like secretaries. Information is associated with a patient, meaning that for an employee to have access to information about a patient in accordance with his/her role, the patient has to be admitted, and registered, to the ward at which he/she is currently employed. The information needed to construct permissions for a user is typically retrieved from the human-relations (HR) system that contains information about job position and place of work for a user, and combined with information about the patient's current location in the hospital from the patient-management (PM) system. Note that while role and place of work are system-independent properties registered externally, the permissions for a role in a specific system is enforced by the system itself, utilizing this external information, and may vary from one system to another. Notice also that in the figure a role is associated with organizational-level permissions in addition to specific permissions within one healthcare information (HI) system. These permissions determine which systems users associated with this role will be able to log on to. Typically, only systems in this list for a particular role will be available in a customized desktop for users assigned to this role. One other thing worth noting is the division of system-specific permission in those related to role and those related directly to the user's identity. Role-related permissions is part of what we may call the *normal* access control mechanism -

this is supposed to handle most access requests. The exception access permissions have been added to deal with access in emergency and unexpected situations. Exception access permissions are linked directly to identity because only a small and manageable number of users should have this permission as allowing exceptions raises the level of risk for privacy breaches. We will now move on to explain the functioning of this generalized model in more detail through examining access control administration and the use of exceptions. The use of exceptions is a key issue as it illustrates the shortcomings of the generalized model for access control in healthcare.

ACCESS CONTROL ADMINISTRATION

Access control administration is about assigning and withdrawing permissions for authorized users. Administering access control in the generalized model for healthcare have two aspects: administering access rights (role, place of work, permissions) for users and tracking patients. There are generally five events that trigger the need for manual administration of access control related information:

- Hiring a new employee.
- An employee changing job function and/or place of work.
- An employee quitting.
- A patient being admitted to a ward.
- A patient being transferred from a ward.

Access right administration today is largely a manual process governed by the filling-out, signing and filing of specific forms. For example, to assign permissions to a newly-hired employee, the manager of the ward where the employee is going to work fills out and signs a form stating the job function (role) and name of the ward. This form also contains information about assigning the user exception access rights in specific systems, if any. The form is then sent to the IT-department where the information is entered into the HR system. Through this process the user is granted access to a set of systems that are part of the permissions for his/her role. The information from the HR system will then be used by each of these systems to determine what the user should be allowed to see and do in each particular system. If an existing employee changes job function or place of work, this triggers the filling-out and filing of another form equal to the first one. At any time the most current form is the one used by the IT-department for entering access rights. This form has a from-date and a to-date, and what happens when an employee is quitting is that the to-date is filled in and the form re-signed. Setting the to-date in the HR system means deactivating access to all systems for this user. As for patient tracking, information about planned patient admissions are received from the patient's primary physician and the date and ward of the patient's expected arrival entered into the PM system. When the patient is discharged a letter summarizing the treatment and results is sent to the patient's primary physician and this triggers the setting of an admission date in the PM-system. Setting the admission date means cutting of access to information about this patient. All this sounds fairly simple, even if work-consuming. However, there are some complications:

- A lot of the times a patient's arrival is not planned. Patients often get admitted due to minor or major acute emergencies. These patients are not registered in the PM system at

the time treatment begins. Often, in case of minor injuries, the patient has already been discharged when the information is finally entered into the PM system.
• Sometimes information flow where patient's do not, for example in the case of lab tests, or one doctor requesting a second opinion from another doctor.
• It is not practical to close access to information about a patient immediately when he or she gets discharged or transferred to another ward. As mentioned earlier there may still be information that need to be documented after the patient have departed - for example test results or notes in the patient record about treatment and observations. Documentation is often done, or completed, post-treatment.
• It is not practical to simply close access when an employee quits - the responsibilities of this employee need to be transferred to someone else first. In order to be able to handle these complicating situations, and other unexpected situations, the base access control philosophy for healthcare systems has been extended with the ability to handle exceptions from normal or predicted treatment and information flow. As a result, while in most information systems there are only two possible outcomes of an access request (granted or denied), in healthcare systems there are four possible outcomes:
• Normal access granted.
• Normal access denied.
• Exception access granted (normal access denied).
• Access denied.
Normal access here means the built-in access control policy of a system that is meant to handle most access request. When an employee has a legitimate need for access to information, but the normal access control mechanism denies access due to some complications described earlier, exception access may be utilized. However, often there exists information about patients that is considered extra-sensitive and that therefore even exception access cannot be used to access. Examples of such information are psychiatry records. Therefore access denied is always a possible outcome of an access request. Access control exceptions have been added as a response to specific needs, with little regard to the effect on the overall access control model or policy of a system. The result is a generalized model that is not directly based on one access control principle, but is a combination of several strategies. This adds to the complexity and difficulty of having an overview of which users have what access rights at any given time as permissions is not only linked to role but also directly to identity. Exception access is powerful concept that may result in broad access rights and to gain knowledge about what a specific user has actually had access to, one need to examine each separate use of exception access. Understanding why exceptions are necessary is the key to understanding the improvement potential for access control in healthcare, and we will therefore examine what triggers the use of exceptions in more detail.

ACCESS CONTROL EXCEPTIONS

Exception mechanisms were initially designed to handle access in emergency situations - thereby dealing with unexpected, but legitimate, access requests. Realizing that emergency situations do occur in healthcare, the access control designers constructed a mechanism to handle this called *blue-light access*. This permission was not related to job-function/role, but was designed as a special permission linked directly to a specific user.

The reasoning for this was that only a limited number of users should be assigned this access, and they should be selected on a personal basis not just based on profession. It should also be possible to trace the use of blue light on an identity basis. Typically only a few persons at a ward have this permission. Blue light access was only supposed to be used in emergencies, when a patient arrives and there is no time for the time-consuming process of registering the information necessary for access control to function. But as the access control designers realized that blue light access may be misused, they added some extra security measures. The measures may vary slightly from one system to another, but the most common ones include:

• Requiring the user to re-enter his/her password.
• Requiring the user to provide a reason for using blue light access.
• The use of blue light access grants access only to a smaller to subset of the available information about a patient, not the entire patient record. To gain access to other parts of the record, the user has to use blue light access again.
• Enforcing time-constraints - information accessed using blue light will only remain accessible for the user for a short time period, defined by the organization. Re-gaining access requires yet another use of blue-light access.
• The use of blue light access triggers extensive auditing of the users access from that point on.

The purpose of these measures is to raise the user's awareness that the mechanism is indeed only meant to be used in emergency situations. Making the mechanism less easy to use also reduces the risk for unintended use. This is important because this mechanism is essentially a built-in backdoor mechanism. It can be misused. The purpose of extensive auditing is twofold: as a preventive measure by letting users know that all their actions are traceable, but also as a means of controlling and being able to detect if the mechanism is misused. If blue light is misused the audit logs provides the evidence needed to take the appropriate actions, Blue light was only intended as a mechanism for handling access in emergency situations. However, as pointed out in the previous section, there are a number of other situations where the normal access control mechanism fails. The most common examples are referrals, second opinions - generally un-planned treatments. And it turns out (surprisingly - for the developers) that a major part of treatment at a hospital is un-planned. In fact at some hospital wards most admissions are not planned. As we have seen, planning is a crucial for the normal access control mechanism to function as patient's need to be registered in the system, and linked to the right ward, in order for health care personnel to be allowed to access their information. As this is often not the case, there are situations when exception access needs to be used although the situation is not really an emergency. In some systems blue light access is used also in these situations. However, some other systems have constructed yet another exception access mechanism to be able to handle these situations. This *exception access* mechanism is quite similar to blue light access, an a lot of the same prevention and detection mechanism are used. The main difference is that while blue light is only meant to be used in specific (emergency) situation that will always occur at hospitals and need to be handled, the general exception mechanism was constructed to handle all access that are legitimate and normal from a user's point of view, but that the normal access control mechanism of the system is not able to handle. Some other differences between blue light and the generic exception mechanism is:

• Exception access is also assigned directly to users, not through roles, but generally exception access is assigned to quite a few more users than blue light access is.
• Exception access generally results in access to a broader set of information - often the entire patient record, and often for a longer time period than blue-light access. Blue light access is only meant as a temporary mechanism because the normal one takes time to function due to all the manual updating required. Exception access may be used in situations where normal access is never expected to function - like entering information in a patient's record at a lab the patient never visits himself.
• As the hospitals have identified the most common reasons for using exception access, they have often constructed a drop-down list of common reasons, and associated time intervals, that the users may choose from when activating blue light access. The reason is to make the mechanism easier to use, which one may argue against from a security viewpoint but this is mainly a result of usability requirements. One may also discuss whether the users then will provide the correct reason, or simply the first one in the drop down list - or the one with the longest time interval for convenience reasons.
Constructing this exception access mechanism may be considered as admitting defeat for the access control designers. It was invented in a realization that the normal access control mechanism is not capable of handling all normal access.

CONCERNS

But is this a good, long-term solution? Shouldn't the main/normal access control model be able to handle all access requests? Figure 3 illustrates the use of access control mechanisms in three different situations that requires the use of both normal, blue light and exception access.

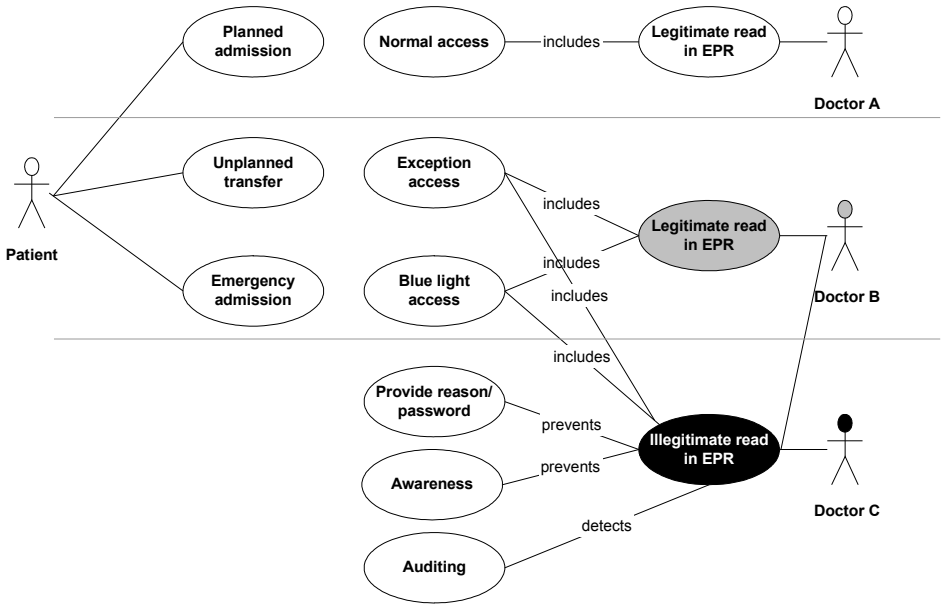

Figure 2 - This figure illustrates the three kinds of access and potential for misuse related to reading a patient's electronic patient record (EPR).

The figure is a combination of use-case and misuse-case where the left-hand side represents three different situations and the right-hand side illustrates actions taken by doctors when reading the patient's electronic patient record (EPR) in response to the situation on the left-hand side. The notation of misuse-cases is defined in (Sindre and Opdahl, 2005). The notation used here has some slight extensions. All white elements are ordinary use-case elements with their ordinary meaning. The *includes*-relationship means one use-case relying one another to perform its task. The black figures and the *prevents* and *detects* relationships are part of the misuse-case definition. Black elements symbolizes illegal or unwanted actions - in this context that means intentional illegitimate access by users, i.e. users accessing information without any medical reason or responsibility that justifies doing so. This is a potential consequence of blue-light and exception access that we want to avoid. The grey elements are not part of any published definition. They are used here to visualize the grey-zone areas of using exception access to perform legitimate access. As the figure shows, there is the possibility of a user (doctor B - the grey one) unintentionally obtaining illegitimate access. This may be due to share ignorance - the doctor simply does not know that what he is doing is wrong, for example accessing the EPR of a patient simply because he/she has heard from another doctor that it is an interesting case. This is not allowed according to the policy for normal access control that defines that there should be a relationship, in the form of a common organizational location, between a patient and a doctor for the doctor to be allowed access. The result is violating the patient's privacy rights, the hospital policy and probably also the country's laws and regulations - even if it was not deliberately. This illustrates how the introduction of exception access mechanisms increases the importance of awareness-raising measures. Responsibility for privacy-protection is shifted from the

system to the users. Making sure that everybody knows when exception access may and may not be used, and are aware of the possible penalties is an important preventive measure. The users need to be made aware, and reminded, of how the system is allowed to be used - and also what not to do. From a security point of view exceptions is a bad thing. Exceptions means loosing control of information flow. Exceptions mean unexpected behavior. Security is about having control and as such exceptions are bad. That said exceptions in the form of emergency access probably will always be part of access control for healthcare. Emergencies are a unique feature of this sector. But one should aim at minimizing the need for exceptions. And any required exceptions should be handled as part of the standard access control mechanism, to allow maximum possible control. It is probably not possible to not have exceptions at all - but a better suited access control model would minimize the need for using exceptions (leaving only emergencies). The remainder of this paper examines if the requirements of healthcare systems discussed here could be implemented through an access control mechanism based solely on the RBAC principle, without the need for exceptions or with support for handling exceptions incorporated into the model, not as an add-on.

Rbac in healthcare: theory versus practice

Role-based access control (RBAC) is considered to be state of the art for access control in healthcare systems today as we have seen. The generalized access control model for healthcare presented in this paper uses a variant of roles as part of its foundation. Before we can compare the generalized model with RBAC we need to take a closer look at the RBAC principle.

KEY CONCEPTS OF ROLE-BASED ACCESS CONTROL (RBAC)

The concept of RBAC was formalized in the paper *Role-based access control* (Farraiolo and Kuhn, 1992). The key concept in RBAC is to define roles that correspond to job titles and responsibilities within an organization. Each role is associated with a set of access rights. Employees holding the same job within an organization are assigned to the same role - thus the number of roles is considerably lower than the number of employees in an organization. As RBAC evolved from the original proposition into being implemented in commercial systems, the need for standardization arose. Different vendors started implementing RBAC in their security products, but there was no common understanding of what the main RBAC features are, and how they should be implemented. (Essmayr, Probst and Weippl, 2004) gives a thorough overview of RBAC implementation in commercial systems today. Having a common standard to use as a base specification is important to enable interoperability of the different implemented solutions. In response to this need the National Institute of Standards and Technology (www.nist.gov) in USA proposed a standard (Ferraiolo, 2001) for RBAC. The RBAC standard presents the core RBAC model, which encompasses the most important RBAC features. Figure 4 presents an overview of the core RBAC model.

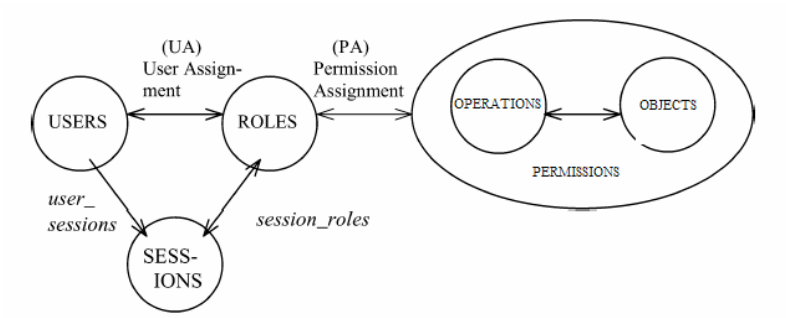

Figure 3 - Core RBAC model.

A permission is a set of allowed operations on objects. An object may be an information element or a Fig. 4. The core RBAC model - adapted from the NIST RBAC standard resource in a computerized system. A role is associated with a set of permissions. Users are assigned a set of (one or more) roles, and hence the access rights for one user are defined by the set of roles currently assigned to that user. In addition the core RBAC model uses the term session. The set of roles assigned to a user through the User Assignment (UA), defines the total set of roles a user can possibly have at any given time this is a static relationship. However, all of these roles are not necessarily activated at one single time. The relationship between users and sessions and sessions and roles indicates that in any given session a subset of the total possible set of roles for a user may be activated. That means that in a session the user may be assigned to the total number of possible roles for him/her, or only a subset. In addition to the core RBAC model, the standard defines rules for separation of duty and the concept of role hierarchies and inheritance relationships. The purpose of separation of duty is to enable constraint enforcement, and maintain role-role relationships and interdependencies. Some roles may contain conflicting permissions, meaning that it should not be possible for the user to be assigned to both roles simultaneously. For this purpose, the RBAC standard defines two mechanisms for separation of duty:

• Static separation of duty (SSD) - enforces constraints on which roles may not at the same time be part of the static set of roles for any user.

• Dynamic separation of Duty (DSD) - enforces constraints on which roles may not be activated simultaneously in one session.

A role hierarchy in the RBAC standard reflects how the level of access rights corresponds to the level of responsibility within an organization. The roles at the top pf the hierarchy contain the most access right and the level of access decreases as one traverse the hierarchy downwards. Role hierarchies also enable the specification of role specialization. A specialization of a role means that one role inherits all the permissions of its predecessor, with added extra permissions. For example one may define a hierarchy where the parent role is named secretary and contains permissions to perform all the tasks that is normally part of a secretary's job, and the roles secretary-sales and secretary-legal inherits all the permissions of the secretary role, but in addition contains permissions that are specific for these kinds of secretaries.

RBAC IN HEALTHCARE

RBAC is considered to be suitable for healthcare as it is designed specifically for commercial applications serving organizations with a high number of employees, but relatively smaller number of job functions (”roles”) and where flexibility and scalability of the access control mechanism is key requirements. Why is it then that access control mechanisms implemented for healthcare systems, claiming to provide role-based access control fail? This question is investigated further in the next section by comparing the main shortcomings of existing access control mechanisms for healthcare identified earlier, with the capabilities of the core RBAC model.

Healthcare access control requirements mapped to RBAC

The main problems with the generalized access control model for healthcare is it's inability to adapt to changes. The model is designed with a static basis and relies on manual procedures for update. In order to add flexibility exception mechanisms have been added. This results in shifting responsibility for information protection and privacy to the users of the system. The generalized access control model is centered around entities (users, patients, organizational units) while in fact work performed in the health sector is process-oriented and dynamic in its nature. The generalized access control model for healthcare consists of two main parts: normal and exception access. We will examine and compare each mechanism separately to se how or if it relates to the RBAC standard, before examining the possibility of combining the two in one RBAC-based model.

Normal access mapped to RBAC

We first compare aspects of the normal access mechanism to RBAC:
• The set of roles is flat - no hierarchy, no inheritance.
• No inter-role dependencies are defined, so separation of duty constraints specified.
• A user may only be assigned to one role. No set of roles, no activation of roles in sessions.
• A list of permissions is associated with each role - on a top level (permissions for access to systems) and in each separate system (permissions for information in a system).
To conclude the normal access control maps to RBAC in its simplest form. It does conform to the principle of relating roles to permissions, and users to role, thereby benefiting from the ease of administration due to only having to assign a pre-defined role to a new user instead of having to construct a new set of permissions for each user.

Exception access mapped to RBAC

Exception access in the generalized model is not realized as roles. The reason for this is the need to be able to restrict these permissions to small set of users, and the roles defines ad part of the normal access control are to few and to broad in their definition thereby encompassing too many users.

A generalized RBAC model for healthcare

Keeping the key concepts and taking advantage of all the features of RBAC as defined in the standard (Ferraiolo, 2001), the generalized access control model based on RBAC for healthcare could be constructed by:
• Adding the concept of role sets - allowing a user to have multiple roles.
• Adding the concept of sessions would allow only a subset of roles to be activated in a session – this could be used to handle a user having several places of work. Only roles related to the currently active (this information could be retrieved from the roster) place of work would be activated in a session.
• Role sets also enable the definition of very specific roles that is only assigned to some users. Instead of adding permissions directly, these users would have an extra role in their set for handling exceptions. This explains how the generalized model can be easily transformed to be entirely based on RBAC. But is this an improvement? Would it be perceived by users as an improved access control mechanism? Is it an improvement for the designers? For security people? Or should the entire model be re-designed?

Limitations of RBAC for healthcare

The main problem of the exception mechanisms in the general model is that they are used for handling a variety of situations. Simply mapping the exception mechanism to RBAC does not solve the real problem. Our goal is to minimize the need for exceptions. In short we need to extend the *normal* part of the generalized access control model to be able to handle situations like:
1. Unplanned admissions and transfers.
2. Post-discharge and pre-treatment access to information.
3. Referrals.
4. Second opinions.
5. Laboratory tests.
Situation number 1 and 2 are examples of processes taking unexpected turns. Handling this is not a matter of roles. However, contextual information (for example the current geographical location of a patient, or the bed he/she is in, health care personnel present etc.) could possibly be used in combination with roles to handle these situations. Contextual RBAC has been discussed in several papers (Thomas, 1997), (Georgiadis, Mavridis, Pangalos and Thomas, 2001), (Kumar, Karnik and Chafle, 2002), (Wilikens, Feriti and Masera, 2002), but some work remains on figuring out what constitutes relevant contextual information for access control in healthcare. Also, the main problem here is that healthcare is process oriented, and so should the access control mechanism be. Exploring process-oriented access control remains to be done. Situation number 3, 4 and 5 on the other hand are examples of events. This could possibly be achieved through delegation and revocation of roles as proposed in (Na and Cheon, 2000). Delegation of roles is triggered by the user who delegates some of his or her permissions through delegating a role to another user - or possibly to another role. Revocation of delegated roles can be either based on a timestamp of validity, automatic when a task is completed or user-initiated. The use of role delegation shifts some responsibility for preserving the access control policy onto the system users. However, there is the possibility to place constraints on role delegation, to regulate which roles may be delegated to who by whom

thereby minimizing the probability and possibility of errors. The who and whom mentioned here are roles - not individual users. If an access control mechanism is based on RBAC all access should be based on the notion of roles to avoid unnecessary and risk-increasing complexity.

Conclusion and future work

In this paper we have discussed requirements for access control in healthcare in relation to implemented access control mechanisms in existing healthcare systems. It is clear that the implemented mechanisms are not ideal. The strongest indication of this is the use of exceptions in addition to the normal access control mechanism. The implemented mechanisms are based on a combination of role- and identity-based access control. Redesigning to a pure role-based model does not immediately help minimizing the need for exceptions. The main problem to be solved is to move from an access control model based on static properties to a model that adheres to the dynamic nature of healthcare organizations. In other words we need to design an access control model that is process-oriented and is able to adapt to unplanned events. This model may or may not be based on roles. That remains to be investigated. We intend to continue research on how to construct an access control model better suited for healthcare applications. To do so we will work on identifying and formalizing processes that should be supported, and see how these may be incorporated into a process-oriented access control model.

Bibliography

Lillian Røstad received her master's degree on the topic "Securing healthcare information in distributed systems" from the Norwegian University of Science and Technology in 2002. Currently she is working part-time as a research scientist at SINTEF ICT, as well as working on a PhD. The topic for the PhD-thesis is: "Dynamic and context-sensitive access control for distributed healthcare applications".

References

David Ferraiolo and Richard Kuhn (1992), "Role-Based Access Control", *Proceedings of 15th National Computer Security Conference*, USA.

Ferraiolo et. al (2001) , "Proposed NIST Standard for Role-Based Access Control", *ACM Transactions on Information and System Security*, Vol. 4, No. 3, pp. 224-274.

Mark Evered and Serge Bgeholz (2004), "A Case Study in Access Control Requirements for a Health Information System", *Proceedings of Australasian Information Security Workshop 2004*

Marc Wilikiens, Simone Feriti, Alberto Sanna and Marcelo Masera (2002), "A Context-Related Authorization and Access Control Method ased on RBCA: A case study from the health care domain", *Proceedings of ACM Symposium on Access Control Models and Technologies*.

Wolfgang Essmayr, Stefan Probst and Edgar Weippl (2004), "Role-Based Access Controls: Status, Disseminatio ,and Prospects for Generic Security Mechanisms", *Electronic Commerce Research*, No. 4/2004, pp. 127-156.

Rattapoom Tuchinda (2002), "Access Control Mechanism for Intelligent Environments", *Bitstream: The MIT Journal of EECS Student Research. Cambridge, MA*.

Roshan K. Thomas (1997), "Team-based Access Control (TMAC): A Primitive for Applying Role-based Access Controls in Collaborative Environments", *Proceedings of ACM Symposium on Role-Based Access Control*.

Christos K. Georgiadis, Ioannis Mavridis, George Pangalos and Roshan K. Thomas (2001), "Flexible Team-Based Access Control Using Contexts", *Proceedings of ACM Symposium on Access Control Models and Technologies*.

Arun Kumar, Neeran Karnik and Girish Chafle (2002), "Context Sensitivity in Role-based Access Control", *ACM SIGOPS Operating Systems Review*, Vol. 36 , No. 3, July 2002, pp. 53-66.

Marc Wilikens, Simone Feriti, and Marcelo Masera (2002), "A Context-Related Authorization and Access Control Method Based on RBAC: A case study from the health care domain", *Proceedings of ACM Symposium on Access Control Models and Technologies*.

SangYeob Na and SuhHyun Cheon (2000), "Role Delegation in Role-Based Access Control", *Proceedings of ACMWorkshop on Role-Based Access Control*.

Guttorm Sindre and Andreas L. Opdahl (2005), "Eliciting Security Requirements with Misuse Cases", *Requirements Engineering Journal*, Vol. 10, No. 1, January 2005, pp. 34-44.

Paper B

A Study of Access Control Requirements for Healthcare Systems Based on Audit Trails from Access Logs

A Study of Access Control Requirements for Healthcare Systems Based on Audit Trails from Access Logs

Lillian Røstad and Ole Edsberg
Norwegian University of Science and Technology (NTNU)
Department of Computer and Information Science
Trondheim, Norway
{lilliaro,edsberg}@idi.ntnu.no

Abstract

In healthcare, role-based access control systems are often extended with exception mechanisms to ensure access to needed information even when the needs don't follow the expected patterns. Exception mechanisms increase the threats to patient privacy, and therefore their use should be limited and subject to auditing. We have studied access logs from a hospital EPR system with extensive use of exception-based access control. We found that the uses of the exception mechanisms were too frequent and widespread to be considered exceptions. The huge size of the log and the use of predefined or uninformative reasons for access make it infeasible to audit the log for misuse. The informative reasons that were given provided starting points for requirements on how the usage needs should be accomplished without exception-based access. With more structured and fine-grained logging, analysis of access logs could be a very useful tool for learning how to reduce the need for exception-based access.

1 Introduction

Security is a key concern for healthcare systems that contain sensitive data, like the Electronic Patient Record (EPR). Access control is at the heart of this concern. While healthcare personnel need access to the right information at the right time to provide the best possible care, it is also important to ensure patient privacy.

Over the last few years, we have seen a development in access control research towards more dynamic, workflow-based and user-centered models [1]. However, the state of the art in existing healthcare systems appears to be the traditional Role-Based Access Control (RBAC) model [2], where roles correspond to job functions and administration is centralized. These systems are not well-suited for handling unplanned and dynamic events like patients being transferred between wards, doctors asking for second opinions from colleagues or simply unplanned patient arrivals. Consequently most such systems have exception mechanisms in place in addition to the normal role-based access control for handling these situations. Use of these exception mechanisms typically triggers additional logging of the user's actions. Including these mechanisms makes the systems much more convenient to use. However, from a security viewpoint the use of exceptions leads to added complexity and a need to perform regular auditing to ensure that the mechanism is not misused. With an exception mechanism in place that allows the users to override the normal access control mechanism, technical measures alone cannot ensure privacy and security. This increases the need for manual control mechanisms and awareness training for users to limit the use of the exception mechanisms. However, studying how these access control mechanisms are used - in what situations, to cover what needs - may teach us something about how normal access control mechanisms should be changed to better suit the needs of the users, thereby eliminating or at least minimizing the use of exception mechanisms. Also, it is interesting to investigate if the audit logs contain the necessary information to trace any misuse of such exception mechanisms, or if not - what information is lacking.

In this paper we will examine access logs from an installation of DocuLive EPR[1], a system with extensive use of exception-based access control. Doculive EPR is used by many of the largest hospitals in Norway. We have pulled information from the access logs from all eight hospitals in the Central Norway Health Region (CNHR). The aim of this work is to investigate if the audit trails may uncover information about the real user needs that will be helpful in designing better access control mechanisms for healthcare and also to examine if the logs contain the information needed

[1] DocuLive is a product of Siemens Medical Solutions

Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC'06)
0-7695-2716-7/06 $20.00 © 2006 IEEE

to uncover misuse. Additionally we aim to explore if any of the principles set forward in access control research in recent years may be applied to create better-suited access control mechanisms for healthcare systems.

2 Related work

To our knowledge there has been no previous work published on investigating audit trails from EPR systems to extract access control requirements for healthcare systems. However some work has been done on eliciting access control requirements for healthcare systems by other means. Evered and Bögeholz in 2004 published a paper [3] describing how they performed a detailed case study on a small aged-care facility in Australia that at the time of study only used paper-based records. The study illustrates that even for such a small example, the access control requirements are very complex. In a short (one page) paper from 1998 [4] Beznosov discusses requirements for access control in the US healthcare domain and states that it should be based on role, affiliation, location, time and relationship. It is however not clear from the one-page paper what these conclusions are based on. In a classic paper [5] from 1996 R. J. Anderson presents a general security policy model for clinical information systems, which includes access control. He bases the motivation for this policy on a number of identified threats towards healthcare systems. Based on his experience and involvement in international EPR architecture and security standards, Blobel in 2004 [6] published a paper describing a set of models for authorization and access control in healthcare systems.

3 The subject of study

Norway is divided into five health regions: north, south, east, west, and central. Each region has a regional health authority and several health enterprises. Each health enterprise encompasses one or more hospitals, and together the health enterprises in one region encompass all hospitals within the region. In the Central Norway Health Region (CNHR), which was the object of this study, there are four health enterprises and eight hospitals. All of these hospitals use DocuLive EPR. Norwegian laws prohibit sharing of medical records between health enterprises. Medical information may be transferred based on a specific request, but not shared in real-time, e.g. through a common EPR-system. As Figure 1 shows there are therefore separate installations of the EPR-system for each hospital. However, there is one common organization, CNHR IT, which is responsible for the daily operation and maintenance of the EPR-systems for all hospitals in the region. Because they all use the same EPR-system, DocuLive, it is possible to extract and compare log data across hospitals.

Figure 1 also illustrates how the EPR system for one hospital is divided into three domains: somatic, psychiatry and child and youth psychiatry. Information in the patient record is assigned to a domain. Domains are used to protect information that is considered ”extra sensitive”. This means that a user working on a ward in the somatic domain does not have access to parts of a patient’s EPR that belong to any of the other two domains - even if the patient currently is at this user’s ward. Only users working in psychiatry or child and youth psychiatry can access parts of the EPR that are assigned to these domains.

In DocuLive access decisions are based on a user’s *role* (e.g. doctor, nurse, secretary), current place of work (ward) and the type of information being accessed. The role determines which documents in the EPR a user is allowed to access. At any given time a user has access permissions according to his or her role for the patients that are currently registered at the ward where he or she works. Note that a user may be assigned to several roles and places of work. In addition, there are two exception mechanisms for access:

- Actualization - allows a user to open the EPR of a patient that he/she does not have access to through the normal access control mechanism. The user is granted access to the EPR as though the patient was registered at the ward where he/she works. The permission to use the actualization mechanism is not part of a user’s role, but is granted on an individual basis. When using actualization the user has to provide a reason for doing so, and the action is recorded in a separate log for use of actualization and emergency access. The EPR is then opened for a specific time period, which depends on the reason provided. In CNHR there are currently eight predefined reasons for using actualization which are shown in Table 1 with corresponding time intervals. There is also the option for entering a self-defined reason and time interval. Actualization is also used as an automatic mechanism by the system for opening EPRs for users who are assigned an approval-task (signing) for documents in the EPR and for opening the EPR of patients who are scheduled to arrive at the hospital soon, but have not been admitted yet. The time-period for automatic actualization is set to 7 days.

- Emergency access - allows a user to open a single document in a patient’s EPR that he/she does not have access to through the normal access control mechanism. The emergency mechanism is stricter than actualization in that it has to be used on every single document that the user wants to open. In CNHR only some of the hospitals use emergency access - most make due with only actualization. However - where in use - emergency access is used to access EPR documents across

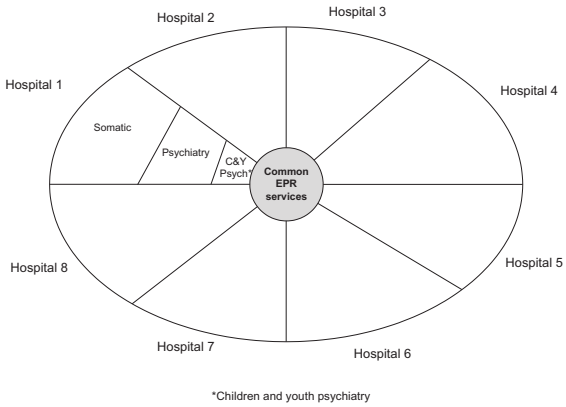

Figure 1. EPR Hospital model

domains within one hospital. That is: some use it as a way for users in the somatic domain to access information in the psychiatry and child and youth psychiatry domains. As for actualization, when using emergency access control the user has to provide a reason and the action is recorded in the same log as use of actualization log. Note that there are no predefined reasons for using emergency access; the user always has to manually provide a reason. Also note that the time interval where the document remains accessible after using emergency access is firm. In CNHR this time interval is set to 10 hours. Not all documents in the EPR are accessible through emergency access, only those specifically labeled so, and only some users have the permission to use emergency access. Emergency access is assigned to users much in the same way as roles - meaning that the permission to use emergency access is linked to a ward or hospital.

4 Methods and materials

In this study we collected access log data from the EPR-system from all eight hospitals in CNHR for one month (March 2006). There are two separate logs:

- Access log - every time a document is opened an entry is created in the access log containing information about the user, the patient and the document being accessed.
- Actualization and emergency log - an entry is created in this log whenever an EPR is opened using actualization or a document is opened using emergency access. This record also contains information about the provided reason and time interval.

Note that it is only the action of actualization or emergency access that is recorded in a separate log. Any subsequent use of the EPR within the time interval is recorded in the normal access log. Therefore we had to extract and combine information from the two logs to get a complete view of use of EPRs within an actualization or emergency access period.

The IT-unit in CNHR was very helpful in creating anonymized versions of the logs - removing names of users and patients and replacing with anonymous, but unique indexes. In addition to the log-extracts, we also collected an anonymized listing of users in the region including their assigned access permissions. The log-extracts we received consisted of:

- All records:
  - Anonymous user ID
  - Users's place of work - hospital and ward
  - Anonymous patient ID
  - Patient location - hospital and ward
- Only in records from access log:
  - Time stamp
  - Document ID
  - Document type
  - Document code
- Only in records from actualization/emergency log:

Reason	*Time(hours)*
Healthcare - provide/plan/consider	48
User support	3
Research project	24
Write/complete EPR documents	48
Scan	2
Quality assurance - administrative/professional	48
Obliteration/editing/deletion/blocking/merging	1
Control committee	24
Other (self-defined)	-

Table 1. Predefined reasons and time intervals for use of actualization

- Start time
- End time
- Reason

4.1 Research questions

After reviewing the type of information available, we constructed a set of research questions to structure our work. The questions were selected to collect information that we hope will contribute to uncovering access control requirements for healthcare systems. The questions we aim to investigate and hopefully answer are:

- Q1: Is actualization/emergency access used sufficiently infrequent to be considered an exception?
- Q2: Which users (role) use actualization/emergency access the most?
- Q3: Which wards use actualization the most?
- Q4: What reasons are provided for using actualization/emergency access?
- Q5: What kind of information is most often accessed using actualization/emergency access?
- Q6: What information should be recorded in access logs to be able to investigate misuse?

5 Results

5.1 Some basic numbers

Table 2 contains an overview of basic user data: how many users in total, how many have actualization permission and how many have emergency access permission. The table shows that out of a total of 16723 DocuLive users in the health region, 74% have been assigned the permission to actualize EPRs, but only 0,25% have the permission to use emergency access. Note that emergency access is only used by two of the hospitals in the region. The others use only actualization.

	Count	%
No. users actualization perm.	12 298	74
No. users emergency access perm.	41	0.25
No. DocuLive users (total)	16 723	100

Table 2. Number of users and permissions

	Count	%
Actualized EPRs	54 095	54
EPRs accessed using emergency	67	0.07
Number of patients (total)	99 352	100

Table 3. Overall use of actualization

5.2 Q1: Is actualization/emergency access used sufficiently infrequent to be considered an exception?

As Table 3 illustrates, in March 2006 a total of 99 352 distinct patients were in contact (i.e. their EPR's were accessed in some way) with the hospitals in the region. Of these patients 54% had their EPR accessed using actualization. This fact combined with the fact that 74% of all users are assigned the permission to use actualization indicate that use of actualization is indeed not an exception. This motivates further investigations as to how actualization is used.

Emergency access is, by comparison, only used 67 times and only very few users are assigned this permission. The numbers are therefore so low that they are difficult to use as a basis for any reasoning. We will therefore focus on the use of actualization, and only return to emergency access in the discussion - as in the true meaning of it's name this mechanism will probably always need to be present. However the way this mechanism is used in the hospitals in this study, as we have explained earlier, does not really reflect on the name *emergency* access.

Table 4 illustrates the proportions of use of actualization and emergency access compared to the total number of accesses in EPR. One access corresponds to opening of one

	Count	%
Accesses using actualization	297 742	17
Accesses using emergency	67	0.004
Total number of accesses	1 794 153	100

Table 4. Number of accesses in total and using actualization or emergency access

EPR or a folder or document inside an EPR. Based on these numbers we find that 17% of the accesses are based on actualization. On average there were registered 2.31 accesses in an EPR within one actualization period.

5.3 Q2: Which users (roles) use actualization access the most?

Table 5 presents an overview of defined roles, number of users assigned to this role in total, percentage of users within each role who are assigned actualization permission, and percentage of users within each role who have used actualization in the period. Note that we have removed the roles where no users are assigned actualization permission, which were a total of three roles: perfusionist, dental health secretary and acupuncturist.

If we assume that the percentage of actualization assignment for one role reflects the current perceived need or requirement for use of this functionality for users within this role (and possibly also a level of trust in users within this role) - then it is interesting to take a closer look at the differences between actualization assignment and use. DocuLive has been in use since 1998 (from 2002 for the entire region) so it is reasonable to assume that the distribution of roles and permissions are fairly stable now. We may then assume that the percentage of use of actualization reflects the actual needs or requirements of users within a role. If we examine Table 5 more closely we see that on average the actual percentage of use of actualization is significantly lower than the percentage of assignment of actualization. This may lead to the interpretation that actualization is in fact assigned to many users that do not need this functionality - at least not on a regular basis. For instance it seems to be the rule that all doctors should have permission to actualize - but only 52% of doctors did in fact need to do so within this period. Of the nurses, who represent the largest group of users by far, only 22% used actalization - while 61% has the permission to do so. Thus it would be interesting to further investigate who of these users, in what situations actually do require the functionality provided by actualization. However the log-data does not provide sufficient information, and would have to be supplemented with other information - possibly from questionaires, interviews, observations etc.

Role	*Count*	*%act*	*%use*
Nurse	9 234	61	22
Doctor	2 957	99	52
Health secretary	1 934	97	51
Enrolled nurse	799	31	5
Physiotherapist	411	93	52
Midwife	382	83	17
Psychologist	196	99	57
Ergonomist	150	84	38
Social worker	128	95	59
Educationist	101	96	47
Consultant	80	56	30
Social educator	79	84	28
blank/incompr.	48	75	25
Radiation therapist	34	100	44
Audiometrist	31	97	65
Radiologist	26	96	35
Speech therapist	25	80	40
Nutritionist	21	100	71
Bioengineer	16	94	6
Activator	15	67	7
Pharmacist	9	11	0
Welfare worker	9	44	11
Orthopaedy engineer	7	100	14
Dentist	7	100	14
Genetic advisor	4	100	100
Orthoptist	4	100	100
Occupational hygienist	2	100	0
Optician	2	100	100
Child welfare consultant	2	100	50
Ambulance personnel	1	100	0
Dental mechanic	1	100	100

Table 5. Overview of roles with % assigned and use of actualization permission.

$Ward$	$Users$	$\%act$	$\%use$
Medical ward (18)	2 834	86.9	49.8
Surgical ward (21)	2150	75.2	33.2
Anaesthesia ward (8)	629	99.5	30.3
Emergency ward (10)	482	71.1	27.6
Out-patient clinics (43)	473	99.7	62.6

Table 6. Overview of users employed at ward types with % assigned and use of actualization permission. The number of wards of a type is given in parentheses. Wards that were not covered by a major type were excluded.

5.4 Q3: Which wards use actualization the most?

From Table 6 we can see that actualization is used rather frequently at the medical ward[2]. According to [7], 90-95% of the patients who are admitted to the medical ward need immediate help. Only 5-10% are planned patient encounters. As such, the high number of actualizations for this ward is unsurprising. It is interesting to note that for the surgical, anaesthesia and emergency wards the percentage of users assigned actualization permission is significantly higher than the percentage of actual use. Out-patient clinics represent the wards with the highest count of actualization use. This is probably due to the fact that patients are not admitted to these wards, they are just there for a short time in the day, and as such it would make sense to have an access mechanism in place to handle this.

5.5 Q4: What reasons are provided for using actualization/emergency access?

Table 7 shows that out of all uses of the actualization functionality, a self-defined reason was only entered in 1.76 % of the cases. We investigated this number further and found that out of all the users who had used actualization functionality in the period only 8% had, at least one time, provided a self-defined reason for doing so. Actualization was used a total of 133 918 times, and a self-defined reason was only provided in 2 357 of these actualization occurrences. Several reasons were provided multiple times, so these 2 357 reasons again map to 730 unique reasons.

These numbers tell us a couple of interesting things. First of all: the availability of predefined reasons means less specific information about why actualization was used. The predefined reasons are so broadly defined that they convey very little information about the user's needs. What we can see is that signing information in the EPR is a common task, that should be included in the normal access control regime.

Although the 730 unique reasons provided are too few to base any quantitative conclusions on, we nevertheless decided to take a closer look, working from the hypotheses that when users took the trouble to manually enter a reason they felt that the predefined reasons did not apply to their situation or did not describe their need accurately. If some of these manually entered reasons are recurring then this implies a need shared by several users. 730 entries are so few that it was possible to examine them one by one and attempt manual classification to see if we could create categories of recurring reasons or types of reasons. We found that the most commonly provided reasons are:

- Out-patient clinic patient encounters.
- Physician referrals.
- Hand over patient information to other hospital/health personnel on request.
- Request for information from a patient or next of kin.
- Release information to other external entity: insurance, legal, complaints.
- Patient not registered correctly in admin system (results in access denied, even though patient is physically present at ward).

As such, these should be considered as candidates for inclusion in the normal access control regime and constitute access control requirements that are not fulfilled.

5.6 Q5: What kind of information is most often accessed using actualization/emergency access?

Table 8 shows how the rate of actualization usage varies with the document category. The high rate of the top entry might be explained by the fact that it includes second opinions, where the provider of the opinion might often need access to the patient record across ward boundaries. The same type of need could also explain the high rate of the second entry, which covers reports from physiotherapists, psychologists and other non-physician specialists. The relatively low rate of the nursing-related entries might be due to the fact that nurses mostly work with the patients admitted to their working ward.

We also see that image-related lab results have almost twice the actualization rate of tissue and fluid-related lab results, perhaps because specialists from other wards are often called upon to interpret images.

With a more fine-grained and well-structured category hierarchy, we might have been able to construct a more

[2] The medical ward mainly offers internal medicinal treatment.

Reason	%
Healthcare - provide/plan/consider	32.87
User support	0.03
Research project	1.64
Write/complete EPR documents	41.27
Scan	2.02
Quality assurance - administrative/professional	2.83
Obliteration/editing/deletion/blocking/merging	0.88
Control committee	0.11
Automatic for signing	10.33
Automatic from planned patient list	6.26
Sum predefined and automatic reasons	98.24
Manually provided, self-defined reasons	1.76

Table 7. Actualization reasons: usage in percent.

Documentcategories	*Totalaccesses*	*%withactualization*
External correspondence	218381	32.80
Reports from other disciplines	60431	25.81
Lab results: Image diagnostics	24438	23.64
Physician's journal	503496	23.09
Declarations etc	13664	19.96
Summaries, not further classified	83810	18.49
Observation and treatment	22883	18.28
Lab results: Tissue and fluids	69046	13.09
Own discharge summaries	106968	12.50
Lab results: Organ function	26342	12.04
Nurses' summaries	10688	7.81
Nurses' documentation	482919	6.37
Other	154326	5.51
Patient orientation	12005	5.30

Table 8. Percentage of accesses performed within actualization periods, for different categories, as classified in the EPR system. The category *Other* collects accesses to documents without category or in categories with fewer than 10000 accesses.

Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC'06)
0-7695-2716-7/06 $20.00 © 2006 IEEE

informative chart of actualization rates. If a decision was made to reduce the usage of actualization, such a chart could be used to detect the best possibilities for reduction.

5.7 Q6: What information should be recorded in access logs to be able to investigate misuse?

Exception access in some form will always have to be present in healthcare systems to handle emergencies. Therefore it is important to have sufficient and usable mechanisms to trace any misuse.

It is clear from the work presented here that the DocuLive logs do not present sufficient information to effectively investigate suspicions of misuse. We had to combine data from two separate logs and the user database to be able to do this work, and still we believe that more information is required. The main shortcoming is the predefined reasons for using actualization that mask the real intent.

For an audit trail to be usable it should:

- be available through a usable interface for the administrators, and
- contain sufficiently detailed information to get a picture of what has happened.

6 Discussion

The system under study here in many ways conforms to the ideas of *optimistic security* put forward in [8]. However, this study illustrates how difficult it is to trace events in such a system. Being able to trace events is essential to provide adequate security for systems containing sensitive health information. Therefore we believe that healthcare systems require a stricter form of access control, where the usage of exceptions is minimized. Having examined the audit logs we have found some recurring events fulfilled with actualization, that should be candidates for inclusion in requirements for an access control model that is better suited for the real needs of the users. Thus this should aid in minimizing the use of actualization.

We would also like to point out that when exception mechanisms are introduced, it is important to have regulations on who should be assigned this permission and to ensure that these regulations are followed. It should be easy to obtain an overview over which users, or roles, have the permission to use exception mechanisms. Minimizing risk includes minimizing the user base that has the potential for exploiting exception mechanisms.

Based on this study, we have not been able to conclude on a firm set of requirements for access control in healthcare systems. However, we have identified some initial requirements that we intend to explore further. Most of what we have seen indicates the need for a more dynamic and user-controlled access control solution. We believe that RBAC should be the foundation, but with added ability for handling dynamic events, workflow and collaboration. Several papers, including [9] [10] [11] have been written on the concept of *role delegation* which allows a user to delegate his/her role to another user. This may be used as a mechanism to handle referrals, second opinions and transfer of patients. To be able to do this we should introduce the notion of health personnel-patient *relationship*, meaning that they are linked by something more than just a common ward.

We also think the notion of Team-Access Control [12] centered around a cooperating team seems promising. Based on our findings of provided reason, we believe that the notion of *tasks* and related *responsibilities and duties* provides a promising platform for access control decisions in healthcare systems.

7 Conclusion and future work

Although we have been able to identify some requirements, or initial requirements, in this study, more work needs to be done. We intend to continue our investigation by supplementing with data from other systems from the same period (including admission/discharge dates) to see when actualization is primarily used. In addition we hope also to be able to observe healthcare personnel's information needs in situations where common tasks need to be performed. For that purpose, interviews are another possibility we hope to explore.

Acknowledgements

The authors would like to thank the people at Central Norway Health Region who helped make this study possible. We would also like to thank our advisors Øystein Nytrø and Svein Johan Knapskog, as well as our fellow PhD-student Thomas Brox Røst, for valuable input and help.

References

[1] W. Tolone, G.-J. Ahn, T. Pai, and S.-P. Hong. Access control in collaborative systems. *ACM Comput. Surv.*, 37(1):29–41, 2005.

[2] D. F. Ferraiolo, D. R. Kuhn, and R. Chandramouli. *Role-Based Access Control*. Computer Security Series. Artech House Publishers, Boston, 1 edition, 2003. ISBN: 1580533701.

[3] M. Evered and S. Bögeholz. *A case study in access control requirements for a Health Information System*. Proceedings of the second workshop on Australasian information security, Data Mining and Web

Intelligence, and Software Internationalisation - Volume 32. Australian Computer Society, Inc., Dunedin, New Zealand, 2004.

[4] K. Beznosov. *Requirements for access control: US Healthcare domain*. Proceedings of the third ACM workshop on Role-based access control. ACM Press, Fairfax, Virginia, United States, 1998. ISBN: 1581131135.

[5] R. J. Anderson. *A security policy model for clinical information systems*. Proceedings of the 1996 IEEE Symposium on Security and Privacy. IEEE Computer Society, 1996.

[6] B. Blobel. Authorisation and access control for electronic health record systems. *International Journal of Medical Informatics*, 73(3):251–257, 2004. ISSN: 1386-5056.

[7] St. Olavs Hospital - medical ward. URL: http://www.stolav.no/stolav/Virksomhet/behandling/medisin/index.htm. Last accessed: May 28th 2006.

[8] D. Povey. *Optimistic security: a new access control paradigm*. Proceedings of the 1999 workshop on New security paradigms. ACM Press, Caledon Hills, Ontario, Canada, 2000. ISBN: 1581131496.

[9] L. Zhang, G.-J. Ahn, and B.-T. Chu. *A role-based delegation framework for healthcare information systems*. Proceedings of the seventh ACM symposium on Access control models and technologies. ACM Press, Monterey, California, USA, 2002.

[10] S. Na and S. Cheon. *Role delegation in role-based access control*. Proceedings of the fifth ACM workshop on Role-based access control. ACM Press, Berlin, Germany, 2000.

[11] E. Barka and R. Sandhu. Framework for role-based delegation models. In *Annual Computer Security Applications Conference (ACSAC)*, pages 168–176, 2000.

[12] R. K. Thomas. *Team-based access control (TMAC): a primitive for applying role-based access controls in collaborative environments*. Proceedings of the second ACM workshop on Role-based access control. ACM Press, Fairfax, Virginia, United States, 1997.

Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC'06)
0-7695-2716-7/06 $20.00 © 2006 IEEE

Paper C

An extended misuse case notation: Including vulnerabilities and the insider threat

An extended misuse case notation: Including vulnerabilities and the insider threat

Lillian Røstad[1]

Norwegian University of Science and Technology, Trondheim, Norway
Lillian.Rostad@idi.ntnu.no

Abstract. Misuse cases are a useful technique for eliciting and modelling security requirements and threats. In addition they may be very useful in a risk analysis process, particularly as part of the system development process. The original misuse case notation adds inverted use cases to model threats and inverted actors to represent attackers. However, an attack is usually performed by exploiting a vulnerability in a system and it would be useful to be able to represent vulnerable functions in a model. In addition, it should be possible to discern between insiders and outside attackers in a model, as they have very different abilities and potential for attacking a system. This paper therefore proposes an extended misuse case notation that includes the ability to represent vulnerabilities and the insider threat, and discusses the use of this extended notation in the system development and risk analysis processes.

1 Introduction

Security is being increasingly recognized as an important quality of IT-systems. Much of the reason for this can be explained by the evolution of IT-systems towards what Gary McGraw in [13] defines as the *trinity of trouble*: connectivity, extensibility and complexity. While these three properties typically improves the possibilities of what a system can do, they also significantly increases the risks. Being secure means having control and being able to keep the bad guys out - but the more complex a system is the harder it is to manage, and the possibility of third-party extensions only adds to the complexity. Connectivity is seductive as it greatly increases the potential use of a system, but it also greatly increases the number of attackers that can have a go at breaking into or otherwise harm the system. In some systems, like health care, defence and banking, security has always been considered an important property. But as the system's operational environment changes, so does the threat scenarios and need for defence mechanisms. Where isolation previously has been considered an appropriate defence, this is no longer an option.

An excellent example of this, and the original motivation for the work presented here, is access control in healthcare systems. In healthcare systems protecting the patient's privacy is a major concern - however it always has to be balanced against the need for access to information to make sound medical decisions and provide the best possible care. The current state-of-the art is Role-Based

Access Control (RBAC) [8] and a role in existing systems is typically a rather static structure combined of a user's profession (doctor, nurse etc), place of work (ward) and where the patient is currently admitted (ward). There is currently a move towards making the systems more dynamic and user centric and enabling information sharing. As patients are able to select hospital or place of care more freely there is a need to be able to make a patient's medical information available to those providing care. This significantly adds to the complexity and changes the requirements for the access control mechanisms - static structures are no longer sufficient. Also, most existing systems include mechanisms that allow a user to override the access control mechanism in emergency situations. In such situations there is no time to register the patient at the correct ward to enable the normal access control mechanism to function. Emergency access control effectively constitutes a vulnerability in the system that may be exploited by insiders - that is; legitimate system users that may misuse the functionality. As systems become connected the user bases grow, thereby increasing the potential risk for exploitation.

To be able to design secure solutions in a changing threat scenario one needs to be able to perform risk analysis [21] based on system requirements and design [13]. UML use cases [1] have become a widely used technique for elicitation of functional requirements [7] when designing software systems. One of the main advantages of use cases is that they are easy to understand with only limited introduction to the notation, and therefore are a very well-suited tool for communicating and discussing requirements with system stakeholders. A use case model illustrates required usage of a system - i.e. expected functionality. In risk analysis it is equally important how one should *not* be able to use a system - i.e. potential threats and exploitation. Misuse cases [18] have been proposed as an approach to identifying threats and required countermeasures. The notation is very simple and complements the UML use case notation. However, the usability of the notation or the ability to give a more complete risk overview could be significantly improved by adding some minor extensions enabling the specification of vulnerabilities and the insider threat in misuse case models. The remainder of this paper presents such an extended misuse case notation and discusses potential use in system development and risk analysis.

2 Related work

The notation proposed here builds upon work done on how to utilize use cases as a tool for eliciting and modelling security requirements. John McDermott [11] and Chris Fox [12] used the term *abuse cases* in their approach where they explored how threats and countermeasures could be expressed using the standard UML use case notation. In their approach they kept the abuse cases in separate models.

Later, in a series of papers [15], [16], [17], [19], [14], [18], Guttorm Sindre and Andreas L. Opdahl have proposed, and elaborated on, the concept of *misuse cases* including both graphic and textual description. Misuse cases [18] extends

the UML use case notation by adding inverted use cases to model misuse and inverted actors to model attackers. Sindre and Opdahl [18] define *misuse cases* and *misusers* as:

- *Misuse case* - a sequence of actions, including variants, that a system or other entity can perform, interacting with misusers of the entity and causing harm to some stakeholder if the sequence is allowed to complete.
- *Misuser* - an actor that initiates misuse cases, either intentionally or inadvertently.

Misuse cases are created by extending a use case model and thus provide the ability to regard system functions and possible attacks in one coherent view. In the initial work on misuse cases two additional relationships were defined [15]: *prevents* and *detects* and it was pointed out that the UML use case relationships *include* and *extend* may also be used to connect misuse cases. They also pointed out that the *include*-relationship may be used between a misuse case and use case to illustrate that an attack utilizes system functionality. This in fact corresponds to exploiting a vulnerability, but they did not provide a tailored notation for this.

Ian Alexander has written several papers discussing misuse cases as a tool [6] [5] and experiences from application of misuse cases [2]. He has also discussed misuse cases in relation to goal-oriented requirements engineering [3] [4]. In this case Alexander stays true to the graphic notation of inverted use cases proposed by Sindre and Opdahl, but he defines four different relationships: *threatens*, *mitigates*, *aggravates* and *conflicts with*. It is interesting to note that in their latest (at the time of writing this paper) [18] publication on misuse cases, Sindre and Opdahl have refined the relationships in the misuse case notation adopting *threaten* and *mitigate* as suggested by Ian Alexander. By their definition a *use case mitigates misuse case* and *misuse case threaten use case*. Exchanging the *prevents* and *detects* with the softer *mitigate* makes sense as it is unlikely that any countermeasure applied will entirely eliminate a threat.

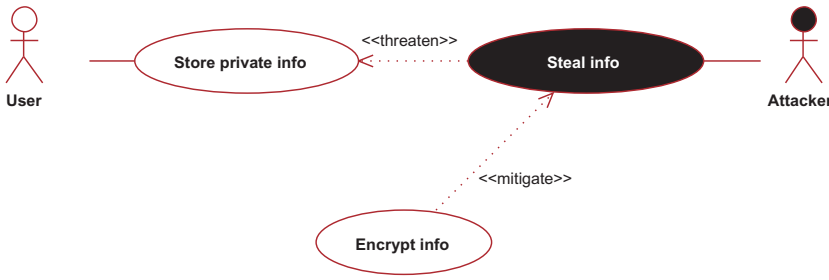

Fig. 1. Simple use and misuse case illustrating the notation

Donald G. Firesmith has discussed the concept of *security use cases* [9] where a security use case represents functionality needed to protect a systems assets

from identified threats. The idea of security use cases as a way of representing specific security functionality, or countermeasures, has been adopted by Sindre and Opdahl [18] and linked directly to the *mitigate* relationships. Security uses cases have not been given a specific graphical notation, but are represented as ordinary use cases in the models.

Figure 1 depicts a very simple misuse case that illustrates the current notation. In this figure *encrypt info* is a security use case added to protect against the threat (*steal info*) identified as a potential misuse case.

3 Extended misuse case notation

This paper proposes an extended misuse case notation to enable visualisation of vulnerabilities and the insider threat. The original misuse case notation only defines outside attackers [18]. However, inside attackers also pose a serious threat. An insider, to an organisation or a system, usually has much easier access to a system and thereby may perform other attacks and exploit other weaknesses than an outside attacker. As such it is useful to be able to model insiders as a separate actor type in order to get a comprehensive and complete overview of possible threats and attacks. In the original misuse case notation misuse cases are linked directly to use cases that they threaten. In other words attacks are linked to system functionality that may be disabled or otherwise damaged as a consequence of a successful attack. However it would be useful to be able to visualize what vulnerabilities are exploited to perform that attack. Threats towards a system may only be realized in an attack if the system contains vulnerabilities that can be exploited. It is important to be able to illustrate vulnerabilities to be able to identify all possible threats and attacks. We define an *insider* and a *vulnerability* as:

- *Insider* - a misuser that is also member of an authorized group for the entity being attacked - e.g. an authorized user of a system, a member of the development team, an employee of an organization.
- *Vulnerability* - a weakness that may be exploited by misusers.

Figure 2 presents a combined overview of the notation for use cases and extended misuse cases. In addition to actors representing insiders and misuse cases representing vulnerabilities an additional relationship *exploit* is defined. The *exploit* relationship is used to link a threat to a vulnerability. Insiders and vulnerabilities have been given the same grey colour in this extended notation. This choice of colour indicates that both represent weaknesses in a system that may or may not be exploited. Either way it is important to have knowledge about the weak spots of a system as this constitutes the systems *attack surface* that may be exploited. The remainder of this section presents examples of how to use the extended notation. We have included three examples that illustrates different situations and systems where the notation will be useful.

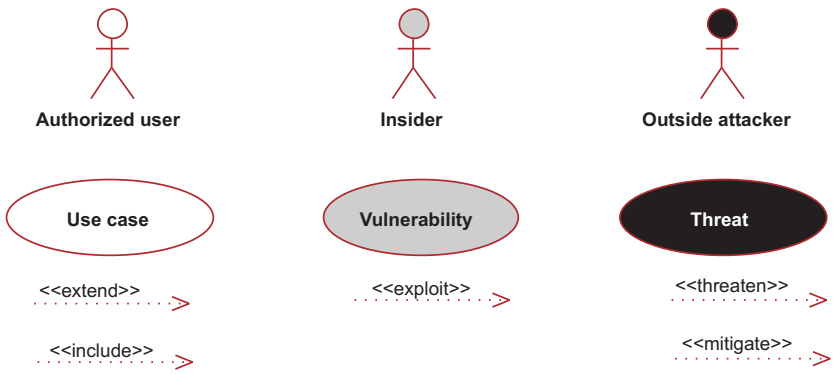

Fig. 2. Extended misuse case legend

3.1 Examples of use of the extended notation

Emergency access control in healthcare systems Figure 3 depicts an example misuse case model using the extended notation proposed in this paper. The model illustrates use and misuse of the access control mechanism in an Electronic Patient Record (EPR) system. As explained in the introduction, such healthcare systems often have emergency access control mechanisms designed to be able to override the standard access control mechanisms in situations where access to information is of vital importance but there is no time to register the patient in the system and link him/her to a specific ward - which is necessary for the standard access control to function properly. In these situations healthcare personnel are authorized, by their organization and the law, to use the emergency access control mechanism to gain access to information that they have a legitimate need and right to view. However, for such an emergency mechanism to be useful, it has to be available at all times. This effectively leads to a backdoor into the system that may be misused by insiders to snoop around when they should not. Most system users will not attempt misusing this mechanism although it is possible. But, it is important to be able to consider the possibility and map out potential consequences and apply proper countermeasures if the consequences are grave. And that is the reason why this addition to the misuse case notation is important. You cannot get a complete overview of potential risks and threats towards a system if you do not consider the complete picture. By identifying emergency access as a vulnerability we are also able to consider proper countermeasuers to apply in order to minimize the risk for misuse - in this case auditing (enables traceability and detection of misuse) and awareness training (e.g. making sure that system users are aware of the consequences of misuse - and what is considered misuse).

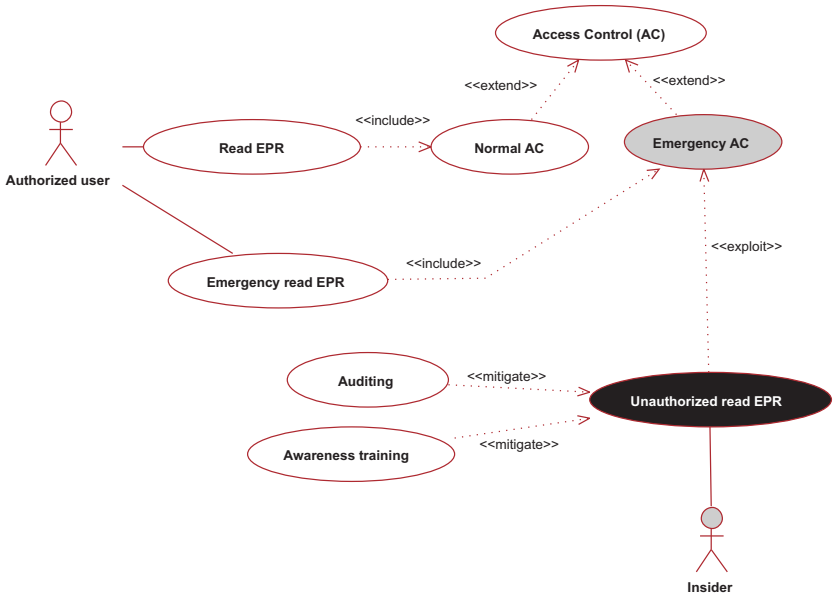

Fig. 3. Extended misuse case example: access control

User input in web-enabled systems In an IT-system all input, from users or other systems, should be handled with caution. Figure 4 illustrates a generic login procedure for a web application - the user has to enter a username and password to log in. Identified attacks include (but are definitely not limited to):

- Injection - for instance sql-injections to tamper with database content or override password check.
- Overflow - entering unexpected or large quantities of data in the input fields to observe system reaction or possibly take control over the system.

Input validation is identified as a countermeasure that helps mitigate these threats. This model illustrates how the extended notation helps highlight vulnerabilites that may be exploited. An insider is not inlcuded because these attacks are typically performed by outside attackers. Highlighting vulnearbilities in this way may be particularly helpful in a risk analysis process, where the customers are involved. By visualizing vulnerabilities, attacks and what may happen it will hopefully be easier to get acceptance and resources to apply security measures.

An insider on the system development team This example illustrates how the extended notation may be used not only on a system level, but also on a business- or organizational level. An insider may exist inside a development

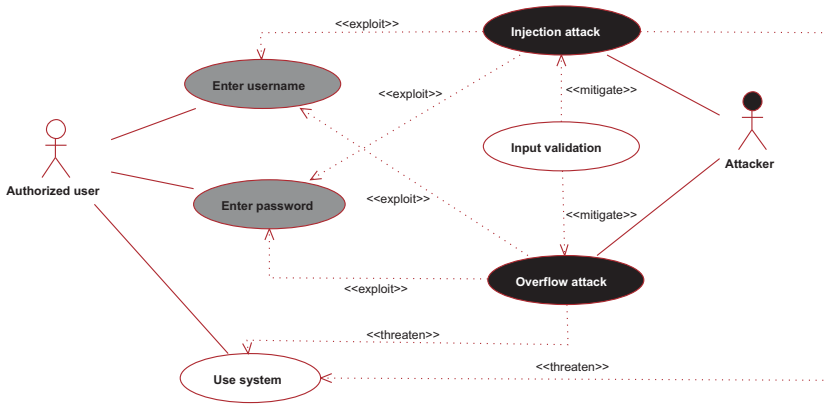

Fig. 4. Extended misuse case example: user input

team or an organization. For example a disgruntled employee working on a development project may inject code into a system that opens up a backdoor that attackers may exploit like Figure 5 illustrates.

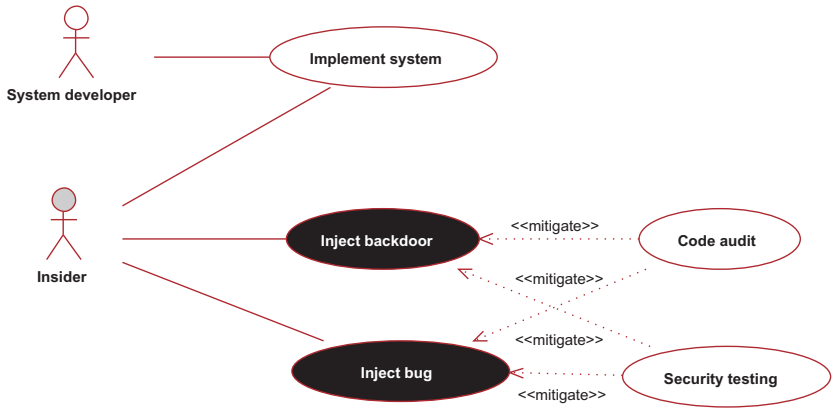

Fig. 5. Extended misuse case example: insider in development team

3.2 A step-by-step approach: how to apply the extended notation

In [15] Sindre and Opdahl propose guidelines, a set of steps, to perform when using misuse cases to elicit threats and countermeasures. The approach described

here for applying extended misuse cases is based on their guidelines, but refined to include the necessary activities to consider the insider threat and uncover vulnerabilities.

1. **Identify actors and use cases for the target system.** Only with an overview of what the system is supposed to do, and who will use it, is it possible to start identifying potential attackers, weak spots and threats. Create UML use case models. These will serve as input to the subsequent steps.
2. **Identify potential outside attackers.** With knowledge about the system discovered in step 1 one should be able to identify who might target the system for attack purposes from the outside. This may include a wide range of persons with a wide variety of motivations - from the unskilled hacker using downloaded tools to jam the system using a DoS[1]-attack, to highly skilled industrial spies. At this point it is important to create as complete a list as possible of potential attackers. Later, in the risk analysis process one can eliminate attackers that are deemed unlikely or that will probably not be able to cause any harm.
3. **Identify potential insiders.** As with outside attackers there may be a variety of insiders. For example there are the people developing a system - who may intentionally inject backdoors - and there are different kinds of users of the operational system that have different rights and thereby differ in the harm they are capable of inflicting on the system. On this point as well, the main concern is to generate as complete a picture as possible.
4. **Identify threats.** Having identified potential misusers who may harm the system, the next step is to identify what types of harm they may want to inflict on the system - i.e. potential threats and attacks. To be able to do this one should consider what might be the goal of the identified misusers - what would they want to achieve?
5. **Identify vulnerabilities.** This step means analysing how threats and attacks may be performed. Given the identified threats - how may an outside attacker, or insider, do this? This means examining the systems functionality identified in step 1 and consider each use case carefully to decide if it may be exploited for malicious purposes. When a potential vulnerability is identified it should be labeled accordingly in conformance with the extended misuse case notation.
6. **Identify security requirements.** Having identified misusers, threats and vulnerabilities - in this step the focus is on countermeasures. This is done by adding security use cases (as earlier mentioned these use the notation of ordinary use cases) to the models and adding the *mitigate* relationship to the threats or vulnerabilities they protect against.
7. **Revise findings so far.** This is of course an iterative process that may be carried out several times before one is satisfied that the result is reasonably sound and complete. Creating a 100% complete overview of all risks is infeasible but applying a structured risk-based approach and using the right

[1] Denial of Service

people with the required knowledge [13] should help ensure the best possible result.

Note that the steps need not necessarily always be carried out in exactly this order. Specifically steps 2 through 5 may be intertwined as it may be hard or possibly not beneficial to completely separate these steps.

4 Relation to risk management in system development

Risk analysis, and risk managed development processes, is a well known technique for making decisions in many engineering fields. Building secure systems is about managing risks. It is not possible to build a system that is absolutely secure against all attacks, known in the present or that may be invented in the future [22]. Risk managed system development is about creating systems that are reasonably protected against known attacks and with a robust build using design principles that will hopefully make the system able to withstand future attacks. What is reasonable protection and what risks should be handled is for the system stakeholders to decide - i.e. the customer. To decide what risks to handle one needs to rank the identified risks and this requires assigning a value. A risk value is calculated as:

$$Risk = Probability \times Consequence \tag{1}$$

Misuse cases provide an overview of information that is very useful in a risk analysis process [10]. However, misuse cases only provides an overview and should be a starting point for creating attack trees [22] and doing threat modelling [20] to get a complete view of the threats and vulnerabilities in a system. Adding notation for expressing vulnerabilities and the insider threat makes the misuse case notation richer and adds more detail which should provide a better starting point for the continuing risk management process.

5 Discussion

Although not all vulnerabilities may be represented in a use case or misuse case model, it is important when considering adding functionality to a system to examine if it represents a vulnerability that can be exploited. Only then is it possible to make a risk-based decision whether to not include that functionality or apply the necessary countermeasures to ensure protection. The possibly greatest power of use and misuse cases is that they are so graphical and easy to understand, and work very well as a basis for discussion with system stakeholders. Typically customers are not eager to spend money on security as it does not directly add to the value of the product. Misuse cases can help convince customers that security is important. Extending the misuse case notation helps this process as it enables:

– Visualisation of effects of adding functionality that might seem desirable, but actually represents vulnerabilities. The extended misuse case notation enables explicitly stating how vulnerable functions may be exploited.
– The insider threat should not be neglected. Insider attackers have very different possibilities from outsider attackers and by using a separate notation for insiders one is able to emphasize this.

The extensions proposed here are simple, in accordance with the original misuse case notation. The idea is to keep close to the UML use case notation and only add what is needed to include security concerns, while keeping the models very easy to understand.

6 Conclusion and further work

This paper has presented and shown examples of an extended misuse case notation including notation for expressing vulnerabilities and insider attackers. This adds to the expressiveness of misuse cases while still keeping the notation very straightforward and easy to understand. The extended notation enables expressing a richer and more complete picture of security threat considerations for a system which is useful when using misuse cases in risk analysis. To further investigate the ideas presented here, it would be useful to create a textual representation of extended misuse cases. Also, security functionality is currently represented as ordinary use cases. It might be useful to create a specific notation for security functionality, or countermeasures that have been added to mitigate vulnerabilities and threats.

7 Acknowledgements

The ideas described in this paper was inspired by the project on case studies of access control in healthcare performed by Julie-Marie Foss and Nina Ingvaldsen at the Norwegian University of Science and Technology (NTNU) in the fall of 2004 which I had the pleasure of supervising. Misuse cases was used in that project to model findings and some alterations, initiated by very useful comments and suggestions from Guttorm Sindre, to the notation had to be made to be able to express all findings. These alterations were the starting point of the extended notation described in this paper.

References

[1] Unified modeling language: Superstructure. Technical report, Object Management Group (OMG), August 2005. http://www.omg.org.
[2] I. Alexander. Initial industrial experience of misuse cases in trade-off analysis. In *IEEE Joint International Conference on Requiremenst Engineering*, Essen, Germany, 2002. IEEE.

[3] I. Alexander. Modelling the interplay of conflicting goals with use and misuse cases. In *Goal-Oriented Business-Process Modeling (GBMP) 2002*, volume 109, London, UK, 2002. CEUR Workshop Proceedings.

[4] I. Alexander. Modelling the interplay of conflicting goals with use and misuse cases. In *International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ) 2002*, Essen, Germany, 2002.

[5] I. Alexander. Misuse cases help to elicit non-functional requirements. *Computing & Control Engineering Journal*, 14(1):40–45, 2003.

[6] I. Alexander. Misuse cases: Use cases with hostile intent. *IEEE Software*, 20(1):58–66, 2003.

[7] I. Alexander and N. Maiden. *Scenarios, Stories, Use Cases: Through the Systems Development Life-Cycle*. John Wiley & Sons, 2004. ISBN: 0470861940.

[8] D. F. Ferraiolo, D. R. Kuhn, and R. Chandramouli. *Role-Based Access Control*. Computer Security Series. Artech House Publishers, Boston, 1 edition, 2003. ISBN: 1580533701.

[9] D. G. Firesmith. Security use cases. *Journal of Object Technology*, 2(3):53–64, 2003.

[10] P. Hope, G. McGraw, and A. I. Anton. Misuse and abuse cases: Getting past the positive. *IEEE Security & Privacy*, 2(3):90–92, May/June 2004.

[11] J. McDermott. Abuse case models for security requirements analysis. In *Symposium on Requirements Engineering for Information Security (SREIS)*, Indianapolis, USA, 2001.

[12] J. McDermott and C. Fox. Using abuse case models for security requirements analysis. In *Annual Computer Security Applications Conference*, Phoenix, Arizona, 1999.

[13] G. McGraw. *Software Security - Building Security In*. Addison-Wesley Software Security Series. Addison-Wesley (Pearson Education), Boston, 1 edition, 2006. ISBN: 0321356705.

[14] G. Sindre, D. G. Firesmith, and A. L. Opdahl. A reuse-based approach to determining security requirements. In *9th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'03)*, Klagenfurt/Velden, Austria, 2003.

[15] G. Sindre and A. L. Opdahl. Eliciting security requirements by misuse cases. In *37th International Conference on Technology of Object-Oriented Languages and Systems (TOOLS-Pacific 2000)*, pages 120–131, Sydney, Australia, 2000.

[16] G. Sindre and A. L. Opdahl. Capturing security requirements through misuse cases. In *Norsk Informatikkonferanse (NIK)*, Tromsø, Norway, 2001.

[17] G. Sindre and A. L. Opdahl. Templates for misuse case description. In *Seventh International Workshop on Requirements Engineering: Foundation of Software Quality (REFSQ'2001)*, Interlaken, Switzerland, 2001.

[18] G. Sindre and A. L. Opdahl. Eliciting security requirments with misuse cases. *Requirements Engineering*, 10(1):34–44, 2005.

[19] G. Sindre, A. L. Opdahl, and G. F. Brevik. Generalization/specialization as a structuring mechanism for misuse cases. In *2nd Symposium on Requirements Engineering for Information Security (SREIS'02)*,, Raleigh, NC, USA, 2002.

[20] F. Swiderski. *Threat Modeling*. Microsoft Press U.S., 2004. ISBN: 0735619913.

[21] D. Verdon and G. McGraw. Risk analysis in software design. *IEEE Security & Privacy*, 2(4):79–84, 2004.

[22] J. Viega and G. McGraw. *Building Secure Software: How to Avoid Security Problems the Right Way*. Addison Wesley, 2001. ISBN: 020172152X.

Paper D

The iAccess Handbook: A Methodology for Access Control Integration

The iAccess Handbook: A Methodology for Access Control Integration

Per Håkon Meland[a], Lillian Røstad[b], Inger Anne Tøndel[a], Øystein Nytrø[b]

[a] *SINTEF, Software Engineering, Safety and Security, Norway*
[b] *Department of Computer and Information Science, NTNU, Norway*

Abstract

Health care information about a patient is usually scattered among several clinical systems - potentially more than a hundred separate systems just within one hospital. System integration and interoperability is difficult to achieve, and various strategies for integration exist. However, one topic that has not received much attention is how to integrate system specific security mechanisms such as access control. This paper presents the iAccess handbook, which is a tool to aid this process. It consists of a repository of reference information and a set of methods for collecting information and presenting results, and concerns the legal, organizational and technological aspects of integrated access control for health information systems. The methods have been applied on two separate integration efforts in Norway, which affect ten hospitals in total.

Keywords:

Access to Information, Information Protection, Computer Security, Medical Records

Introduction

Health care information about a patient is usually scattered among several clinical systems - potentially more than a hundred separate systems just within one hospital. To get a clear understanding and overview of a patient's medical problems, health care personnel need access to all relevant information in a uniform view, but this can be difficult to achieve today. Therefore, system and information integration is one of the current key issues in health care. One topic that has not received much attention yet is how to integrate security mechanisms that are specific to each system, like access control, when systems are integrated. Access control and access decisions are closely linked to knowledge about information in a system, available operations in a system and the users of the system. A sound and sufficient access control scheme is critical in health care systems both to protect the patient's right to privacy, but also to make efficient use of information. When information is integrated - resulting in even larger repositories of information - enforcing access control becomes all the more difficult.

In this paper we present the iAccess (*Integrated Access Control for Health Care Information Systems*) handbook which is a tool to be used during planning, designing and describing access control for integrated health care solutions. The handbook consists of a repository of reference information and a set of methods for collecting information and presenting results.

The methods have been applied on two separate integration efforts in Norway, which affect ten hospitals in total. Due to confidentiality agreements, it is not possible to present direct results from each hospital, but we have created generalized examples of results by combining findings for illustration purposes. We will also discuss and share the general experiences from each of the applied methods. Both the feedback and results have been very positive and useful. With this paper, we wish to encourage similar activities in other countries based on the structure and methods of the handbook.

Related work

As far as we know, little research has been done on the topic of access control in system integration. However, some relevant work has been done on the topic of combining access control policies. Jajodia et al [1] introduced in 1997 the flexible authorization manager (FAM) for enforcing multiple access control policies. In [2] Jajodia et al introduced a language to define decision rules to resolve conflicts among authorizations. Also relevant is the work done by Hu et al and Ferraiolo et al on using what they call a Policy Machine (PM) [3,4]. The PM is a standardized access control mechanism that should require changes only in its configuration to be able to enforce different access control policies. They claim that the PM is also able to support combinations of policy instances e.g. Role-Based Access Control and Multi-Level security. In addition the work of Siewe et al [5] is of interest. They have created a language, Interval Temporal Logic (ITL), which allows for formal specification of access control policies and can handle the enforcement of multiple policies through policy combination. The potential for specification of temporal dependencies in access control rules using ITL is of particular relevance for health care as a collaborative and dynamic environment.

The iAccess handbook

The purpose of the iAccess handbook is to serve as a collection of information and methods that are useful and appropriate when integrating the access control of heterogeneous health care information systems. The handbook itself is web-based, and both readable and editable for registered users, such as people from health care organizations, researchers and students (doctoral fellows primarily). The handbook has been created using the free MediaWiki software package[1], which allows easy publication and development of content in a collaborative setting. The following sections will briefly explain the main contents from each of the three parts of the handbook.

Handbook part 1: Reference information

For the reference part of the iAccess handbook, we have borrowed the concept of viewpoints from the software architecture field - specifically from *IEEE 1471-2000 Recommended Practice for Architectural Description of Software-Intensive Systems* [6]. We have defined three viewpoints; *legal, organizational and technical*. These viewpoints were selected in recognition of the fact that access control is not merely a technical issue. Organizational measures are important in enforcing access control and ensuring patient privacy. The legislation defines *if*, *how* and *when* sharing of sensitive health information can take place.

Legal viewpoint

This viewpoint gives an overview of relevant paragraphs in the Norwegian legislation, a dictionary of legal terms from the selected texts and definitions of legal terms that are used but not formally defined in the legal texts. The purpose of the definitions of terms is to create a common basis and understanding when discussing legal issues and possible interpretations of regulations. It is not so much the legal texts themselves, but rather the current interpretation of them, that limits sharing of health information. The handbook makes this information available to people who need to understand these rules, but who do not necessarily have any formal juridical background. The information has been grouped into six categories for easy lookup and cross-linking:

- Limitations on managing health information.
- Orders, permissions and conditions regarding sending, receiving and exchange health information.
- Information quality.
- Ensuring confidentiality, integrity and availability of health information.
- Internal control.
- Particular technical, physical or organizational requirements for managing health information.

[1] MediaWiki is a free software wiki package originally written for Wikipedia, see http://www.mediawiki.org.

Organizational viewpoint

This viewpoint concentrates on the organizational aspects that influence access control. The goal is to give an overview of how an organization and different work processes can influence access to health care information.

Aspects related to system purpose are organized as follows:

- The users of the system: Some systems are used only by some individuals while others are used by all hospital employees.
- How the system is used: Different systems are used in different ways; e.g. for reading or writing, in emergencies, for a long or short period of time.
- Type of patient treatment in relation to the system: Patients can be treated by one fixed health care professional, by a fixed group, a dynamic group or by everyone on a ward, to name a few alternatives.

Aspects related to the organization are organized in this way:

- Written policies related to access control and information security behavior.
- Informal policies: This is what is considered acceptable behavior among colleagues, and will not always be the same as what is stated in the written policies.
- Acceptable risk: A health care organization will always be subject to risk requirements, both from the authorities and the general public.
- Relevant organizational measures, e.g. routines, awareness-building and training.

Technical viewpoint

This viewpoint contains information that is closely related to the technical concepts of access control. No assumptions are made about the prior technical knowledge of the users of the handbook, and this section has a twofold purpose; providing users that are newcomers to the field with sufficient information to get an overall grasp of access control concepts, models and mechanisms; and equally important, provide a structure of properties of access control models and mechanisms that can be used for classification of systems. The information is structured as follows:

- Reference models: Definition and description of different access control models, such as discretionary or mandatory [7], role-based [8], task-based [9], team-based [10] and domain-based [8].
- Access control regimes: Systems can have no access control at all, it can be an integral part of the system or access control can be enforced by some entity external to the system itself.
- Attributes that can be used for access control decisions: Group [11], role, ward affiliation, physical location, time/shift, relation to information owner (patient), security clearance level.

- Dependency on other systems: The access control mechanism in a system can e.g. depend on information from other systems, like a user database or human-relations system, for making access control decisions.
- Additional measures: These are not a direct part of the access control mechanism, but can provide information that allows detection of misuse. Examples are logging and surveillance.

Handbook part 2: Survey methods

This part of the handbook defines a set of survey methods, and we will here briefly present the main ones and discuss our experiences with them.

Study of documentation

From studying documentation you can learn a lot about an organization and/or a system that will be useful when working with access control integration, and also for planning workshops and interviews (see the following sections). There are certain kinds of documentation with relevance for access control, which should be present in a health care organization, and we recommend focusing on the following:

- Organizational information security policies.
- Organizational structure - including roles and responsibilities for information security.
- Organization-level access control policy.
- Strategies - for instance integration strategy.
- High-level system documentation.
- Description of implementation and use of access control mechanisms.
- Requirements with respect to identification, authentication and authorization.
- Risk analysis based on system security and patient safety.

Different organizations have different types of documentation. The results from a documentation study greatly depend on the organizations studied. It can be just as interesting to see what kind of documentation exists, as to study the actual content. If the anticipated documentation does not exist, asking for it still serves a purpose by making the organization aware of what information could be expected to be present.

Our experiences were that a lot of the information that actually existed was outdated. As systems develop and routines change, documentation is not always updated accordingly. Also, information can be scattered across many documents, with varying level of details and target groups, making it difficult to get a clear understanding and overview.

In many cases, system documentation was considered to be sensitive, and some organizations were reluctant to give that kind of information. Specifically, this was a concern for the systems that the organizations had designed and built themselves to fit their needs.

Process workshops

Our process workshop is based on the methodology defined by Dingsøyr et al in [12]. The purpose of the process workshop is to get input from a heterogeneous group of people that are involved in a given process. In other words: While documentation can provide information about how something is supposed to be done - the process workshop conveys information about how it is actually done.

During a process workshop, the participants are presented a scenario, and then each one writes down keywords related to activities, participating roles, documentation and tools that are used to solve the scenario. This is done in the same way as a traditional brainstorming session. After this, the workshop participants gather around a process map and eliminate/add/reorganize the notes on the map until they have reached a version of the process description that they can all agree upon, as shown in Figure 1.

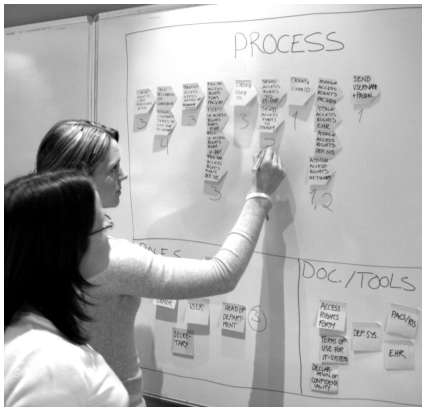

Figure 1 - Creating the process map by using sticky notes and active discussions.

According to the participants, the process maps give a correct and clear view of the real-life situation; that is how things are done, conflicting elements and possible improvements. During the workshops, the participants raise and answer questions such as: What is done, why it is done like this, why not like this, what can be improved and what is most important.

The maps are useful as basis for discussion, and give insight into something that is scattered throughout the organization in documents, intranet pages and people's knowledge. A typical two-hour workshop will usually result in high-level process descriptions, and seldom detailed technical information about the access control mechanisms.

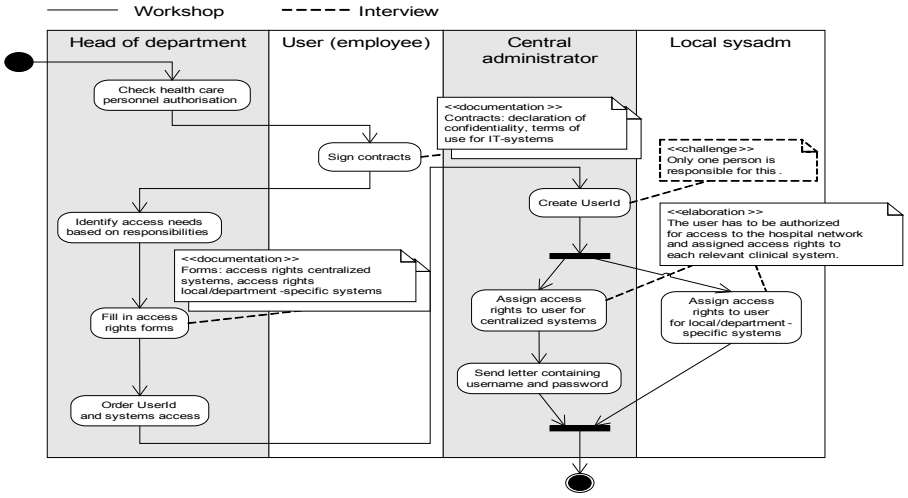

Figure 2 - Sample UML activity diagram including documentation and comments from interviews showing the process of assigning access rights to a new employee.

Semi-structured interviews

Semi-structured interviews [13] are used to gather detailed information to complement and elaborate the process descriptions obtained in the workshops. This is be done by using a set of pre-defined questions (an interview guide), but allowing the interviewees to answer freely - there are no categories to select from and the interviewees are allowed to ask questions. An interview guide should start by explaining the interview motivation, and briefly explain the relevant terms. The interview guide should be based on the process maps.

We found it valuable to be able to interview end-users with different professions and from different wards. The challenges related to physicians are not necessarily the same challenges as those experienced by nurses, who have other experiences than secretaries. The access control solution should consider this.

In most cases, the experiences of the end-users coincided with the opinions of the decision makers and system developers/maintainers, but they also disagreed on several matters, for instance on routine efficiency. The interviewees talked a lot about physical access to computers and login routines, while workshop participants concentrated more on size of logs and details related to the possibilities of better technical solutions. At the end of the interviews we asked the participants how they felt about being interviewed about access control related issues. The response was only positive. Some said that they had never really thought about access to information before, others said that they were glad to be able to tell someone about their experiences with getting access to information, and hoped that things would be better in the future.

Handbook part 3: Combining and presenting results

This part of the handbook describes how to combine and present the results from the survey methods. The most promising technique is the use of an extended version of UML[2] activity diagrams for modeling the process descriptions and information related to processes. An UML activity diagrams is well suited for describing processes and activities, both related to human and system behavior, because of a visual organization that is easily understandable by most people. An example activity diagram can be found in Figure 2, which describes assigning access rights to a newly employed person. In this diagram, the activities are organized in swim lanes, depending on who has the main responsibility for performing the activity. Shadowed boxes represent documentation relevant for the described activities. Attached to the activities are also other types of comments. Four such stereotypes have been defined, two of which are shown in the example diagram. These are: <<challenge>>, <<suggestion for improvement>>, <<user experience>> and <<elaboration>>.

We have found that the extended UML activity diagrams represent the combined findings of the surveys in a very clear way. It is easy to get a grasp of the overall process, at the same time as more detailed information is readily available. However, keep in mind that if too many comments are added, the diagrams may become messy and incomprehensible. Balancing the detail of information and ease of understanding is both a science and an art.

[2] The Unified Modeling Language, see http://www.uml.org

General discussion

The original reason for creating the iAccess handbook was to have an instrument for surveying and documenting real-life access control integration efforts of health care systems in Norway. The results so far have mainly been used by the health care organizations themselves and by doctoral fellows researching how technical, legal and organization challenges should be solved. A survey gives a snapshot of today's situation, what has improved from the past and what is planned in the near and distant future. Just as interesting, some improvements may be negative for the majority of the users, and it is important to share that kind of information with the rest of the community in order to avoid reoccurrence.

There exists a myriad of clinical health care systems in most hospitals today. Systems that individually are not very suitable will probably not be improved by integration, and the way systems are used in real-life must be properly examined. It is our firm belief that research on legal, organizational and technical matters will be important to achieve integrated access control solutions that actually fit the context in which they are used, and in the end – improve health care while protecting the privacy of the patient.

Conclusion

We have presented the iAccess handbook, which consists of three parts relevant for analyzing planned or existing efforts for access control integration for health care systems. Representing multiple views from various stakeholders in unified diagrams eases the understanding on how things are and what should be done. The methods in this paper are first and foremost qualitative, and our future work will add methods that provide more quantitative results.

Acknowledgments

The project Integrated Access Control for Health Care Information Systems (iAccess) is funded by the Norwegian Research Council. The research performed in this project would not be possible without the cooperation and effort of the participating hospitals and organizations. We would also like to thank Professor Dag Wiese Schartum and his associates from the University of Oslo for their valuable contributions.

References

[1] Jajodia S, Samarati P, Subrahmanian VS, Bertino E. A unified framework for enforcing multiple access control policies. In: Proceedings of the 1997 ACM SIGMOD international conference on Management of data; 1997; Tucson, Arizona, United States: ACM Press; 1997. p. 474-485.

[2] Jajodia S, Samarati P, Sapino ML, Subrahmanian VS. Flexible support for multiple access control policies. ACM Trans. Database Syst. 2001;26(2):214-260.

[3] Hu VC, Frincke DA, Ferraiolo DF. The Policy Machine for Security Policy Management. In: Proceedings of the International Conference on Computational Science-Part II; 2001: Springer-Verlag; 2001. p. 494-506.

[4] Ferraiolo DF, Gavrila S, Hu V, Kuhn DR. Composing and combining policies under the policy machine. In: Proceedings of the tenth ACM symposium on Access control models and technologies; 2005; Stockholm, Sweden: ACM Press; 2005. p. 11-20.

[5] Siewe F, Cau A, Zedan H. A compositional framework for access control policies enforcement. In: Proceedings of the 2003 ACM workshop on Formal methods in security engineering; 2003; Washington, D.C.: ACM Press; 2003. p. 32-42.

[6] IEEE Std 1471-2000 IEEE Recommended Practice for Architectural Description of Software-Intensive Systems: IEEE; 2000.

[7] Gollmann D. Computer Security. 1st ed: John Wiley & Sons; 1999.

[8] David F. Ferraiolo DRK, Ramaswamy Chandramouli. Role-Based Access Control: Artech House Publishers; 2003.

[9] Thomas RK, Sandhu RS. Task-based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-oriented Authorization Management. In: IFIP WG11.3 Workshop on Database Security; 1997 1997; Lake Tahoe, California: Chapman & Hall; 1997.

[10] Thomas RK. Team-based access control (TMAC): a primitive for applying role-based access controls in collaborative environments. In: ACM Workshop on Role Based Access Control; 1997; Fairfax, Virginia, United States: ACM Press; 1997. p. 13-19.

[11] Sandhu R. Roles Versus Groups. In: ACM RBAC Workshop; 1996: ACM Press; 1996. p. 25-26.

[12] Dingsøyr T, Moe NB. The Process Workshop: A Tool to Define Electronic Process Guides in Small Software Companies. In: The Australian Software Engineering Conference; 2004 13-16 April; Melbourne, Australia: IEEE Computer Society Press; 2004. p. 350-357.

[13] Fontana A, Frey JH. Interviewing: The Art of Science. In: Norman K. Denzin YSL, editor. Handbook of Qualitative Research. 2nd edition ed: SAGE Publications; 2000. p. 361-376.

Address for correspondence

Per Håkon Meland, SINTEF ICT, NO-7465 Trondheim, Norway.
E-mail: Per.H.Meland@sintef.no, Web: http://www.sintef.com/ict

Paper E

Access Control and Integration of Health Care Systems: An Experience Report and Future Challenges

Access Control and Integration of Health Care Systems: An Experience Report and Future Challenges

Lillian Røstad, Øystein Nytrø
Norwegian University of Science and Technology
Department of Computer and Information Science
N-7491 Trondheim, Norway
lilliaro@idi.ntnu.no, nytroe@idi.ntnu.no

Inger Anne Tøndel, Per Håkon Meland
SINTEF ICT
N-7465 Trondheim, Norway
inger.a.tondel@sintef.no, per.h.meland@sintef.no

Abstract

Health information about a patient is usually scattered among several clinical systems, which limits the availability of the information. Integration of the most central systems is a possible solution to this problem. In this paper we present one such integration effort, with a focus on how access control is handled in the integrated system. Although this effort has not yet solved all the issues of access control integration, it demonstrates a practical approach for creating something that works today and serves as input to the discussion on future challenges for access control when integrating multiple systems.

1 Introduction

In order to ensure patient safety, rapid access to relevant, correct and consistent health information is crucial for healthcare personnel in many situations. Even though electronic patient records (EPR) are becoming more and more prevalent, the patient information is usually scattered over several clinical systems since the clinical information is local or specific to wards. A patient may easily have hundreds of separate, overlapping records in various systems. This limits the availability of the information.

A typical solution to this problem is integration of the most central clinical systems, such as the laboratory, X-ray and EPR systems. However, it is vitally important that the advantages of information integration are not achieved at the sacrifice of patient privacy. Access control is therefore one of the key issues to handle in order to be able to successfully merge and make efficient use of these large quantities of information. Information flow between systems should not compromise the access control rules for the information in any of the systems, and this can be a challenge to achieve if not properly planned and implemented.

In this paper we describe an ongoing integration effort at a Norwegian hospital, with a focus on the implemented access control strategy. This effort serves as background for a discussion of unresolved matters for access control in integrated healthcare systems.

2 Rikshospitalet and the Clinical Portal

Rikshospitalet University Hospital[1], founded in 1826, represents the highest level of specialist care in Norway and is one of the largest Norwegian hospitals. The hospital has about 4000 employees including 500 medical doctors and 1500 nurses. Each year the hospital handels 160 000 outpatient clinic consultations and 50 000 patients are hospitalized for one day or more. A myriad of IT systems of varying age and technological sophistication are used in the everyday treatment of patients. They estimate that a total of approximately 160 clinical systems exists and are in use at the hospital. At Rikshospitalet they recognize the need to integrate these systems to make better use of the clinical information. Access to all relevant information about a patient should aid healthcare personnel in providing the best possible treatment for their patients.

Rikshospitalet has chosen an integration approach based on a web portal solution called the *Clinical Portal*. Through

[1] Rikshospitalet (http://www.rikshospitalet.no)

Second International Conference on Availability, Reliability and Security (ARES'07)
0-7695-2775-2/07 $20.00 © 2007 IEEE

this portal they are currently able to provide integrated access to information from the following systems: PiMS/PAS (patient administrative information), DocuLive (EPR system), RIS-Web (Radiology), Symphathy (Pathology), Miclis (Microbiology) and NetLab (Clinical biochemistry, Immunology and Pharmacology).

The Clinical Portal offers three different desktops to its users:

- **My desktop:** Provides access to e-mail, calendar, contacts, and general news from the hospital.
- **Clinical desktop:** Provides access to information on activities at the user's ward and is the entrance point to all clinical information in the hospital. Through the clinical desktop one can get access to an overview of all patients belonging to one's own ward, and search for patients not currently admitted to this ward.
- **Patient desktop:** Provides access to information on a specific patient, e.g. lab results, orders made (e.g. for new test) and plans. It is also possible to create new orders for this patient.

3 Research Methods

Our goal in the case study was to gather as much information as possible about the clinical portal: technical information about the system itself, about the decisions and choices made when implementing the system, the reasoning behind these decisions, and experiences from use of the system so far. To be able to grasp both technical and administrative information, as well as user experiences, the following methods were chosen for our study:

- **Documentation study:** Two different types of documents were studied: Written policies and routines, and documentation of IT systems that play an important part in enforcing access control.
- **Process workshops:** Two creative workshops were arranged, focusing on different aspects of how access control are handled in the organization and in the IT systems. The workshops were directed towards two different focus groups: Decision makers and system developers/maintainers. The methods used during the workshops were based on [2].
- **Semi-structured interviews:** Interviews were conducted with clinical personnel (e.g. physicians, nurses, nutritionists) and administrative personnel (e.g. secretaries), to get information on how the current access control scheme influences their work day. Interviews were performed in accordance with [1].

The results obtained from using these methods were combined in UML activity diagrams, slightly adapted for this purpose. For more information on the methods used, see [3].

4 Case Study Results: The Clinical Portal and Access Control

In this section the Clinical Portal is described, from a technical, administrative and user perspective. The focus will be on the access control solution. We begin this section by introducing the overall architecture of the clinical portal.

4.1 System Architecture

Figure 1 shows the overall architecture of the Clinical Portal. The architecture is founded on the middleware platform J2EE. We have attempted to keep the layer descriptions on a level of detail sufficient for our discussion of access control in the clinical portal, without getting into too much detail.

- **The Portal Layer:** The portal layer is the interface to the users. *Portlets*[2] handle the users' requests, and interacts with the service layer.
- **The Service Layer:** The service layer provides services for accessing merged information from the source systems either from the Operational Data Store (ODS) which contains merged information from source systems, or by talking to the Hub which connects the source systems.
- **Integration Layer:** The integration layer handles communication between applications and systems. This layer consists of the Hub and a set of adapters that facilitates communication with each source system. To avoid direct changes in the source systems, the individual source systems' data formats are used for fetching and storing data. The data is translated to XML-format by the adapter and transferred from the source systems to the clinical portal. The ODS is also part of the integration layer. The Data Warehouse (DW) contains historical data from the ODS and source systems, and is used to facilitate report generation. Also part of the integration layer is the MetaCatalogue and the OID (Oracle Internet Directory) which is the basis for access control in the clinical portal. We will discuss these in more detail next.
- **The Source System Layer:** This layer consists of the source systems feeding data into the clinical portal.

[2] Java based Web component.

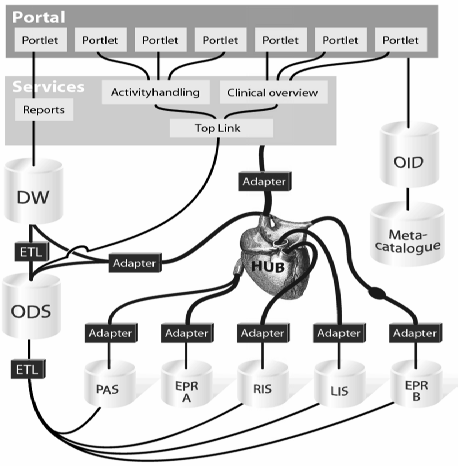

Figure 1. Architecture

4.2 Access Control in the Clinical Portal

4.2.1 Accessing the Main Portal

Figure 2 presents the overall security architecture of the Clinical Portal with a focus on access control. As can bee seen from the figure, access to the portal is role-based. A user gets a different view of the portal depending on his/her role as defined in the MetaCatalogue (an LDAP-server that is part of the Meta directory). A role consists of a user's profession (nurse, doctor etc) and place of work (department, ward). The role also determines information from which source systems are avilable and also which patients (e.g. only those currently admitted to the ward where tue user is working).

In additon to pure roles, Rikshospitalet has adopted the concept of *actualization* in the portal. Actualization is a mechanism defined by Siemens Medical in their DocuLive EPR system. Briefly explained, actualization is an exception mechanism that allows a user to ovverride the role-based access control and gain access to information about a patient. This mechanism is intended for use in situations where there is a user-patient contact that is not known to the MetaCatalogue and therefore access is denied based on roles. Examples of situations where actualization is used include referrals or second opinions, situations when the patient is moved from one ward to another and the MetaCatalogue is not updated when the patient arrives, a patient not currently hospitalized calling in to ask questions about previous treatment and so on. In DocuLive a user has to provide a reason for using actualization, and this triggers extensive logging of the users actions. The Clinical Portal adopts this approach and additionally requires the user to re-enter his/her password when using actualization.

The Clinical Portal offers context-based login, meaning that the users return to the context from which they last logged out. In addition, the portal has a fixed login time, meaning that users are logged out if they have not been active for the last 30 minutes.

Users authenticate themselves to the portal by presenting a username and password. The MetaCatalogue does not contain any information on the passwords of users. The association between usernames and passwords are found in the OID, which is the second LDAP server, and it is only the OID that is involved in authentication. The MetaCatalogue and the OID are continuously synchronized.

4.2.2 Accessing Subsystems

The Clinical Portal pulls information from six different subsystems. Access to the subsystems are handled by the portal, meaning that the portal stores the username and password and forwards these to gain access to information form the differen subsystems. This procedure is enabled by the fact that at Rikshospitalet all users have the same username and password to the Clinical Portal and the six subsystems. This is a first step towards Single Sign-On for all systems at Rikshospitalet. However, for now, this solution requires manual maintenance of identical username/password-pairs in the different systems.

What information is made available is left up to the subsystems to determine. In other words the Clinical Portal simply logs a user onto a subsystems and requests information through an adapter. The subsystem returnes information allowed for this user according to the system's own, internal access rules and returns these to the adapter, who wraps the information in XML and forwards to the Clinical Portal. It processes the information received, from several subsystems, and presents in to the user in a unified fashion.

4.3 Access Control Administration

It is not only the technological solutions that influence how well an access control solution will work in an organization. In the process workshops we therefore also considered the administrative view of the solution, with a special focus on two issues:

- Assigning access rights to a new employee.
- Detection of misuse of the *actualization* access mechanism.

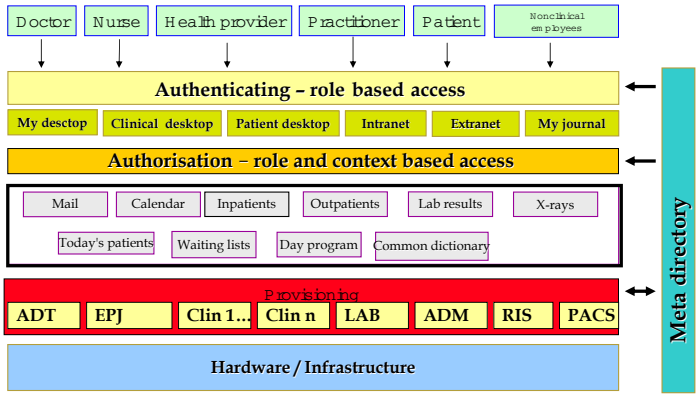

Figure 2. Security architecture

We focused on capturing information about formal and informal procedures, who were involved in the operations performed, and what documentation was used.

4.3.1 Assigning Access Rights to a New Employee

Figure 3 illustrates how a new employee is assigned access rights to all relevant systems. This process involves quite a lot of people from different parts of the hospital. Key components are the paper based access rights assignment forms. There is one form that states which systems a user should have access to - in other words which systems should show up on the desktop when the user logs on the hospital network. Additionally there is a separate form for assigning access to the main electronic record system (DocuLive), which has a rather complex access control solution. The ward leader is responsible for completing these forms and issuing them to the technical staff. Sometimes the user participates in this process, and sometimes office personnel assist the ward leader. The user also has to sign a confidentiality agreement and read and accept the rules for use of the hospital's IT systems. This is a step to ensure security awareness among users.

The access forms are sent to the IT-department which takes care of the practical issues involved in assigning access. Some of the systems may have separate administrators, and in these cases the central IT-department forwards the task of access rights assignment for these systems to them.

4.3.2 Detection of Misuse

Figure 4 depicts the process of discovering and handling misuse. Misuse is not detected automatically, nor are there routines in place for regular auditing of the actualizaton logs from the EPR-system. However, sometimes the logs are checked based on suspicion presented by someone, or possibly motivated by the fact that a highly public figure or celebrity has been hospitalized. If the information found in the logs provide grounds for suspicion of misuse of a patient's record, this is discussed with the patient's primary physician to uncover if it is indeed misuse. The hospital has procedures in place for handling these kinds of incidents. This includes for the ward leader to consider punitive actions in cases of detected misuse.

4.4 Clinical Portal: User Experiences

When introducing a new solution one should always focus attention on user experiences. So far we have only inerviewed a small set of clinical users (only 7), but they did provide some interesting feedback that we have summarized here.

4.4.1 General Experiences with Access Control in the Clinical Portal

Users have experienced that access to information can be cumbersome when systems are not integrated. All users have stories about how they had to log on and off systems several times to get what they want. They are therefore satisfied with the Clinical Portal, where information from several systems is presented together. Most users can only

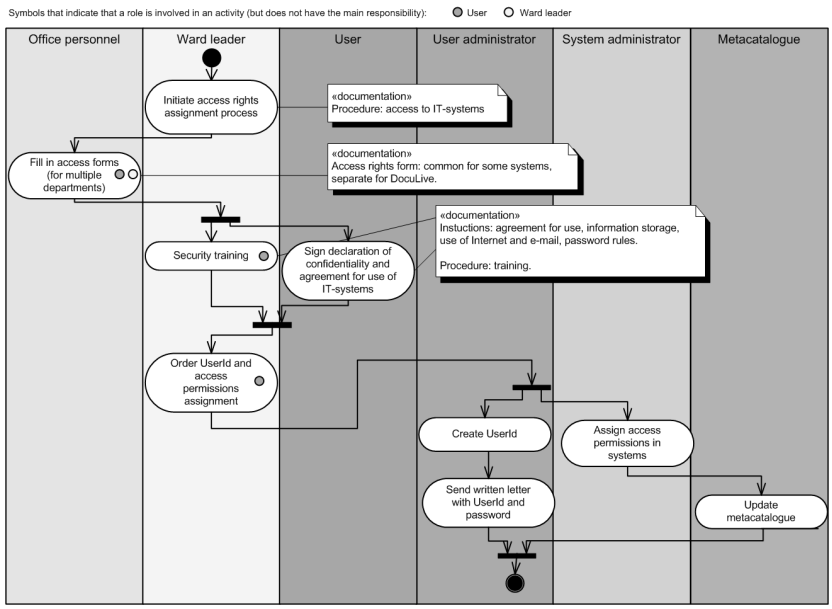

Figure 3. Assigning access rights

think of things that have improved after the introduction of the portal, including access control, the amount of information available electronically and the way the information is presented to the users. When it comes to the access control mechanisms and challenges, the one thing users really react to is that they are, as they put it, "thrown out of the systems" after some limited time.

Integration of systems results in more information being available at the same time, something that may result in information overflow if the information is not presented in a good and flexible way. However only very few users indicate that they sometimes get too much information. Another issue is patient privacy, but no one thinks it is a problem that they get too much information from a patient privacy point of view, though this is an issue they have given little thought before. They feel that this is taken care of by the system, since the most sensitive information can be blocked. They also feel that availability of information is for the patients' own best - it is needed to provide the best care for the patients.

When users are asked which factors should control what information you are allowed to access, many users mention position and place of work, and the patients they are working with. This corresponds well with the factors that are used today - role and ward. Other factors that are mentioned are the needs of the patients and care givers. Strict access control should not reduce service to the patient and the effectiveness and quality the provided care.

The use of actualization is not problematic for users. They are not uncertain of their right to access patient records using this mechanism. Actualization is perceived as necessary and a natural part of their work, though some comment that it could be sensible to have some limitations as to which patients one is allowed to actualize. Finding these limitations is however not easy.

Misuse of access to information, and misuse of actualization in particular, is not a problem, according to the users. Their typical workday is very busy, and there is no time for accessing patient records that are not needed. They are also well aware that access to patient records and the use of actualization is registered in logs. Some of the users have rather high expectations as to what is detected when it comes to misuse. One user even said that if you access a patient record by mistake, you could call the IT-department and say that it was a mistake. Another said that they think the system registers misuse if it recurs.

But though misuse is not experienced as a problem, users are generally acknowledging that checking for and prevent-

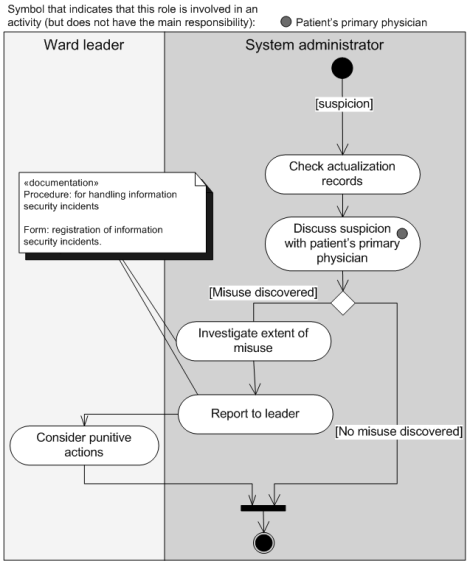

Figure 4. Detection of misuse

ing misuse of information is an important task, and they think it is important that someone in the hospital is working with this.

5 Discussion and Future Challenges

The Clinical Portal represents a step towards integration of clinical systems. However, in the existing portal the access control mechanisms remains unintegrated. A separate access control mechanism has been created for the portal, and the source systems use their own access control to extract information for a given user.

This approach works for now, but it is desireable to be able to develop a more closely integradet solution in the future. However this is certainly not trivial. Access control is closely tied to information. The more fine-grained the access control mechanism is, the more closely it depends on knowledge about information and structure of data. A completely different approach would be to integrate information from all six subsystems, creating an access control scheme tailored to this "new" system. However, this approach requires an enormous intial effort to integrate information from all subsystems and create a new interface to the information. Other disadvantages includes scalability - it would not be straightforward to include information from more systems without considerable effort. Backwards compatibility and historical data is also problematic. It is important to have access to historical clinical information about a patient, so all existing data would have to be incorporated in the new system.

Also, the access control models may differ considerably between the different source systems. Though many systems may use role-based access control, the concept of a role may be very differently defined. Some systems may be using simpler access control models where access is based on e.g. a user's clearance level and/or information category. Some systems may not have any access control at all - if you provide a valide username/password pair you gain access to all information and functions in the system. Some work has been done on the combination of different access control policies, e.g. in the *policy machine* [4][5] but there is still a lot of work that remains to be done on this topic. A key question is if it is even feasible to do or if one should settle for an approach like the one in the Clinical Portal.

Another issue worth discussing is the increased risks related to information exposurer and patient privacy in an integrated system. The more information is made available through one system, the greater is the risk of serious consequences if security is compromised. This concern is taken very seriously by the Norwegia government; the result being that sharing of clinical information between hospitals belonging to different organizations is not allowed in Norway.

This leads us to a general discussion of what type of access control model is suitable and sufficiently secure for healthcare systems. As risks related to information exposure increases, so does the need for an access control mechanism that is sound and precise: which is able to provide healthcare personnell with the required information at the required time - no more and no less. important that is should be no less than required either. The previously mentioned *actiualization* mechanism is a direct result of the inability of the main access control mechanism to fulfill the users information requirements. Actualization is supposed to be an exception. A study of use of this mechanism in the DocuLive EPR system at 8 Norwegian Hospitals showed that 74% of the 16 723 registered DocuLive users were assigned the permission to use actualization. The study also showed that 54% of the patients had had their EPR accessed using actualization. In fact 17% of all accesses to EPRs were performed using actualization. Based on these numbers use of actualization can hardly be consideres an exception, it is in regular use. Allowing use of this mechanism in an integrated solution is probably not a good idea. We should rather strive towards creating an access control model, that is suited for the user's real needs. The more information is included in a healthcare information system, the greater the risk for exposure and need for appropriate protection.

6 Conclusion and Future Work

The shift towards integration and interoperability of clinical systems will continue. In the future inter-hospital integration will also become an issue. Information integration and accessibility offers potentially great benefits for healthcare personnel and patients, but it also greatly increases the risks for patient privacy. As such it is important to focus on sound security mechanisms for authentication, access control and auditing in integrated systems. Even though Rikshospitalet, in their approach so far, has taken some steps towards integration, the issue of access control integration still remains largely unresolved.

References

[1] Norman K. Denzin, Y.S.L.: Handbook of Qualitative Research. 2nd edn., SAGE Publications 2000

[2] T. Dingsøyr, N. Moe: The Process Workshop: A Tool to Define Electronic Process Guides in Small Software Companies, ASWEC2004 2004

[3] P. H. Meland, L. Røstad, I. A. Tøndel: How to mediate between information security and patient safety, Proceedings of the Eight International Conference on Probabilistic Safety Assessment and Management (PSAM8) 2006

[4] Vincent C. Hu, Deborah A. Frincke and David F. Ferraiolo: The Policy Machine for Security Policy Management, In: Proceedings of the International Conference on Computational Science-Part II p.494-506 Springer-Verlag 2001

[5] David F. Ferraiolo, Serban Gavrila, Vincent Hu, D. Richard Kuhn: Composing and combining policies under the policy machine, Proceedings of the tenth ACM symposium on Access control models and technologies, ACM Press, p.11-20 2005

[6] Lillian Røstad, Ole Edsberg: A Study of Access Control Requirements Based on Audit Trails from Access Logs, Proceedings of the 21st Annual Computer Security Applications Conference (ACSAC), IEEE Society Press 2006

Paper F

MG-RBAC: Using Medical Guidelines as a Source of Contextual Information to Activate and Deactivate Roles and Permissions

MG-RBAC: Using Medical Guidelines as a Source of Contextual Information to Activate and Deactivate Roles and Permissions

Lillian Røstad[a]

[a]*Department of Computer and Information Science, Norwegian University of Science and Technology, Norway*

Abstract

Controlling access to information is a key concern in healthcare systems. Some form of Role-Based Access Control (RBAC) is implemented in most healthcare systems. A problem with existing RBAC models used in healthcare is their static nature which doesn't capture the dynamic needs of healthcare providers. In this paper we propose an enhanced access control mode combining RBAC with the use of Medical Guidelines, MG-RBAC. Medical guidelines contain temporal and contextual information that may be used to make more informed, dynamic access control decisions.

Keywords:

Access to Information, Computer Security, Privacy

Introduction

Access control is a key concern in healthcare systems. In order to ensure privacy of patient data, the systems has to provide suitable mechanisms to control access to information. Access control in healthcare has two rather different perspectives:

- at the one hand privacy protection and ensuring that no one gets access to more information than they need, and
- at the other hand patient safety and making sure that healthcare personnel gets access to all information they need to provide the best possible healthcare.

Many existing healthcare systems use some form of Role-Based Access Control (RBAC). Access decisions are typically based on a user's role (e.g. nurse, medical doctor etc) and workplace (department, ward). A user is granted access according to his/her role's permissions for patients that are admitted the ward where he/she is working.

However, these static properties are often incapable of capturing the dynamic needs of healthcare personnel. Patients are transferred and moved more rapid than the systems are updated, resulting in incorrect information forming the basis for incorrect access decisions. To work around these issues, most healthcare systems provide exception mechanisms that allow users to override the access control when they consider their information needs to be legitimate, even if the system thinks otherwise. Using these mechanisms typically requires providing a reason, maybe re-enter your password, and triggers extensive logging of the user's actions. These mechanisms are only supposed to be used in a minority of situations – thereby the name exception mechanisms. However, a study of usage of one such system [1] for one month at eight hospitals in Norway shows that:

- 74% of the users were assigned the permission to use this exception access mechanism
- the exception access mechanism was used on the EPR of 54% of the patients during this period
- in fact 17% of all EPR-accesses were performed using the exception access mechanism.

Looking at these numbers it is clear that usage of this mechanism is in fact not an exception but a common event. This implies that there are situations commonly occurring that should be included in the normal access control mechanism, so the exception mechanism could be left for actual exceptions. The study concludes that there is a need for an access control model for healthcare that is able to handle dynamic events and support workflow and collaborations.

In this paper, present a model for using Medical Guidelines (MG) as a source of information for access control decisions as a way of creating more dynamic access control for healthcare.

MGs (or clinical practice guidelines) are defined by [2] as:

"Practice guidelines are systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific circumstances."

In other words an MG for a given diagnose contain information about best-practice course of treatment developed by experts in the field. Guidelines may exist both as an informal collection of information and in a more formalized, structured manner. There exist several formalized notations for computer-interpretable MGs. MGs may include temporal and event information that implies information needs and therefore may be used in access control.

The next section of this paper provides background information on RBAC and computer-interpretable MGs necessary for the discussion of MG-RBAC. We then move on to some motivating examples explaining how information form MGs may be used for access control. Finally, the general MG-RBAC model is presented before we move on to discussion of potential use, conclusion and our plans for future work on taking MG-RBAC from an idea and a model to testing it out.

Role-Based Access Control (RBAC)

RBAC [3] has become widely popular over the last decade. RBAC is based on the concept of assigning permissions to roles and roles to users. Roles often correspond to positions in an organization. In other words a role represents the permissions needed to perform the responsibilities of a specific position. RBAC has become so popular because of it's:

- Simplicity and ease of administration – there are relatively few roles in an organization compared to the number of users. With RBAC a role has to be defined only one time and can be assigned to many users.
- Flexibility – changing responsibilities for a job position only requires updating permissions for one role, and the update is reflected for all users assigned to that role.
- Scalability – as the organization grows the number of roles may remain unchanged if there are no new positions. New roles may easily be created and assigned to users as needed.

RBAC has been implemented in many commercial systems. Therefore an RBAC-standard [4] has been created to ensure that the main principles remain equal across different implementations. The RBAC standard includes the core RBAC model as shown in Figure 1.

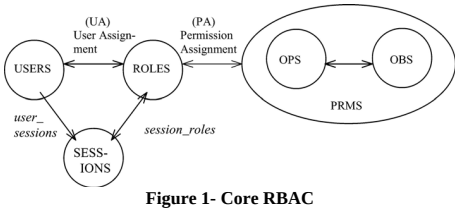

Figure 1- Core RBAC

Figure 1 illustrates how a role is linked to permissions (PRMS). A permission set consists of a set of allowed operations (OPS) on objects (OBS). The Core RBAC model also illustrates that users are assigned to roles either through a static or dynamic (session) link. Through these links the user has a constant set of roles through the static link, and a potential set of roles through the dynamic/session link. Any subset of the roles assigned through the dynamic link may be activated at any given time through a session.
The RBAC standard additionally defines the notion of role hierarchies and permission inheritance, and static (SSD) and dynamic (DSD) separation of duty. Static separation of duty places constraint on the assignment of users to role. Dynamic separation of duty places constraints on the activation of roles in a given session.

Computer-Interpretable Medical Guidelines

Studies [5] have shown that MGs may be effectively used for computer-based decision support – aiding clinicians in making the best decisions. There exist a number of different formats for computer-interpretable MGs and they have several common properties [6] including the organization of treatment plans in decisions and action tasks. A key feature is also the possibility of directly linking the MGs with patient data which enable patient-specific decision support.

Asbru [7] is only one example of such an MG specification language, and many of the others available may be used for informed access control decisions. Asbru has been chosen as the notational example used in this paper because it contains constructs for defining periodic and event-triggered clinical tasks that suits our demonstration needs, and because Asbru MGs are encoded in XML (eXtensible Markup Language[1]) which is a widely used format for exchange of data.

The Asbru Language

Asbru is a time-oriented, intention-based, skeletal plan-specification representation language [8]. A skeletal plan specified in Asbru consists of a name and a plan body and may additionally contain (optional): a set of arguments, a time annotation, preferences, intentions, conditions and effects. The plan body contains a set of plans (child plans) and information about how/in which order these plans should be executed and also conditions on which child plans must be completed in order to complete the parent plan.

An Example MG in Asbru

An example of use of Asbru for encoding a guideline for treatment and observation of Gestational Diabetes Mellitus (GDM – a form of diabetes found in pregnant women) is available at [8]. Use of the guideline is initiated if a glucose tolerance test in the third trimester shows a blood sugar level between 140 and 200 mg/dl. The guideline consists of three main parts:

- Glucose monitoring: measurements performed by the patient herself and/or by the physician. Check to verify that glucose level kept below a limit of 130 mg/dl for 1-hour post meals, < 100 mg/dl fasting and preprandial.
- Nutrition: treatment is based on teaching patient a diet. The goal is to manage GDM with diet and without insulin therapy for as long as possible. Regular follow-ups (every 1-4 weeks) are recommended and should be scheduled individually for each patient.
- Insulin therapy: initiated if blood sugar consistently > 100 mg/dl fasting and/or one hour postprandial con-

[1] http://www.w3.org/XML/

sistently higher that 130 mg/dl and attempts at diet modification has failed.

Note that this is only a short excerpt of the information contained in the guideline.

MG-RBAC

The Asbru guideline for GDM contains both temporal and contextual information that may be used for access control:

- Periodic information needs: visits to physician while under treatment every 1-4 weeks (specific value set for a patient). The EPR does not need to be accessible to the physician in-between visits.
- Events that trigger information needs: when blood sugar readings are too high the patient needs to visit her physician and review treatment. The EPR should be made accessible to the physician when too high readings occur.

Motivation – examples of use

A set of UML use cases have been created to illustrate the envisioned use of medical guidelines in access control for healthcare systems.

First of all a guideline has to be selected for treatment based on a diagnosis as shown in Figure 2.

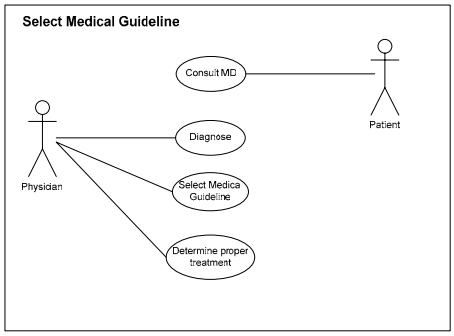

Figure 2 - Guideline selection

In the GDM example the condition for diagnose and guideline selection was a blood sugar measurement of 14-200 mg/dl. A guideline contains generalized treatment advice, and has to be tailored by the physician to treatment of this specific patient.

One example of such tailoring is the periodic consultations that are part of the guideline for GDM. The advice in the guideline only states that there should be regular consultations every 1-4 weeks. A specific time interval has to be selected for a specific patient – e.g. every 4 weeks. From a privacy viewpoint it is desirable to limit accessibility to the patient's EPR as much as possible. Even if the physician is regularly seeing the patient he/she does not need access to the EPR at all times. The physician might need to prepare for an appointment and enter some information after the appointment, but it should be sufficient for the EPR to be accessible to the physician e.g. two days prior to and two days past the next scheduled visit for a patient. Figure 3 illustrates how this may be done. The physician will have an assigned role that includes permissions to this patient's EPR as he has a responsibility for this patient. But the role providing access rights do not need to be activated at all times as explained in the section on the Core RBAC model. The doctor should receive an alert, through role membership, that the EPR has been made available.

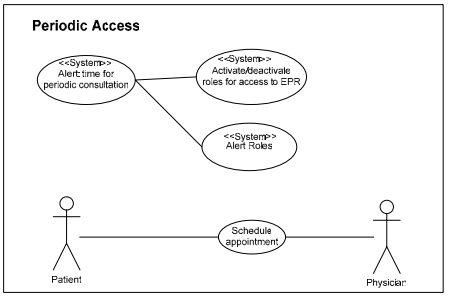

Figure 3 - Guideline: periodic access

The third example of use of guideline information for access control decisions is the occurrence of events that trigger information needs. A typical example of such an event is a measurement of some sorts, made manually or by a sensor, which triggers further actions. For the GDM example the glucose monitoring illustrates such an event. The patient is to measure her own blood sugar level 4 times a day. If the measured level is above some specified limit further action needs to be taken. To determine further actions the physician needs access to the patient's EPR. Figure 4 illustrates how roles are activated if the guideline specifies that a measure results in an action that requires access to the EPR and the relevant role (or roles) is activated.

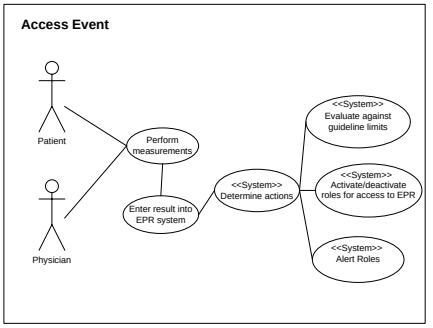

Figure 4 - Guideline: event trigger

The MG-RBAC Model

Using these examples of use as input a preliminary MG-RBAC model, shown in Figure 5, has been created, showing in a bit more detail how this would work in a system.

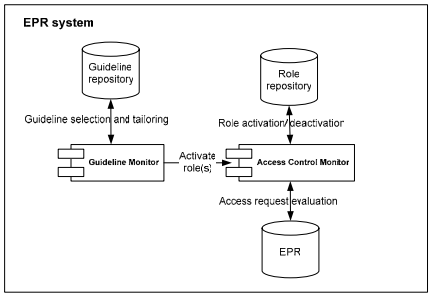

Figure 5 - The MG-RBAC model

A Guideline Monitor would be responsible for receiving events like sensor data or manual measurements and track time for the next periodic event. When a triggered or periodic event occurs the Guideline Monitor would request that the Access Control Monitor activates the appropriate roles. The Access Control Monitor would then be responsible for alerting the users assigned these roles and for evaluation of subsequent access request.

Discussion

The model presented here for MG-RBAC is only very preliminary and serves to inform about a promising idea that requires further work.
The examples presented are based on a guideline representation in the Asbru language. Certainly for such a model to be useful it should be able to use guidelines in many different notations. One possible solution would be to integrate a guideline translation engine in the Guideline Monitor module.
Work remains as to examine in details information contained in other guideline specification languages and how they may be translated.
The examples presented here only illustrate triggered and periodic events. There may be additional information contained in guidelines that could be utilized in access control, but this has not been fully explored yet.

Conclusion and Future Work

In this paper we have presented an idea and a preliminary model for using medical guidelines as input to access control. The idea is that guidelines contain information that can assist in creating a dynamic and context aware access control model for healthcare.
We intend to continue to explore this idea further by creating a more detailed model and developing a proof-of-concept implementation.

Acknowledgments

I would like to thank my supervisors Øystein Nytrø and Svein J. Knapskog for always being ready to discuss my ideas and for challenging me.

References

[1] Røstad L., Edsberg O., A Study of Access Control Requirements for Healthcare Systems Based on Audit Trails from Access Logs, In Proceedings of the Annual Computer Security Applications Conference (ACSAC), Miami, 2006.

[2] Field M.J, Lohr K.N., Clinical Practice Guidelines: Directions for a New Program, The National Academy of Sciences, 1990.

[3] Ferraiolo D.F., Kuhn D.R., Chandramouli R., Role-Based Access Control, Artech House Publishers, 2003, ISBN 1-58053-370-1.

[4] American National Standard for Information Technology: Role Based Access Control, ANSI INCITS 359-2004, American National Standards Institute, 2004.

[5] Shiffman R.N., Liaw Y., Brandt C.A., Corb G.J., Computer-based Guideline Implementation Systems: A Systematic Review of Functionality and Effectiveness, JAMIA, 1999, v. 6, p. 104-114.

[6] Peleg M, Tu S., Bury J., Ciccarese P., Fox J., Greenes R.A., Hall R., Johnson P.D., Jones N., Kumar A., Miksch S., Quaglin S., Seyfang A., Shortliffe E.H., Stefanelli M., Comparing Computer-Interpretable Guideline Models: A Case-Study Approach, JAMIA, 2003, v. 10, p.

[7] Miksch S., Shahar Y., Johnson P., Asbru: A Task-Specific Intention-Based, and Time-Oriented Language for Representing Skeletal Plans, 7th Workshop on Knowledge Engineering: Methods and Languages (KEML-97), 1997.

[8] The Asgaard Project, http://www.asgaard.tuwien.ac.at (last accessed: December 2006).

Address for correspondence

Lillian Røstad

Department of Computer and Information Science

Norwegian University of Science and Technology

N-7491 Trondheim

Norway

Paper G

Towards Dynamic Access Control for Healthcare Information Systems

Towards Dynamic Access Control for Healthcare Information Systems

Lillian Røstad[a], Øystein Nytrø[a]
[a]*Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway*

Abstract. Access control is a key feature of healthcare information systems to protect the privacy of patients and to ensure access to information as required by healthcare professionals. A problem with many existing access control mechanisms is their static nature. In this paper we propose combining workflow information from medical guidelines, observations and audit logs to create dynamic access rules that are adapted to the actual workings of a hospital. Our aim is to help minimize the use of "break the glass" access.

Keywords. Security, Data protection, Evidence based guidelines

Introduction

Access control is one of the key features of health care systems. Access control is about restricting as well as ensuring access to information. These are two inherently different viewpoints. For privacy it is important that access is only granted when there is a legitimate need. For availability it is equally (some would argue more) important that access is granted to all information required to provide the best possible care. The goal of the work presented here is to narrow the gap between these viewpoints, by proposing a method for dynamic access control rules that adheres to the actual flow of work and responsibilities in a hospital setting.

1. Access Control Concepts

Access control is about enforcing rules on which operations a user is allowed to perform on a resource (eg. information) in a system. There are several different access control models. The most common ones are mandatory access control (MAC), discretionary access control (DAC), and role-based access control (RBAC) [1]. RBAC is the preferred model used in many implemented access control mechanisms in health care systems and serves as the foundation for the ideas presented here.

1.1.Role Based Access Control

The concept of Role Based Access Control [1] has gained increasing popularity over the last decade. In RBAC a set of roles is created that corresponds to job functions in

an organisation. Each role consists of a set of access rules. Rather than assigning access rules directly to a user, a user is linked to the role and thus has all the access rules associated with that role. Several users may be assigned to the same role, and one user may take more than one role. A typical RBAC role in a health care system would be that of a nurse. The nurse role consists of access rules that correspond to the information access a nurse needs to perform his job.

The main advantages of RBAC are ease of administration, flexibility, and scalability. RBAC is considered a good fit when there are considerably fewer roles than employees in an organisation. When a new nurse is hired there is no need to create a specific access profile for him – he can simply be assigned to the existing nurse role. This scales well as it is easy to add more nurses as an organisation grows, and it is flexible because changing the access rules for all nurses only requires changing one role.

Some health care systems [2] combine a role with the user's place of work to make access decisions. In short, this means that a nurse only has access to users that are currently admitted to the ward where he works. Dynamic RBAC is extended to assign roles temporarily, according to work shifts or work processes.

1.2.Optimistic security

In [3] Povey proposes the concept of *optimistic security*. The key feature in an optimistic security mechanism is the use of *retrospective control*. There are no access rules that are enforced when a request is made. The concept relies on the ability of someone to examine the logs later and determine if the access was legitimate. Auditing and traceability therefore are keys to enforcing optimistic security. Povey argues that optimistic security is well suited for systems such as healthcare where there may be situations when a user needs to exceed his normal privileges.

Optimistic security exists in many healthcare systems as a "break the glass" mechanism intended to be used in emergency situations. A study [2] has looked into use of the "break the glass" mechanism in a system where normal access control is enforced as a combination of role and workplace as explained earlier. In the study audit data was collected for one month's use of the system at eight hospitals. The study found that 54% of the patients admitted in this time period had their record accessed using the "break the glass" mechanism. Out of all accesses made in this period, 17% were performed using the "break the glass" mechanism. These findings strongly suggest that the rather static approach to access rules (role and ward) does not perform very well in a dynamic hospital setting.

The 17% accesses resulted in almost 300 000 entries in the audit logs. The study also found that there were no automatic audit analysis tools in place. The amount of audit trails and the absence of tools make the task of analyzing audit trails for retrospective control impossible. A condition for optimistic security to work, is that the amount of use is minimal so manual review is realistic.

In health care there will always be situations where availability of information is crucial and "break the glass" mechanisms are needed. One example is emergency situations when there is no time to properly register the patient in the administrative systems, which often is a requirement for normal access rules to apply. The goal is therefore not to completely eliminate the use of "break the glass", but to reduce the use to an amount where it is feasible to perform retrospective control. One approach towards this goal is developing access control mechanisms that are better adapted to the

actual workings of a hospital and are dynamic in the sense that they are able to change and adapt as situation and context change. We will explore this idea further in this paper.

1.3.Dynamic access control – related work

As stated earlier, a problem with many access control rules in health care is the "define once – use always" approach and the lack of dynamic properties and adaptability. Several extensions to RBAC have been proposed to include dynamic properties. Examples include *role delegation [4]* and *context-sensitivity [5]*. Role delegation allows a user to delegate her role to another user to transfer responsibilities either permanently or time-limited. In the proposed context-sensitive RBAC models, context is used to activate and deactivate roles. A user may have a large pool of assigned roles and only a subset of these may be activated at any given time. Context properties may be used to regulate the activation of a role. E.g info about work schedule may be used to activate roles depending on time and place of work.

Though some propositions have been made on how to make RBAC more dynamic, a discussion of exactly what properties or values may be used remains. In the remainder of this paper we propose combining established best practices (medical guidelines), collected observational data, and audit data to learn patterns of information used in healthcare and apply these patterns to create access control rules that will help minimise use of «break the glass» access.

2. Workflow knowledge

Medical guidelines, work plans, observed behaviour, and audit data all contain information about workflow in healthcare. While medical guidelines are the idealised version of the medical activities related to a problem, observational and audit data reflects what actually happens [6]. Moreover, guidelines do seldom assign roles or resources. However, by combining theses sources of knowledge we can create a coherent view of enacted workflows in healthcare, with an emphasis on information access requirements that may be utilized for access control.
In this section we discuss medical guidelines, observation data and audit logs separately and provide motivational examples of how this information may be used for access control purposes.

2.1.Medical Guidelines

A medical guideline (MG) for a given diagnosis contains information about best practice course of treatment developed by experts in the field. Guidelines may exist both as an informal collection of information and in a more formalised, structured manner. MGs often include temporal and event information that implies information needs that may be utilized for access control purposes.

An example of a guideline for treatment and observation of Gestational Diabetes Mellitus (GDM – a form of diabetes found in pregnant women), encoded in the Asbru language for computer-interpretable medical guidelines, is available at [7]. Use of the guideline is initiated if a glucose tolerance test in the third trimester shows a blood

sugar level between 140 and 200 mg/dl. The guideline consists of three main parts: glucose monitoring, nutrition, and insulin therapy.

The Asbru guideline for GDM contains both temporal and contextual information that may be used for access control:

- Periodic information needs: visits to physician while under treatment every 1-4 weeks (specific value set for a patient). The EPR does not need to be accessible to the physician in-between visits.
- Events that trigger information needs: when blood sugar readings are too high the patient needs to visit her physician and review treatment. The patient record should be made accessible to the physician when too high readings occur.

2.2.Observational data – empirical grounding of guidelines

Guidelines are constructed by experts and represent idealized treatment processes – what is expected to happen given a diagnosis. In reality, each patient and care process is unique; furthermore, a complex problem will require that different guidelines are combined. A guideline may serve as a starting point, but will often need to be adapted to the specific situation at hand. In [6] the authors discussed how to use methodical observations of clinical care situations to improve guideline implementation.

An observational study was carried out in the summer of 2005. Two medical students observed clinicians at work in the pre-rounds meeting and ward rounds. They took detailed notes of who were present, the subject of discussion (patient), information sources (written/electronic and oral), and specifics about what type of information was used. In each observation session they followed one clinician and from her viewpoint they noted who else were present and what role they had in the situation. We have reviewed these data to construct an example of how observational data may be used to create patterns of information\needs, shown in Figure 1.

Due to space limitations, Figure 1 shows only the first few interactions in the pre-visit meeting, but it is sufficient to serve our purpose as an illustrative example. In this case they are discussing patient NN. The patient is new to the doctor so the nurse fills him in on some background info. Several information sources are used – some are paper-based (the patient list and the patient chart) and some are computer-based information systems (the electronic patient record (EPR) and the radiology imaging system (IDS)). The figure illustrates communicative acts between the actors present and the actors and the information sources they use. Roles are used to label the actors. This figure illustrates how observation may be used to uncover information needs in specific situations with a specific diagnosis (in this case heart failure), and link these to roles. Though not shown in Figure 1, the observational data shows that the diagnosis changes as test are being done and test results received and reviewed, as is very common. Through observational studies we can examine these transitions and study transfer of responsibilities and access requirements related to this.

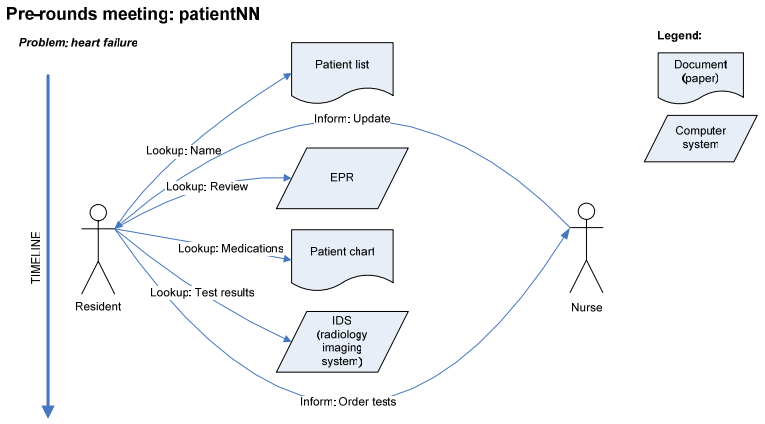

Figure 1 – Information needs in pre-rounds meeting

Even if observations provide real-world examples that may be collected over time, generalized, and used to improve guidelines, they still only give us a relatively high-level view. To complete this picture and get detailed and accurate information about information accessed and actions performed, we turn to the audit logs.

2.3. Usage patterns from audit logs

Most health care systems keep complete history; of changes in information and of user actions. The purpose is to always be able to roll back to a previous state, and to have complete traceability. This means that there exist audit logs with very detailed traces of user actions: the user's role at the time, what information was accessed, for which patient and what actions were performed [2]. From these audit logs it is possible to create generalized usage patterns per role. If a system allows "break the glass" access, it is also common to require the user to provide a reason for doing so and keep a log of these reasons as well [2]. We suggest utilizing this information for access control by:

1. Examine the reasons for using "break the glass" – any reasons that occur often should be considered as candidates for inclusion in the access control rule set.
2. Look for common usage patterns that describe workflows inwards. Examples include:

   Temporal patterns

   If action X occurs – then action Y occurs within Z time.

   Responsibility patterns

   If action X is performed by Role A – then action Y is performed by role B.

   Location patterns

   If action X is performed at ward 1 – then action Y is performed at ward 2.

   Situation patterns

   Role X is in situation S in a guideline, and requires specific information.

3. Discussion

"Break the glass" access is necessary to handle unexpected situations, but it constitutes a security risk and may be misused. The ideas presented here aim at minimizing the need for glass-breaking and making retrospective control feasible.

In access control, the main concern is privacy, where access should only be granted to the information required by an actor in any situation. Clinicians may well disagree with this from the viewpoint that it is better to have broad access. In this paper we therefore suggest an approach to access control that combines guidelines and learning from observations and logs. The goal is to take another step towards the goal of having access mechanisms that support the work of care providers, while protecting the privacy of patients.

The approach presented here is not another "do once – use forever approach". It is fundamental to this idea that observing, learning, and improving should be a continuous process, allowing access rules to adapt to a dynamic, ever-changing environment.

4. Conclusion and future work

In any clinical situation, the information about a patient can be ordered along a continuum from highly relevant, via interesting, to irrelevant, and at the other extreme; illegal according to laws of privacy. Being able to sort correctly may mean life and death. The main problem facing today's busy clinician is avoiding irrelevant information and at the same time getting access to relevant information. In this perspective, relevance ranking and access control depend on the same knowledge about situation, role, guideline, and care process. We believe that optimistic access control, based on analysis and learning from practice as intended and as enacted, is a first step towards both effective relevance ranking and optimal access control.

References

[1]. Ferraiolo, D.F., D.R. Kuhn, and R. Chandramouli, Role-Based Access Control, 1 ed. Computer Security Series. 2003, Boston: Artech House Publishers, ISBN: 1580533701

[2]. L. Røstad, O. Edsberg, A Study of Access Control Requirements for Healthcare Systems Based on Audit Trails from Access Logs, Proceedings of the 22nd Annual Computer Security Applications Conference, Miami, USA, 2006.

[3]. D. Povey, Optimistic security: a new access control paradigm, Proceedings of the 1999 workshop on New Security Paradigms, Ontario, Canada, 1999.

[4]. Zhang, L., G.-J. Ahn, and B.-T. Chu, A role-based delegation framework for healthcare information systems. Proceedings of the seventh ACM symposium on Access control models and technologies. 2002, Monterey, California, USA: ACM Press. 125-134.

[5]. Wilikens, M., et al. A context-related authorization and access control method based on RBAC, in Symposium on Access Control Models and Technologies. 2002. Monterey, California, USA: ACM.

[6]. D. Sørby, T. B. Røst, Ø. Nytrø, Empirical Grounding of Guideline Implementation in Cooperative Clinical Care Situations, AI Techniques in Healthcare: Evidence-based Guidelines and Protocols (Workshop proceedings), Riva del Garda, Italy, 2006.

[7]. The Asgaard Project, http://www.asgaard.tuwien.ac.at (last accessed: November 2007).

Paper H

An Initial Model and a Discussion of Access Control in Patient Controlled Health Records

An Initial Model and a Discussion of Access Control in Patient Controlled Health Records

Lillian Røstad
Norwegian University of Science and Technology
Department of Computer and Information Science
Sem Sælands vei 7-9, 7491 Trondheim, Norway
lilliaro@idi.ntnu.no

Abstract

Health information about a patient is usually kept local to the hospital or clinic where the patient was treated. Patient Controlled Health Records (PCHR) has been proposed as a means to collect all this information and make it available to the patient. In a PCHR the patient is in control and determines who gets access to his health information. In this paper we present a set of usage scenarios to explore the concept of a PCHR. From the scenarios we deduce a set of concerns of relevance when designing an access control model for a PCHR. Finally we outline an initial access control model for a PCHR.

1 Introduction

Improved information technology is seen by many [1] [2] as the best means of making health care delivery more consistent, comprehensive, safe and timely. Accurate and complete medical records are a prerequisite to this vision.

Health information about a patient is usually kept in local systems, specific to a ward or clinic, and accessible only to health care personnel. For every point of care there are separate systems to record information, and information flow between systems is very limited. Even if the information is immobile, the patient is not. As a consequence, patients often find themselves having to retell their medical history and redo tests whenever they encounter a new health care provider.

Personal Health Records (PHR) have been proposed as a potential solution to this problem. The Markle Foundation, a public-private organization, in a report from their Connecting Health care in the Information Age Project defines PHR as follows:

> "An electronic application through which individuals can access, manage and share their health information, and that of others for whom they are authorized, in a private, secure, and confidential environment." [3]

As pointed out by [4] this definition is a good starting point, but more information is needed on the context and use of PHRs. A PHR, in its most common current form, is a web-based system where a patient can enter notes and information about his health condition and share this information with his health care providers. Some PHRs also import clinical information and make this accessible to the patient through the system. However, most PHRs are local and specific to one point of care [5], or to a set of care sites that subscribe to the same PHR from one software vendor. As such, most existing PHRs only provide the patient with limited insight into parts of his health care information.

The goal of a *Patient Controlled* Health Record (PCHR) [6] is to assemble the patient's complete health history and grant the patient control over who gets access to this information. A PCHR differs from the usual PHR in that it exists outside of organizational boundaries. A PCHR contains data from multiple care sites, and the patient is in complete control of the information. This means that it is the patient himself who is administrator and assigns access rights to grant other users access to his information. Through a PCHR the patient may choose to share his data with health care providers and family members. He may also use the PCHR to release part of his health information for research studies or public health purposes.

PCHRs provide a technology that may address several common problems in health care and health information exchange but, as a technology, they present some new challenges. Developing an approach to supporting access controls that corresponds to the dual needs for protecting patient privacy and autonomy (on the one hand) while preserving a high degree of flexibility so that the real variation in the conditions and circumstances of patients is served is the main challenge.

The remainder of this paper explores in detail the concept of a PCHR focusing on how it may be used for sharing and what this means for access control. From this discussion we deduce a set of concerns that need to be included in an access control model. Based on these concerns we outline an initial access control model for a PCHR. We conclude by summarizing what future work is needed on the topic.

2 Related work

The main concepts of Role-Based Access Control (RBAC) is introduces in this section as it is fundamental the initial model. Much of the background for he discussion in this paper comes from the Indivo[1] PCHR system. Work on this system has been ongoing for years, and the author of this paper was fortunate enough to get to work with the Indivo team to study PCHRs and the issues related to this system. In this section we briefly describe the Indivo system with a focus on the current access control model. The information presented in this section serves as a basis for further discussion in the remainder of this paper.

2.1 Role-Based Access Control

Role-Based Access Control(RBAC) [7] has become one of the most common access control models, and is by many [8] [9] [10] [11] considered particularly well-suited for health care systems. RBAC is based on the concept of assigning permissions to roles and roles to users. Roles often correspond to positions in an organization. In other words a role represents the permissions needed to perform the responsibilities of a specific position. The access profiles mentioned in the previous section on Indivo may be considered to represent roles in the system. RBAC has become so popular because of its:

- Simplicity and ease of administration. There are relatively few roles in an organization compared to the number of users. With RBAC a role has to be defined only one time and can be assigned to many users.
- Flexibility. Changing responsibilities for a job position only requires updating permissions for one role, and the update is reflected for all users assigned to that role.
- Scalability. As the organization grows the number of roles may remain unchanged if there are no new positions. New roles may easily be created and assigned to users as needed.

[1] http://indivohealth.org

RBAC has been implemented in many commercial systems. Therefore an RBAC-standard [12] has been created to ensure that the main principles remain equal across different implementations.

2.2 The Indivo PCHR

The architecture of the Indivo (formerly PING) PCHR system was outlined in [13]. Key features of Indivo are:

- The patient is in complete control and in charge of determining who gets access to his information.
- Information is imported into Indivo from clinical systems and made available to the patient.
- The Indivo code is open source and available for anyone to download and customize or adapt to their needs.
- Public health surveys may be deployed through Indivo and the patient given the option to participate.

2.2.1 Access Control in Indivo

When a user is registered in Indivo he is given a role (researcher, patient or provider) that is used to restrict functionality that is available to the user in the system. For example only an administrator may create new users. In addition to this, access profiles are used to set permissions when a user is sharing his record with another user. The current implementation presents the patient with a set of predefined access-profiles to choose from when sharing. An access profile is a set of permissions. Assigning an access profile to a user means granting this user all the permissions included in the profile. There are currently five access profiles that are available to a patient when sharing his record in Indivo: primary care provider, family member, friend, school and research administrator. The users that have been assigned the role of a provider has an additional access profile to choose from - patient - that allows a provider to connect with his patients in the system.

The Indivo access control model provides the patient with the opportunity to share his record and to some extent also to determine what access rights are granted. However this control is limited by the predefined access profiles, and the patient has no way of knowing the exact permissions included in each profile. Through trials, system tests and focus groups the Indivo team has collected knowledge about expected use of the system, as well as concerns and wishes from the system users. This knowledge has been used to formulate a set of usage scenarios and concerns that capture requirements that needs to be fulfilled by a more comprehensive access control model for a PCHR. These usage scenarios and concerns are presented in detail in the next two sections of the paper, and serves as a basis for our proposed access control model for a PCHR.

3 Usage Scenarios

The expected use of the system is best presented by a set of usage scenarios. These scenarios have been selected because they illustrate the most common expected uses of the system, as well as a set of identified likely, but uncommon, use of a PCHR. All scenarios focus an sharing. The scenarios illustrate usage around which design decisions have to be made.

3.1 Patient moving or changing care providers

Very few people live in the same place for their entire life. One of the main purposes of a PCHR is to enable patients to manage access to their own health information. When moving or switching to a new care provider for other reasons, the patient may use the PCHR to grant access to his health information to his new health care provider.

3.2 Patient in need of medical care while traveling

A patient may become ill or injured while traveling or away from home. Through a PCHR the patient can give health care providers access to the information they need to provide proper care.

3.3 Sharing with family or friends

Sometimes a patient may want to share his health information with family or friends. Common cases may include young adults seeking advice from their parents, and persons with chronic or long-term diseases who rely on help from others. Yet another case that is expected to be common, is elderly parents seeking advice and help from their younger, and more computer-literate, children.

3.4 One-time sharing

There are several situations when a patient may want to share selected health information for a very limited time. Examples include colleges requesting access to the latest physical examination, insurance companies requesting a bill of health. For these situations the patient should be able to limit the time sharing is valid for. However, if a patient has chosen to share parts of his medical data with, say, an insurance company for a limited time - there is no sure technological way to keep them from making a copy of the information and keep that in their own record. The only way to force someone to ”forget” may be by law. But, through time-limited sharing one can at least deny sharing of the evolving data.

3.5 Sharing in emergency situations

In an emergency situation a PCHR can serve as a valuable source of information for the emergency care team. Depending on the severity of the patients injuries he may grant access or an access mechanism for emergency situations may be used.

3.6 Patient is an adolescent

When a patient is an infant or a child, the parent (or legal guardians) typically have access to the medical records and act as administrators on the patients behalf. They make decisions about the circumstances and conditions for sharing information. As the patient becomes older there may be specific information he wants to keep from his parents. When the patient becomes an adult he takes over as administrator. The parents does not have access anymore unless the patient chooses to share with the parents. Complicating the matter here is the fact that the rights and obligations of an adolescent and his parents is regulated by the law. These laws are specific to a country, or even state within a country, and as such, even the definition of the age-range for an adolescent may vary from one state to another. The question is which rules prevail.

3.7 Patient has a legal guardian

If a patient, at any time in life, becomes incapable of caring for his own interests due to incapacity or disability, a legal guardian is appointed. The legal guardian is then acting as administrator of the record on the patient's behalf.

3.8 Sharing for research

Through a PCHR the patient can contribute his data to research. What data is shared depends on the research project. Sometimes, research studies reveal medical knowledge about individuals that they should be made aware of. However, as research data are usually de-identified, there is no other means for researchers to get in touch with these patients other than issuing a public notice and hoping that they read it. In [14] it was proposed how the PCHR can be utilized for this purpose. The authors suggested a model where the researches could broadcast an electronic notice. An *agent* within each patient's PCHR could then examine the notice and determine if the patient should be notified. This way, the research subjects stay anonymous, but the researchers has a means of reaching specific subjects with important, targeted information.

4 Concerns

This section presents concerns that are common to all usage scenarios. These concerns represent key points and decisions that has to be made when designing the access control model. We have identified three main groups of concerns:

- Simplicity
- Time
- Transparency

Related concerns have been grouped under these headings. We will continue to use these three main groups in our discussion, to justify and explain our choices.

4.1 Simplicity

This is a matter of what it means to grant the patient control over his own information. What is detailed enough to facilitate all usages scenarios - yet simple enough that the patient is capable of doing it?

Patient control
In the simplest case; the patient is given a set of roles to select from when sharing and the roles determine the permissions granted. The roles are predefined in the system and the patient is not allowed to change them. The only options are sharing using one of those roles - or not sharing at all.

On the other extreme, we have the possibility of giving the patient complete control. When sharing the patient could be allowed to specify exactly what information to share and what permissions to grant on this information.

Sharing with identity and/or groups
Through the PCHR a patient should be able to share his health information with his family, friends and health care providers. It may be the case that the patient wants to share his information with a practice rather than a specific provider at that practice. How can the patient trust this practice to enforce local access rules and protect his privacy?

Delegation of sharing rights
When sharing information with a provider, this provider may wish to share with other providers e.g. to get a second opinion. If this is allowed the patient loses control over who has access to his information. However, if this is not allowed it means the provider has to ask the patient to grant access to another provider, before they can interact. This appears inconvenient.

Information sensitivity
All health information is considered sensitive. However, some types of information, e.g. related to psychiatry and STDs, may be considered to be more sensitive. Information classification may be used to limit what information is shared. The main questions here are: who determines what information is ”more sensitive”? And how do we label this information?

4.2 Time

Time is a central issue when sharing information. The issue is the patient's understanding of what is shared. Is it:

- a snapshot of the record at the time of sharing?
- only historical information?
- or the evolving record also as new information is added?

Another issue related to time, is *for how long* the record is shared. Is it forever or only for a limited time period? And what happens when that time period is over?

4.3 Transparency

In a PCHR the users are given control to set access permissions. Tasks that are usually performed by system administrators are shifted to the users. To ensure that the users are able to take advantage of the possibilities the system presents without compromising their privacy, it is important to strive to keep the users informed about the consequences of their decisions to share. In other words the consequences of actions should be transparent to the system users. The question is how to keep the users informed.

Informed sharing
The patient needs to be informed about the consequences of sharing. The key here is to keep the patient informed of exactly what sharing means, so he can make informed decisions. Consequences of sharing should be obvious and the process of assigning access transparent.

Auditing
Extensive auditing is important to ensure traceability of actions. Audit logs are usually only accessible to system administrators. When the user is in charge of his own information, he should also be able to trace the actions of people he is sharing with. Auditing system logs is a formidable task. Providing the user with insight into access logs may help making this task easier, assuming that the user has a special interest in keeping track of who does what with his health information. The key challenge is to make the audit logs accessible and understandable to the patient.

5 An Initial Access Control Model for a PCHR

In this section we present our initial access control model for a PCHR. The model is very much unfinished, but we have chosen to put it forth to generate discussion on the subject. There are many aspects of access control for a PCHR that needs to be explored further. In this section we provide a sketch of what we consider most important to focus on, and how we think this may be solved.

5.1 Simplicity

Simplicity and transparency is key when the patient is put in charge of assigning access to other users. It should be easy to do and the consequences of sharing, i.e. what permissions are granted, should be obvious. Assigning permissions by selecting from a list of system-defined roles is arguably a very simple way for a user to assign permissions, but there is a tradeoff between simplicity and specificity that needs to be considered. To allow for flexibility and adaptability in privacy requirements, the user should be allowed to change a role and/or create new roles with specific permissions to fit his needs. Ideally we should like to provide both simplicity through role selection and complete specificity in selection of permissions for a user. To keep the model clean and consistent we propose an approach that allows us to do both, and keep the role as the base element for access assignment.

Central to the concept of RBAC is the fact that there is an administrator that is responsible for role creation. Certain models propose allowing the user some control on role assignment and delegation [9] [15], but not on role creation or adaption. In a PCHR the patient *is* the administrator. Though this model takes advantage of the base principles of RBAC, the viewpoint of the administrator is fundamentally different from the usual RBAC case.

Figure 1 provides an overview of the main elements of the proposed PCHR role model. There are two main classes of roles:

- System roles: are assigned when a user is created in the system. There are only three system roles: patient, provider and researcher. These roles are used to restrict the functionality available in the system, and as a control when sharing. E.g. a patient may only assign the role of primary care provider to a user with the system role ”provider”.
- User roles: are assigned by the patient to other system users he wants to share his record with.

The system roles are straightforward. The user roles are more complex. As Figure 1 shows, the user roles are structured as follows:

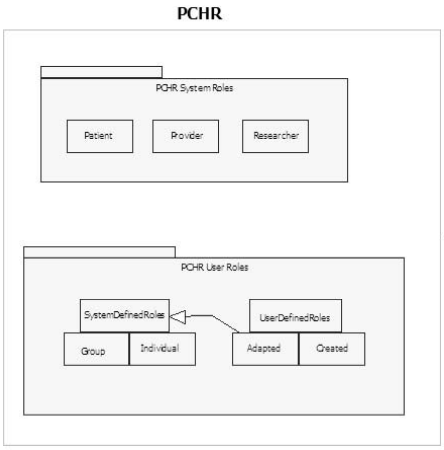

Figure 1. PCHR role model

- A set of system-defined user roles. These roles are grouped into individual roles and group-roles. Group roles allows sharing with a practice. The set of individual roles should be fairly small, not more than 10. T
- User-defined user roles. These are specific to each user. The system maintains a repository for each user of the roles he have created and assigned to other users to share his record. These roles may be based on a template (adapted from a system-defined role) or created from scratch. A user-defined role may also be based on one of the users previously defined roles. The user is responsible for giving the roles he defines meaningful names. Once a role is defined, the user may use this role many times and assign it to several people.

The model adopts the principle of permission inheritance from RBAC. The system-defined roles are structured in a hierarchy where lower level roles inherit permissions from higher level roles. The further down in the inheritance hierarchy, the more specific the role is. For instance one may have a parent role *provider* that specifies permissions common to all providers, and then a role primary care provider that inherits all the permissions of the general provider role, and adds permissions that are specific to a primary care provider. When looking up permissions for a role, the policy of the target role is combined with the policies of its parent roles.

To provide specificity, we want the PCHR users to be able to subtract permissions as well. When basing a new

role on an existing one it should be possible to make it more restrictive in some areas, and add more permissions to others. This means that the rule when combining roles for inheritance is "deny overrides". If one of the roles denies the desired action, then the decision is deny.

In the case of a PCHR, a user may well be assigned to a number of roles. *But* each role is linked to a specific record. In other words, when a patient is sharing his record he selects, or creates, only one role for this user. The assignment of a user role is specific to a record.

One of the main pros of RBAC is the flexibility it provides. To update the permissions of many users one only has to change the role they are all assigned to. This is a task usually performed by a system administrator. As previously stated, for the case of PCHR the patient is the administrator. He can change the roles he has defined as he wishes and these changes are reflected in the permissions of the user he his sharing his record with. However, as Figure 1 shows the user-defined roles may be connected to the system-defined role that was a template for creation of this role. This link is permanent through an inheritance relationship. This means that any change in a system-defined role is reflected in all user roles that is related to this role through an inheritance relationship. We therefore need to be careful about allowing updates of system-defined roles. If the role changes because of an identified error in the definition or due to changes in the law, these changes should propagate and be effective throughout the system. However, if the change is a less severe update we may want to keep links to the old version of the system-defined role. The user only agreed to sharing on the basis of what the role used to be, and it should not be changed without giving the user the option of agreeing or denying to share based on the updated role. To allow for this the concept of history in the role hierarchies is introduced:

- A template-based user role is linked to a version of a system-defined role. If the system-defined role is updated, a new version is created and the links are maintained for user-roles who were based on the previous version.

- Having complete history in both role sets (user-defined and system-defined) ensures that the user can perform auditing and examine who had what permissions to his information at any given time. Without history of roles, one would only be able to tell what role a user had at some time, but could not be sure if the permissions of the role were the same then as now.

- For the same reason a role should never be completely deleted. Keeping history ensures that it is always possible to find a role, even if it is no longer part of the active role set.

5.2 Time

Time is one of the main concerns listed in the previous section. In our model we propose always labeling a role assignment with a start time. The start time is required for useful audit information. In addition it should be possible, but not required, to set an end time of a role assignment.

5.3 Transparency

There currently exists a major push towards patient empowerment in health care. A PCHR system is a great tool for patient empowerment as it enables the patient to exercise control over his health information. However, it also means shifting responsibility and tasks that have usually been performed by educated system-administrators over to the user. "Everyone" is a potential patient, and therefore a potential user of a PCHR. It is safe to assume that these users will span the entire range of computer-literacy. Many will be well accustomed to the use of online services for information management like online banking, while many others will have little experience with web-based systems.

A PCHR is intended to contribute to the patients sense of empowerment and control. To achieve this, the process of sharing, and auditing the shares, must be intuitive and transparent to the user. The goal is to make sure the user is always informed of the consequences of his actions. It is, however, hard to be sure that the user is always informed, and correctly informed. We suggest increasing system transparency by:

- Using graphics to visualize to the patient the consequences of assigning a role. Figure 2 illustrates what this might look like in a PCHR system, illustrated by an adaption of the current Indivo-GUI. When the patient selects a role the permissions of this role are visualized in a cube representing the information in the record. A dark field with grey characters illustrates no access, a lighter shade and black characters illustrates read access, while a white field illustrates edit access.

- Using a graphic notation to visualize an overview of current or historical shares and their permissions.

- Also using this graphic notation when a user creates or adapts a role. E.g. by allowing a user to drag-and-drop a document from the sidebar menu onto the cube to add it to a sharing.

- Presenting the user with a view of the audit log. This allows the user to examine who has access to his record and their actions.

Examining access logs is often an overwhelming task. Still many systems rely on auditing of log data as a security

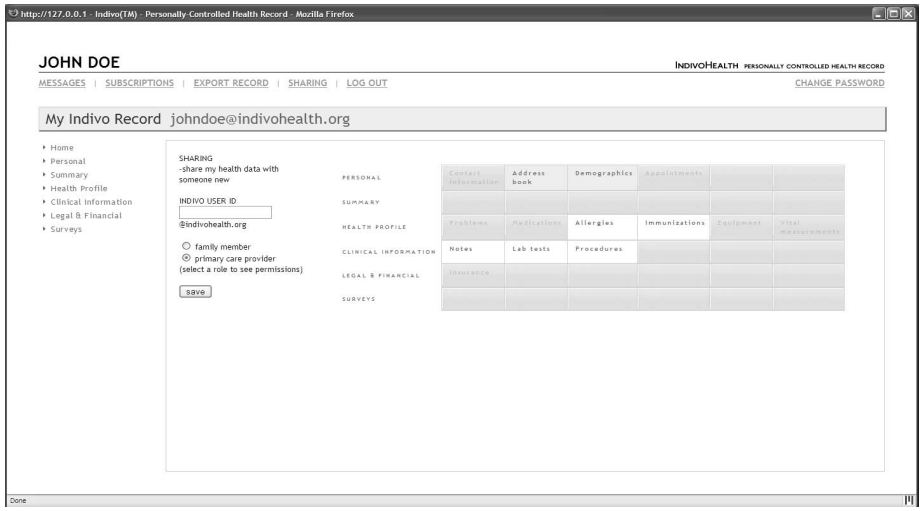

Figure 2. Example graphic representation of access rights

mechanism. However, a study [16] indicates that these log data are seldom used. By distributing this task to let each patient be responsible for auditing the access log for his own record, we reduce the workload. Our assumption is that the patient is both more motivated and knowledgeable to be able to detect any misuse of his health information. Having access to the audit log also serves an educational purpose as examining other user's actions gives the patient insight into what the people he is sharing with are allowed to do with is information.

6 Conclusion and Future Work

In this paper we have presented a discussion of issues related to access control in a PCHR. Based on this discussion we have explored some aspects of what such an access control model would be like. However a lot of work certainly remains. In our model we have made some assumptions that needs to be explored further. For instance, we assume that most users, most of the time, will prefer sharing by selecting a role from a list. We also assume that most users will not want to set up specific permissions each time they share. Though this appears to be a valid assumption, user test should be performed and usage data collected to see if this assumption holds. We intend to continue work on the model with the goal of creating an implementation that can be tested and evaluated properly. Our work will focus on defining a default role set and continue exploring methods for increasing the transparency of the system.

7 Acknowledgments

The author would like thank the Indivo-team at the Children's Hospital Informatics Program in Boston for the opportunity to learn from their work.

References

[1] R. Hillestad, J. Bigelow, A. Bower, F. Girosi, R. Meili, R. Scoville, and R. Taylor. Can electronic medical record systems transform health care? potential health benefits, savings, and costs. *Health Affairs*, 24(5):1103–1117, 2005.

[2] W. R. Hersh. Medical informatics: Improving health care through information. *The Journal of the American Medical Association (JAMA)*, 288(16):1955–1958, 2002.

[3] Connecting for health: The personal health working group final report. Technical report, Markle Foundation, July 1 2003.

[4] P. C. Tang, J. S. Ash, D. W. Bates, J. M. Overhage, and D. Z. Sands. Personal health records: Definition, benefits, and strategies for overcoming barriers to adop-

tion. *Journal of the Ammerican Medical Informatics Association*, 2005.

[5] M. I. Kim and K. B. Johnson. Personal health records: Evaluation of functionality and utility. *J Am Med Inform Assoc*, 9(2):171–180, 2002.

[6] K. D. Mandl, P. Szolovits, and I. S. Kohane. Public standards and patients' control: how to keep electronic medical records accessible but private commentary: Open approaches to electronic patient records commentary: A patient's viewpoint. *BMJ*, 322(7281):283–287, 2001.

[7] D. F. Ferraiolo, D. R. Kuhn, and R. Chandramouli. *Role-Based Access Control*. Computer Security Series. Artech House Publishers, Boston, 1 edition, 2003.

[8] B. Blobel. Authorisation and access control for electronic health record systems. *International Journal of Medical Informatics*, 73(3):251–257, 2004.

[9] L. Zhang, G.-J. Ahn, and B.-T. Chu. *A role-based delegation framework for healthcare information systems*. Proceedings of the seventh ACM symposium on Access control models and technologies. ACM Press, Monterey, California, USA, 2002. 507731.

[10] K. Beznosov. *Requirements for access control: US Healthcare domain*. Proceedings of the third ACM workshop on Role-based access control. ACM Press, Fairfax, Virginia, United States, 1998.

[11] M. Evered and S. Bgeholz. *A case study in access control requirements for a Health Information System*. Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation - Volume 32. Australian Computer Society, Inc., Dunedin, New Zealand, 2004.

[12] A. N. S. Institute. American national standard for information technology: Role based access control. Technical Report ANSI INCITS 359-2004, 2004.

[13] W. W. Simons, K. D. Mandl, and I. S. Kohane. The ping personally controlled electronic medical record system: Technical architecture. *J Am Med Inform Assoc*, 12(1):47–54, 2004.

[14] I. S. Kohane, K. D. Mandl, P. L. Taylor, I. A. Holm, D. J. Nigrin, and L. M. Kunkel. Medicine: Reestablishing the researcher-patient compact. *Science*, 316(5826):836–837, 2007.

[15] S. Na and S. Cheon. *Role delegation in role-based access control*. Proceedings of the fifth ACM workshop on Role-based access control. ACM Press, Berlin, Germany, 2000. 344300.

[16] L. Røstad and O. Edsberg. A study of access control requirements for healthcare systems based on audit trails from access logs. In *22nd Annual Computer Security Applications Conference (ACSAC'06)*, pages 175–186, Miami, Florida, 2006. IEEE.

Paper I

Personalized Access Control for a Personally Controlled Health Record

Personalized Access Control for a Personally Controlled Health Record

Lillian Røstad
Department of Computer and Information Science
Norwegian University of Science and Technology
Trondheim, Norway
lilliaro@idi.ntnu.no

Øystein Nytrø
Department of Computer and Information Science
Norwegian University of Science and Technology
Trondheim, Norway
nytroe@idi.ntnu.no

ABSTRACT

Access control is a key feature of healthcare systems. Up until recently most healthcare information systems have been local to a healthcare facility and accessible only to clinicians. Currently there is a move towards making health information more accessible to patients. One example is the Personally Controlled Health Record (PCHR) where the patient is in charge of deciding who gets access to the information. In the PCHR the patient is the administrator of access control. While it certainly is possible to create roles representing people most patients would want to share with, like primary physician, it is also likely, and desirable, to afford the patients a high level of control and freedom to be able to create specialized access policies tailored to their personal wishes. We entitle this *personalized access control*. In this paper we present a semi-formal model for how we believe personalized access control may be realized. The model draws on and combines properties and concepts of both Role-Based Access Control (RBAC) and Discretionary Access Control (DAC) to achieve the desired properties. Throughout the paper we use the PCHR as a motivating example and to explain our reasoning and practical use of the model.

1. INTRODUCTION

Access control is a key feature of healthcare systems. Enforcing access control on sensitive health data is about protecting the patient's privacy as well as ensuring that clinical personnel have access to the information they need to provide the best possible care. Access control has a unique challenge in that it is always most important to save the patient's life. In other words: though confidentiality is the norm - availability takes precedence when the patient's health is at stake.

A challenge in healthcare today is the lack of connectivity and sharing. Information exists in proprietary information systems local to hospitals or doctors offices and accessible only to health care personnel. Personal Health Records (PHR) have been proposed as a potential solution to this problem. The term PHR has been defined by The Markle Foundation as:

> "An electronic application through which individuals can access, manage and share their health information, and that of others for whom they are authorized, in a private, secure, and confidential environment." [1]

The challenge with most PHRs is that they are local and specific to one point of care [7] and therefore most existing PHRs only contain a subset of a patient's clinical information. The Personally Controlled Health Record (PCHR) has been proposed as a possibility that has the potential to solve many of the PHR's shortcomings. The goal of a *Patient/Personally Controlled* Health Record (PCHR) [8] is to assemble the patient's complete health history by importing data from many source systems. A PCHR differs from a PHR in that it exists outside of organizational boundaries and contains data from multiple care sites. Also, the patient is in complete control of the information in the PCHR. The patient decides what data should be added to the PCHR. Any data import has to be approved by the patient. The patient also decides who gets access to the information in the PCHR. This means that it is the patient who is administrator of access control [11]. Through a PCHR the patient may choose to share his data with health care providers, family members and any other as needed.

One of the main challenges of the PCHR is the duality of empowerment potential and privacy risk. The patient is empowered in that he is given control over his own health information. But, it may also increase the risk of inadvertently leaking sensitive information about himself as the patient is solely in charge of assigning access rights and maintaining these over time. This means that it is important to have an access control model that is easy to use, hard to misuse, yet affords the patient a high-level of flexibility and control.

In many healthcare systems today, Role-Based Access Control (RBAC) is the norm. Healthcare organizations fits very well the RBAC premises of having many users that can be grouped into a relatively small number of roles. One may argue that for a PCHR this is also true. For instance, most people would likely want to give their primary physician access to their PCHR. However, it is also likely that given the

opportunity to share with anyone, many users will construct access policies that are personal, unique and not generalizable. As such there is a need for a model that has both a pre-defined, common set of access policies, for convenience, yet allows the patient absolute control when desired.

In this paper we present a semi-formal model for what we have entitled *personalized access control*. The model is motivated by our work on personal health records [11] and the PCHR is used as a motivating example throughout the paper. The model is semi-formal in the sense that some properties still requires some more discussion and there remains some issues to be resolved.

2. THE PERSONALLY CONTROLLED HEALTH RECORD

A PCHR is a collection of clinical information about a patient [8]. What's unique about the PCHR is that the patient is in charge of deciding who gets access to this information by assigning sharing privileges. In this section we provide some usage scenarios to help explain the PCHR concept in more detail and how it will be used. These examples will be used for explanation throughout this paper as we move from requirements to a more detailed description of personalized access control in a PCHR.

PCHR usage scenarios

1. A patient moves from one city to another. She decides to give her new primary physician access to her PCHR so he can read up on her medical history before their first appointment. She also decides that he should be able to add information to her PCHR, so she will have a complete medical history there in case she has to move again.
2. A patient that has been healthy most of his life, suddenly is diagnosed with a complex disorder. This diagnosis implies that he will from now on need regular services from many health care providers including a physical therapist, an orthopaedist and an occupational therapist in addition to his primary physician. To provide the best care it is helpful if all the service providers are aware of and informed about the other services he receives and how they are progressing. The patient decides to set up a PCHR and grant all of his providers access to read the information in his PCHR.
3. A young girl has had a PCHR for a while. The girl is now 17 and still not legally an adult, but as she is considered an adolescent, she is in control of the PCHR and her parents currently do not have any access. One day she has an accident on the way to school an breaks her leg. The X-ray summaries and the doctor's notes are added to the PCHR as is routine. Her mother is concerned and asks if she can get access to the PCHR so she can read the information. Using the PCHR it is possible for the girl to give her mother access only to the parts of the PCHR that she considers ok to share. The mother is never aware of what she cannot see.

3. REQUIREMENTS FOR PERSONALIZED ACCESS CONTROL

Based on the previous section, we can formalize a set of requirements for our model for personalized access control for a personally controlled health record:

1. The patient is the owner of information in the PCHR.
2. Every information element in the PCHR database is owned by somebody.
3. Any information element in the PCHR database has only one owner.
4. The patient is administrator of access to his/her information. The patient decides what permissions to assign to who.
5. Information in the PCHR is structured in categories. Examples of categories include: lab results, clinical notes, immunizations etc.
6. Every information element in the PCHR belongs to a category.
7. Permissions may be granted on a category or a single information element.
8. Permissions are granted by assigning an access policy to another user. An access policy is a set of permissions.
9. For ease of use it should be possible to define a set of access policies believed to be common to most users (patients).
10. For reuse purposes it should be possible for the patient to create personal access policies.
11. For simplicity it should be possible for the patient to create a new access policy by adapting one of the common policies to his/her specific needs, or by extending or adapting one of his/her personal policies.
12. The patient should not be allowed to update or delete the common access policies.
13. For flexibility the patient should be allowed to update or delete any of his/her self-defined access policies at any time.
14. The patient may at any time revoke an assigned access policy.

4. RELATED WORK

To the best of our knowledge, no model with these exact properties have been proposed before. However, there exists work that has similarities. Most notably there are similarities to both Role-Based Access Control (RBAC)[2] and Discretionary Access Control (DAC). The concept of access policies in our model is similar to roles in RBAC. We will reuse many of the RBAC properties for combining and applying roles to our access policies. We have chosen to use the term *access policy* rather than *role* because in our model an access policy may be personal and specialized while a role in RBAC is supposed to be generalized and "define once - apply many".

In the RBAC standard [2] a role is defined as: *(..) a job function within the context of an organization with some associated semantics regarding the authority and responsibility conferred on the user assigned to the role.* In this traditional definition of a role, the role describes a relation between a person/set of persons and a set of objects. In our work, an access policy represents a relationship between two persons - the owner of the information in the PCHR and the person she is sharing her information with. This is significantly different from standard RBAC where a role is a mapping from a user to allowed actions on data.

Our work has similarities with DAC in that information has an owner, and the owner has discretionary authority over who else can access that information. In [10] Osborn et al. presents how DAC may be implemented using RBAC. In their approach they create an owner role that is associated with each object, and declares ownership by assigning this role to a user. We will not adopt this approach since, as already stated, we are not using RBAC directly and also because while that approach shows that it is possible to implement DAC using RBAC it is not uncomplicated. Also, in DAC it is usually the case that the owner of an object is the one who created the object. For our model the owner is the one the information is about. The owner may allow others to add information, but the information created belongs to the owner of the record it is part of.

In our model we need to allow negative permissions. That is, we want to be able to combine access policies to create an adapted policy that contains most of the permissions of the policies it is based on, but with some exceptions. Negative authorizations have been proposed in [3] where attribute expressions are used to prevent a user from being able to assume a role. The issues of potential conflict in negative authorizations are relevant for the use of negative permissions in our model.

5. PERSONALIZED ACCESS CONTROL (PAC)

In this section we present the core components of our model for personalized access control. From this point on we will use the abbreviation *PAC* for Personalized Access Control.

We start out by defining an access policy as:

Definition 1. An **access policy** is a representation of a relationship between two people. This relationship is reflected in the permissions one user (the owner) grants another user through policy assignment.

Throughout the remainder of this paper we will assume *access policy* and *policy* to have the same meaning. We begin our discussion by elaborating on some of the requirements and from that we construct the core PAC model.

Central to the PAC model is the concept of ownership of information. Every information element in a PCHR is owned by the patient and only the owner may decide who to share information with. In other words only the owner has the power to assign and revoke permissions. And the owner can of course only share her own information. As stated in the requirements an information element has to have one, and only one, owner. Any information created in or added to the PCHR is owned by the patient: ownership is not linked to who creates information, but to who owns the PCHR the information is part of.

A PCHR may over time grow very large. Therefore it does not seem like a good solution to only have the possibility of setting permissions on single information elements. However, we may take advantage of the fact that most healthcare information is heterogeneous, often with complex structure, types and relationships. Information is often grouped by topic - e.g. doctor's notes, immunizations, x-rays etc. The specific information may be complex or simple, we just need a category tag to identify parts of the structure. A category "personal information" may subsume another category "allergies". Note that access to "personal information" and access to "allergies" may conflict, and can only be resolved by taking the information structure into account. The actual meaning of the permissions given by a policy will thus have to be interpreted according to the information model. To simplify the PAC model, for now, we simply state that any information must belong to a category. Note that no restrictions are placed on the number of categories an information element may belong to. This has to be included for practical reasons, though it does lead to some complications when combining and interpreting policies that we will discuss further later on in this paper.

As stated in the requirements, to allow for specific control and high granularity, in the PAC model it is possible to grant permissions on both specific information elements and categories. The assumption is that most of the time granting permissions on categories is sufficiently detailed. Permissions on information elements will probably only be used in specific situations e.g. like in the example of the girl with the broken leg. In general she wants her mother to see her lab results, but not the ones related to the abortion. Also, the existence of categories makes it possible to construct generalized policies, or policy templates, that can be reused.

To fulfill requirements 9-13, we need to construct two sets of access policies in our model. We denote these two policy sets *common policies* and *personal policies*. The common policies are not changeable by the patient while the patient is in complete control of the personalized policies. The set of common policies, describing common relationships, should be system-wide and available to all users. As such there exists only one common policy set while there are just as many personalized policy sets as there are users that are information owners in the system, as depicted in figure 1. Note that the personal policy set for a user may be empty. We will return to the personal and common policies and discuss them in more detail after we have defined the core PAc model.

In the PAC model an access policy, common or personal, may exist without being assigned to anyone. An unassigned policy represents a potential relationship. Assigning the polcy to a user means establishing a relationship between the assigner and the assignee. An example of a potential relationship is that of a *primary physician* as described in scenario 1. This is a relationship that most people have with one person. Most people would probably also agree that

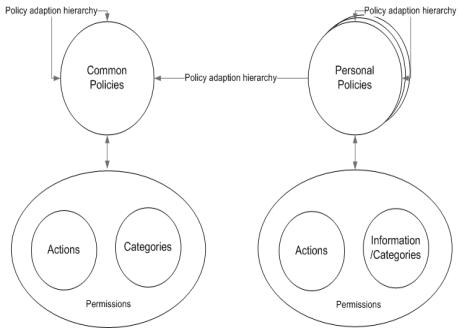

Figure 1: Common and personal policies

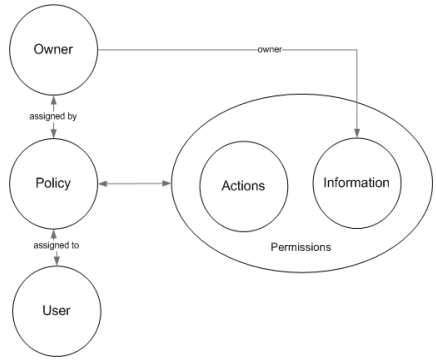

Figure 2: Core PAC Model

your primary physician should have access to most of your clinical information and also be able to add information. It is probably possible to create a common access policy for primary physicians with a minimal set of permissions that most people would agree is representative and appropriate. But it is only when the owner of a PCHR assigns this role to a person that the relationship *user u is primary care provider for owner o* is established.

There are mainly two sets of users: regular users and owners. Not every user has to own data. But only users owning data have a personal policy set and the ability to share their information.

5.1 Core PAC model

With this in mind, we start out by defining the core concepts of the PAC model more formally. Then we move on to discussing policy definition, assignment and revocation in more detail. Figure 2 depicts the Core PAC Model. The figure shows how the relationship between two users is established by policy assignment. It also shows how a permission, in an assigned policy, is a set of allowed actions on information elements and that there is a *owner* relationship linking a user (the owner) directly to information elements. This model is similar to the core RBAC model [2]. The main difference is the introduction of ownership and that a policy links two users. Figure 1 illustrates that the model consists of two policy sets: a set of policies that are common and known to all users and a set of personal policies that are specific to one user.

From this we define the core components of the PAC model:

- A set U of *users*
- A set O of *owners* where $O \subseteq U$
- A set C of *categories*
- A set I of *information elements*
- A set A of *actions*
- A set CP of *category permissions*
  $CP = \{(a, c) | a \in A, c \in C\}$
- A set IP of *information permissions*. Permissions on single information elements.
  $IP = \{(a, i) | a \in A, i \in I\}$
- A set $CPol$ of *common policies*
- A set $PPol$ of *personal policies*
- A *common policy* is a set of category permissions, i.e. $CPol \subseteq CP$.
- A *personal policy* is a set of category permissions and information permissions, ie. $PPol \subseteq CP \times IP$.
- A function ci from a category to the set of all information elements (possibly empty) belonging to that category.
- A function io from an information element to the (unique) user owning that information.

The concepts of common and personal policies, and hierarchies of such, are described in more detail in the following sections.

5.1.1 Common policies

Common policies form policy hierarchies. These hierarchies differ from hierarchical RBAC in that they are not hierarchies under subsumption of a set of permissions, but policies related by being derivable from each other from root to node according to simple rules of addition and removal of permissions. For example, the policy "Significant other" could be related to the superior policy "Family" by adding access to all information of category "Medication" but explicitly remove (just in case it would be accessible) all information in category "Childhood diagnoses". Thus the hierarchies form by virtue of the order that permissions are removed and added. We denote this an *adaption hierarchy*.

It is worth noting here what adapting means. In RBAC, when one role extends another we say that there is an inheritance relationship between the roles. The RBAC standard [2] defines a role inheritance relationship as: *role* r_1 ***inherits*** *role* r_2 *if all privileges of* r_2 *are also privileges of* r_1. In

the case of PAC we want to be able to base one policy on another, but we also want the flexibility to allow this new policy to be both *wider* and *more narrow* than the policy it is based on. In other words we want to be able to both add and subtract privileges, which means that we need to be able to specify negative privileges. Usually, when roles are combined the default rule is *permit overrides*. If one of the roles that are combined allows, then the result is allow. Negative privileges are only represented as the absence of a privilege, and the resulting permission set is the combination of all privileges in the roles that are combined. An actual role is just a set of (positive) category permissions resulting from this process. The permissions of an *adapted policy* is calculated by adding together all the positive permissions, removing a permission if any of the participating policies has a negative permission for this category or information element. As for roles the result of this calculation is a set of only positive permissions, and absence of a permission implies no permission. So the result of policy adaption is the same as when role hierarchies are collapsed, but the calculation process is different. In the calculation process for policy adaption the rule for policy combination is *deny overrides*.

5.1.2 Personal policies

Every user that owns information potentially has a set of personal policies. The set may be empty. As Figure 1 shows, the personal policies differ from the common policies in that they are a mapping of allowed actions on *owned information elements* and/or categories. A personal policy may be more specific and detailed than a common policy. This is necessary to cover those situations where a patient wants to share some of the information belonging to a category, but not all. For instance a patient may want to share knowledge of some of her test results with her mother, but not all of them, as illustrated in example 3.

As for common policies we also use the concept of *policy adaption* for personal policy. A personal policy may:

- Be an independent entity.
- Be based on one or more personal policies by an adaption relation.
- Be based on one or more common policies by an adaption relation.
- Be based on common *and* personal policies by adaption relations.

Figure 3 provides an example adaption policy to illustrate the concept. The top node is the empty set. Note that the set contains two sub-sets: the set of positive permissions and the set of negative permissions. Each policy in the hierarchy consists of a positive and a negative permission set.

5.2 Policy definition

Policy definition is about creating a set of permissions and declaring adaption relationships to other policies. Defining a common policy is simpler than a personal policy because we only deal with categories and there is no concept of ownership. For definition of personal policies we also need to include individual information elements related to an owner. We need to state some rules for adaption relationships on policy definition:

- Adaption relationships form a lattice.
- Every element in the lattice is composed of two sets: a set of positive and a set of negative permissions.
- A permission is an allowed action on a category or an information element.
- The resulting policy is calculated by collapsing the adaption hierarchy in the definition. The last element to be added is the set of specific permissions defined for the policy to be calculated.
- *Deny overrides* is used as the rule for calculation. If any policy denies a permission, then that permission is left out of the resulting calculation.
- The resulting policy is a set of positive permissions.

5.3 Policy assignment

In PAC policy assignment is interpreted as *relationship declaration*. Unassigned common and personal policies may exist. An unassigned policy is simply a policy definition. Figure 2 illustrates relationships in PAC. A relationship is a direct link between two users established through a policy assignment.

An owner assigns a policy (declares a relationship) by:

- Assigning a common policy.
- Assigning a personal policy.

Note that the assigned policy may depend on any number of other policies by definition.

A policy assignment is a one-to-many relationship between an owner and other users. An owner may share her PCHR with many other users. An owner may also assign multiple policies to the same user. This is required to handle situations like when one person is both the father of and e.g. physiotherapist for one patient. In multiple policy assignments the permissions of the policies are combined. For this combination we apply the rule of *permit overrides*, and as such it is different from policy adaption. The resulting permissions are the sum of permissions in all assigned policies. If the policies to be combined are themselves defined in terms of adaption hierarchies, the adaption calculations are performed first and then the resulting policies are combined. Remember that the result of calculating an adaption is a policy containing only positive permissions. We will illustrate this process by example in the next section.

5.4 Policy activation

A policy is activated when a user applies the policy to access information. A policy definition is simply a declaration of how one policy is related to others, and what to add or remove specifically for this policy. An assignment is simply a link between two users in the form of a policy. Only upon

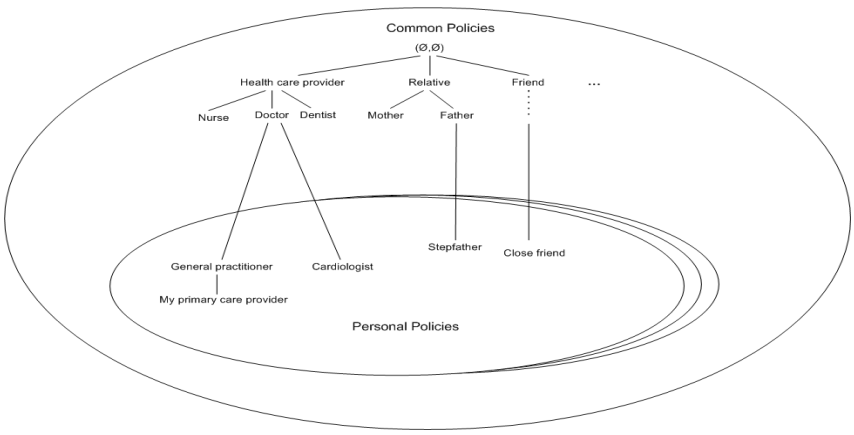

Figure 3: An example policy adaption hierarchy

policy activation, when the policy is applied, is the policy definition evaluated and calculated and the relationship confirmed. For this we need a set of steps for calculating and applying the permissions of the policy to be applied:

1. Calculate the permissions formed by the adaption hierarchy by: first calculate adaption hierarchies formed by common policies - collapse any positive permissions together and do the same for negative permissions (two identical positive permissions results in a positive permissions, two identical negative permissions results in a negative permission etc - the presence of a negative permission at any point results in a removal of the corresponding positive permission if it is present), then take the result of this operation and do the same with any personal policies that are part of the adaption hierarchy. The result will be a set of negative permissions that is the union of all negative permissions in the adapted policies, and a set of positive permissions that is the set of all positive permissions in the adapted policies for which no corresponding negative permission exists in any of the adapted policies.

2. Calculate the intermediate policy by adding any positive permission specific to this policy for which there exists no negative permission, and by adding any negative permission that is not already part of the negative permission set.

3. Calculate the resulting policy by removing the negative permission set. The policy to be applied only consists of positive permissions. When the policy is applied, the absence of a permission is interpreted as *no access*.

4. *If* more than one policy is assigned: repeat the above steps for all assigned policies. Combine the policies by calculating the union of the permissions in all the assigned policies. Again the result is a set of only positive permissions.

This process is repeated any time a policy is applied. This may seem cumbersome and inefficient, but it affords flexibility in that any policy may be updated at any time and those changes will be reflected the next time any policy that is adapted from this one is applied. This allows great flexibility in the model.

5.5 Policy update

As stated in the requirements the personalized policies may be changed by the owner at any time. Changing or updating a policy includes adding or removing specific permissions or adding or removing policies from the policy adaption set. As stated above, the actual policy to be used is recalculated every time it is applied and as such any change will be reflected immediately. Though this approach results in a very flexible model, it also results in potential problems. If the owner changes one of her policies – is it safe to assume that she is able to grasp all the consequences of this action? Updating one policy affects all policies that are related to this one. Deleting a policy also has a cascading effect that it is difficult for the user to foresee. Potential solutions to this problem are:

- Do not allow a user to change a policy when it has been defined. This is not desirable.
- Keep a complete history of policies. If a user updates one policy that only affects policies defined after this. A copy of the old policy is kept and any pre-existing policies keeps their relationship to the old version. This is safer, but may not be what the user expects to happen.
- Only allow updates of policies that no other policy depends upon.

None of these possibilities are ideal, and work remains on how policy updates should be defined in the model.

5.6 Policy revocation

Relationships are not permanent. A patient may switch to a different doctor, visit another hospital etc. Even social relationships like family and friends may not be permanent. In this model the process of revocation is simple: it simply involves the owner removing the policy assignment. However, considering the motivating case, the consequences of revocation are not so simple. Assuming that the owner is in complete control she can deny anyone access. Depending on how the PCHR is used this may have unfortunate consequences. If the primary physician has relied on the PCHR to get information about other care the patient receives, suddenly losing access may result in lesser quality of the care he may provide. Still it is at the patient's discretion to do so. While we consider these to be valid considerations, we also consider this to be out of the scope of what can be included in a model for personalized access control but certainly an issues that needs to be resolve when the model is to be realized.

6. DISCUSSION

One of the main purposes of the PAC model is to allow the owner the power to define very specific permissions when sharing her information with someone, while at the same time preserving the flexibility provided by RBAC. Rather than having inheritance hierarchies as in standard RBAC, PAC has adaption hierarchies where it is possible to both add and subtract permissions. While this is an important property of the model, it also increases complexity by introducing the possibility for conflicting authorizations.

There are also issues that are outside the scope of this model, but nevertheless are important to mention. Many of these were summarized in an earlier paper [11] but we repeat some of them here as they are important to consider. The introduction of PCHRs is a step on the way to *patient empowerment*. Through a PCHR the patient is given complete control over who gets access to her health information. But this also implies an increased responsibility for the patient. It is important to consider how to achieve true empowerment. How do we make sure that the patient understands the consequences of every access decision she makes? Usability becomes an important feature of any implemented access mechanism based on PAC to ensure that the intentions of the model are achieved.

7. CONCLUSIONS

In this paper we have presented a model for personalized access control for use in a personal health record. We strongly believe that the ideas put forward here are important and bring something new to the access control field. In our future work we will focus on unresolved issues, exploring the model for various cases to make it more generic and creating a reference implementation.

8. ACKNOWLEDGMENTS

The authors would like to thank the Indivo team at the Children's Hospital Informatics Program in Boston for the opportunity to learn from, and be inspired by, their work.

9. REFERENCES

[1] Connecting for health: The personal health working group final report. Technical report, Markle Foundation, July 1 2003.

[2] American national standard for information technology - role based access control. Technical Report INCITS 359-2004, American National Standards Institute, Inc., 3 February 2004.

[3] M. A. Al-Kahtani and R. Sandhu. Rule-based rbac with negative authorization. *Computer Security Applications Conference, 2004. 20th Annual*, pages 405–415, 6-10 Dec. 2004.

[4] J. Barkley, K. Beznosov, and J. Uppal. Supporting relationships in access control using role based access control. In *RBAC '99: Proceedings of the fourth ACM workshop on Role-based access control*, pages 55–65, New York, NY, USA, 1999. ACM.

[5] B. Blobel. Authorisation and access control for electronic health record systems. *International Journal of Medical Informatics*, 73(3):251–257, 2004.

[6] C. K. Georgiadis, I. Mavridis, G. Pangalos, and R. K. Thomas. Flexible team-based access control using contexts. In *SACMAT '01: Proceedings of the sixth ACM symposium on Access control models and technologies*, pages 21–27, New York, NY, USA, 2001. ACM.

[7] M. I. Kim and K. B. Johnson. Personal health records: Evaluation of functionality and utility. *J Am Med Inform Assoc*, 9(2):171–180, 2002.

[8] K. D. Mandl, P. Szolovits, and I. S. Kohane. Public standards and patients' control: how to keep electronic medical records accessible but private commentary: Open approaches to electronic patient records commentary: A patient's viewpoint. *BMJ*, 322(7281):283–287, 2001.

[9] S. Na and S. Cheon. *Role delegation in role-based access control*. Proceedings of the fifth ACM workshop on Role-based access control. ACM Press, Berlin, Germany, 2000. 344300.

[10] S. Osborn, R. Sandhu, and Q. Munawer. Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Trans. Inf. Syst. Secur.*, 3(2):85–106, 2000.

[11] L. Røstad. An initial model and a discussion of access control in patient controlled health records. In *The International Workshop on Privacy and Assurance (WPA-2008)*, Proceedings of the The International Conference on Availability, Reliability and Security (ARES 2008), Barcelona, Spain, 2008. IEEE Computer Society.

[12] M. Wilikens, S. Feriti, A. Sanna, and M. Masera. A context-related authorization and access control method based on rbac. In *Symposium on Access Control Models and Technologies*, pages 117–124, Monterey, California, USA, 2002. ACM.

[13] L. Zhang, G.-J. Ahn, and B.-T. Chu. *A role-based delegation framework for healthcare information systems*. Proceedings of the seventh ACM symposium on Access control models and technologies. ACM Press, Monterey, California, USA, 2002. 507731.

Paper J

Visualization for Patient-Administered Access Control: a Usability Study

Visualization for Patient-Administered Access Control: a Usability Study

Abstract

Patient-Controlled Health Records (PCHRs) allow patients complete control over their health information. They decide who to share their information with, which makes the patient the administrator of access control. While PCHRs have a great potential for patient empowerment, they have an equally great risk for breach of privacy if consequences of sharing are not completely clear to the patient. This paper presents results from a usability study that compares three different visual interfaces for sharing in a PCHR. The goal of this study was to evaluate if a visual interface can help make the effects of sharing more transparent to the patient.

1 Introduction

A major challenge in healthcare today is the fractured nature of clinical information. A patient's health record does not exist as one single unit, but rather as small fractions of information scattered among many information systems and accessible only within the organizations where those systems exist. As a result it is rarely the case that a clinician has the complete health history of a patient available when making important clinical decisions.

Personal health records (PHR) have been proposed as a possible solution to this problem. According to the Markle foundation, a PHR is [3]:

> "An electronic application through which individuals can access, manage and share their health information, and that of others for whom they are authorized, in a private, secure, and confidential environment."

A patient-controlled health record (PCHR) [4] is an extension of the PHR concept. Once information has been entered into a PCHR, the patient is considered the owner of that information and is in complete control of deciding who gets access to it, and what they are allowed to do. In other words: in a PCHR the patient is the administrator of access control.

This leads to a number of challenges, the most important of which, and the focus of this paper, is *how to keep the patient informed*. Though a PCHR potentially is a great tool for patient empowerment it may as well lead to increased risk for the patient in terms of privacy breaches. Healthcare information is sensitive. Access control policies are usually created by panels of highly knowledgeable individuals and the administration of those policies carried out by trained experts. Everybody is a potential patient. Therefore the potential user of a PCHR is anybody. The protection of patient privacy therefore relies on the usability of the actual access control, or sharing, interface used in the PCHR. The effects and consequences of sharing should be obvious to the patient. We believe that visualization of data and permissions may help improve transparency.

In this paper we present results from a usability study where we have tested and compared three different sharing interfaces for a PCHR.

2 Related work

Recently, both Microsoft and Google have presented PCHR initiatives. Google Health [1] is an online repository where you can import health information and share with providers. Google Health was launched in early 2008. So far the list of healthcare providers that Google health can connect to and import info from is very limited. When sharing you select from a list of physicians in Google Health. The system is also, for now, only available to people living in the US which makes testing hard for anyone who is not.

The same limitation on availability for testing is true for Microsoft HealthVault [2], which was launched in the fall of 2007.

The added challenge for these PCHRs is that they are created by commercial companies with no historical involvement in healthcare. It remains to be seen if people trust Google or Microsoft to manage their health data.

The Indivo PCHR [5], formerly known as PING, is developed at the Children's Hospital Informatics Program in cooperation between MIT and Harvard. The Indivo PCHR is open-source and freely available and was therefore used as a platform for testing in this study.

3 Study design

The study was performed from Monday April 7th through Wednesday April 9th 2008. This section describes in detail the study setup, including the research hypothesis and rationale for the three demos that were tested.

4 Research goals

The study described here is a usability study. While a PCHR potentially is a great tool for a patient, it is important that the system is usable in that it is transparent and easy to use to fulfill that potential. In this study, the focus is on using simple visualization of information and permissions to enhance usability and transparency in the sharing interface. Three different demos were created and compared. The demos had different designs, utilizing different visualization ideas. In the next section the rationale for the three demos is described in detail.

The main hypothesis of this study was:

> Visualization of information and permissions can increase usability and transparency of the sharing interface in a PCHR.

4.1 The three demos

The three demos were built using a mock-up of the Indivo user interface. The only functional part was the sharing interface, but the rest looked exactly like the standard Indivo interface. The reasoning behind this is that the participants were recruited from the Indivo trial population and were already familiar with that system.

All the demos use the same underlying information structure, functionality and access control model. The three demos simply represent different views on the same things. The information model is the same as used in the Indivo system, as this should be familiar to the study participants. Information is grouped in categories and every category may have sub-categories. For the purpose of this study we only used the permissions "edit" and "read".

Figure 1 show the Indivo sharing interface as it is today. It includes a list of access profiles to choose from. An access profile is a set of permissions and each access profile has been given a name describing the type of person (role) it is created for. This part of the sharing interface was kept mostly unchanged in all the three demos. Even though all the demos allows the user to create their own access profiles, we also kept these as "templates". The purpose was to investigate whether the existence of templates was desired or not, and when give the opportunity to create their own, if the users would actually do so.

The three demos developed are from this point on referred to as: list style, cube style and rainbow style.

Figure 1. Indivo sharing interface

4.1.1 List style

This design is inspired by traditional access control lists, as shown in Figure 2. It simply displays a list of all the information categories with checkboxes to set read and edit permissions. The list is indented to show sub-categories. A row in the list represents one information category. A row is highlighted when permissions are set on a category to make it easier to connect permissions to category. This also serves a purpose when a predefined access profile is selected. Highlighting every row where permissions are set gives an immediate impression of amount how many categories are accessible (in some way the highlighting does not distinguish between edit and read). To make it easier to inspect the list a toggle effect has been added: when the user holds the mouse pointer over a row in the list, that row is highlighted in a darker color.

This is also the design that is most like what is used on many web-sites today. It utilizes controls that are familiar to anyone who are familiar with applications on the web.

4.1.2 Cube style

The cube, shown in Figure 3, was created primarily to visualize amount how much of my information am I sharing. It was inspired by key cards that are solid in places and transparent in other places. Put the key card over an encrypted text and you can read the message in the transparent fields.

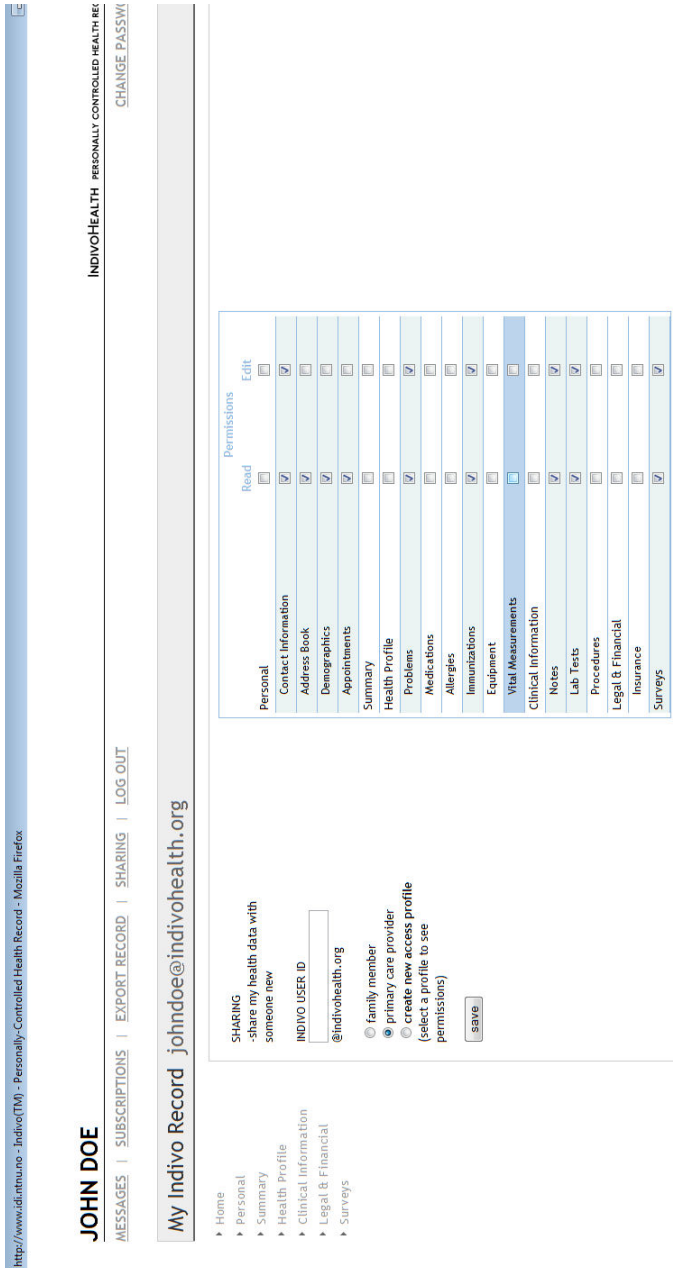

Figure 2. List style visualization of sharing

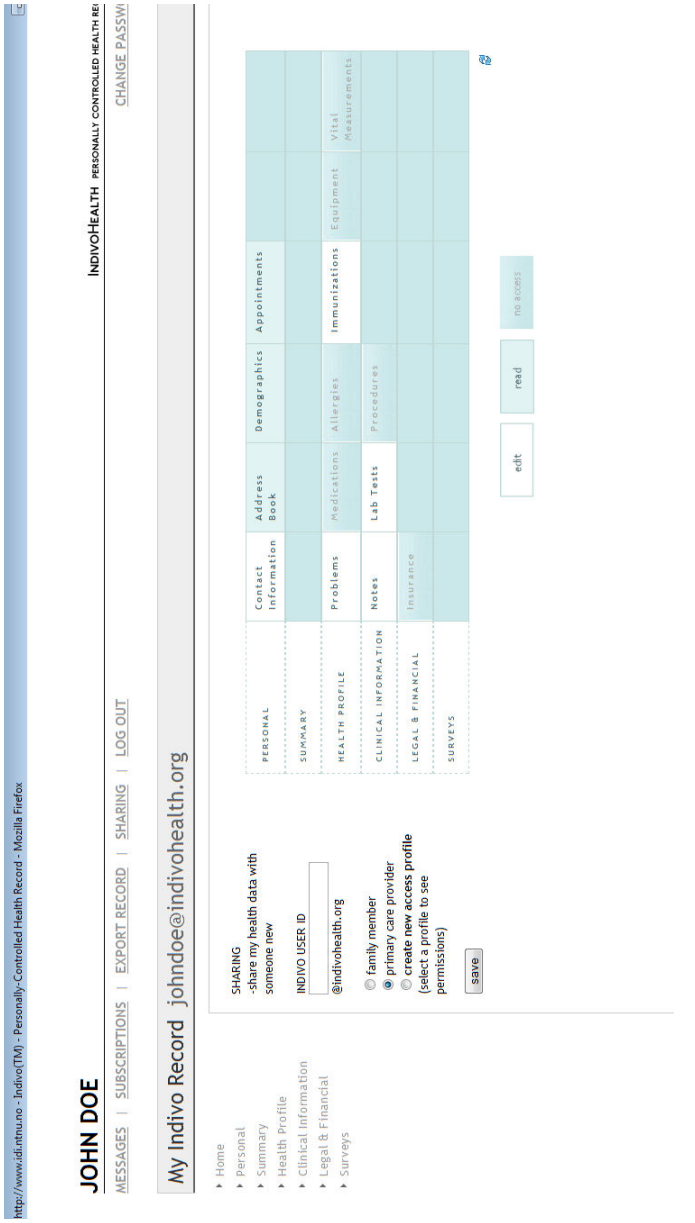

Figure 3. Cube style visualization of sharing

The cube also uses only coloring to visualize permissions.

Early on in the design the intention was to use different colors for the different permissions. However, this could lead to trouble for people who are color blind, so in the end the design was created using different shades of blue to indicate the different permissions.

4.1.3 Rainbow style

This design, shown in Figure 4, was created to take advantage of the menu and to save space. Putting the permission block so close to the menu means there is no need to repeat the names of the information categories. As permissions are changed the coloring of the menu changes too to connect the rainbow to the menu visually. The rainbow takes up so little space that it would be possible to display permissions for several access profiles side-by-side. Also, it could be possible to allow the rainbow to always be present. For example, when a user share with someone she could actually browse through the record, using the menu, and the rainbow would remain in place to show what permissions are set.

4.2 Participants

The participants were recruited from the Indivo MIT[1] study population and from CHIP [2]. As such all participants had some prior knowledge of the system. This was important because we wanted to test for usability, given that the participants were already familiar with the PCHR concept and the idea of sharing health data.

All participants were technically skilled, and as such are not entirely representable for the population in general, which is a limitation of this study. We had a total of 13 study participants ranging in age from first-year college students to retirees.

4.3 Study setup

We started the test by explaining to the participant that what they were going to use was just a mock-up. It was not fully functional and did not contain their health information.

Every participant was then asked to answer two questions before starting the tasks:

- What is your experience with Indivo - how have you used the system before?
- Have you ever used Indivo to share your information with anyone?

The participants were then given three tasks to perform. The tasks were repeated using each of the three demos. We encouraged the participants to "talk aloud" to explain their reasoning and ask any questions they had while using the demos. To minimize the impact of learning, the ordering of the demos when performing tasks were randomized.

The demos contained two predefined profiles: primary care provider and family member. We had intentionally not put much thought into deciding which permissions were included in these profiles. The purpose was to make it more likely for the participants to want to change these profiles if they had a good look at what they included.

The tasks given were:

1. Share with your primary care provider.
2. Share with your physical therapist (assuming you have one).
3. Share with your mother.

About the ordering of the tasks: we elected to give a task for which there existed a usable profile first to coax the participants into selecting this profile from the list which would give them a view of how the interface displayed permissions. The second task was purposefully made to be one where there didn't exist a profile so they had to create a new one. Finally, the third task was made to be one where they could use an existing profile. The purpose was to see if having just completed a task where they had to create a new profile prompted the participants to do so also for the last one or at least adapt/change the existing one and to see if they were more likely to do that now than on the first case when they had yet to experiment with editing profiles.

After they had completed the tasks they were given three cards representing the three different systems and asked to sort them in their preferred order and explain why and what they liked and didn't like about the different ones. This part was conducted as a semi-structured interview, allowing the participants to add any feedback they hadn't mentioned earlier and summarize their thoughts.

At the end of the session each participant was asked two additional questions:

- Would you want there to be template access profiles available?
- Who do you think you would like to share your health data with?

The second question was added to get some feedback on which "roles" are the most common that people think they would like to share with, as these are candidates for templates.

The participants' interactions with the demos were recorded on video, and the entire session was audio recorded. We also kept a log of every access profile created by a user. This allowed us to measure time spent on each task.

[1] Massachusetts Institute of Technology
[2] Children's Hospital Informatics Program

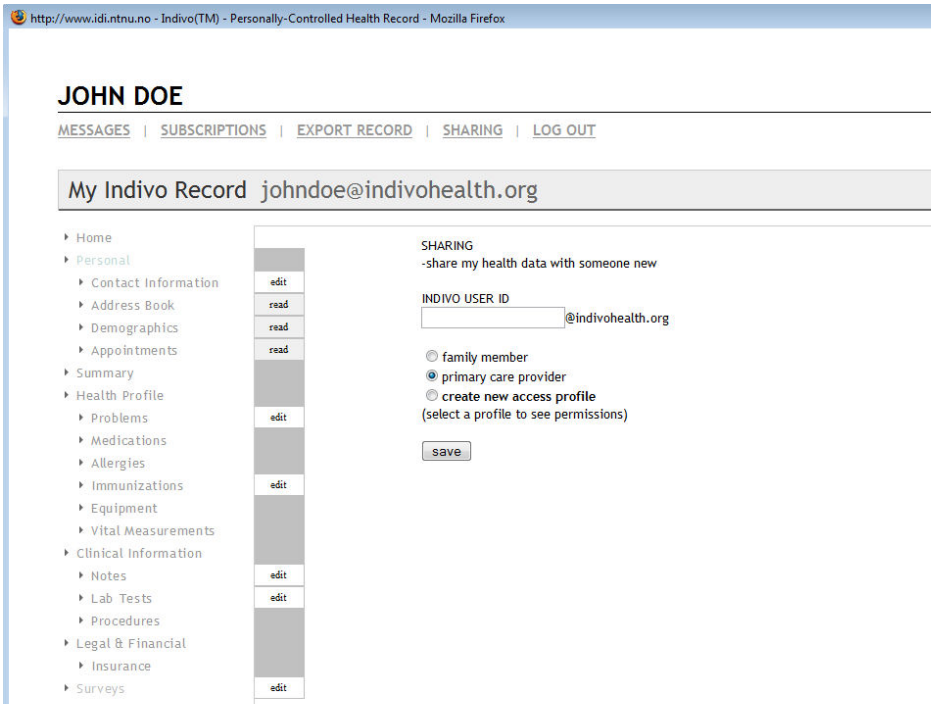

Figure 4. Rainbow style visualization of sharing

5 Results

> "All right little box how do I get you to have words on you?" (study participant)

The time it took to complete one session ranged from 20 to 45 minutes. For most participants it took approximately 30 minutes.

5.1 Familiarity

Most participants reported that they had used the system before. Many had answered surveys on request, and some had also imported their health information[3]. However, none of the participants had ever used the sharing interface before. This reflects that the Indivo study at MIT has been ongoing for a while, but the users have not started to use the system to its full potential yet. However, all participants were aware that they could use the system to share their health information.

5.2 The tasks

5.2.1 Comments on the list style

9 of the 13 study participants selected the list style as their preferred design, and no-one selected it as their least preferred. The most common comment was that this was familiar and intuitive. It did not take long for anyone to figure out how to use it. Positive comments made by participants included:

- "It enables you to see all of the options for read and edit at one time. Invites you to think about read and edit and doesn't require additional instructions."
- "I guess I'm used to seeing things in lists and then multiple possible options for a given parameter. And the ability to both read and edit to select both rather than just click one."
- "(..) every one knows you click a checkbox to select them."
- "(..) very easy to tell what you have selected and what it is you are allowing (the list) whereas the colors (cube) I had to check back if I remembered what the colors meant."
- "I like that it is divided into read and edit - it is very clear."
- "I like simple. Best if you don't have to provide support. List does not require explanation."
- "In terms of very first use - the list. In terms of just looking on it and knowing what to do - but I think that is just because it is the most conventional."

There were also some negative comments given on the list, that included:

- "People can be intimidated when they see a long list of things."
- "(..) but I kind of (points to rainbow) liked how it was just one block that said read/edit rather than having two."
- "Disliked the amount of clicking."

5.2.2 Comments on the cube

3 of the study participants selected the cube as their favorite. However it was also the least favorite of 5 participants. The thing most people criticized about the cube was the choice of coloring. Many complained that the contrast was too low - that it was hard to tell the difference between the blue shades for read and edit.

Some comments made about the cube included:

- "A lot of words, hard to understand organization, takes a while to figure out."
- "Not as used to the horizontal orientation."
- "Because I'm used to reading a table that has labels on the rows and labels on the columns so first I need to figure out that this just has rows labeled."
- "I kind of like the cube. I think it's neat. I don't think the blue works."
- "In terms of understanding I think the grid/cube layout was just a cleaner way of looking at things."
- "A little less confusing than the other two. Think people would be comfortable with this."
- "The repetition of information is an issue. (The menu and the cube basically the same)."
- "Oh read edit..(points to the legend) I didn't... I didn't look at the bottom first. I tried to figure it out and then I realized (points to the legend again)."
- " Oh ok I see the key. So to change it do I select the buttons on the bottom or..?"
- "I didn't notice that it said edit/read. It was also bad colors (not enough contrast). It said it at the bottom so I didn't even notice that it was there until.. subconsciously I thought it was an ok or enter button or something cause usually that's what is at the bottom."

[3] Information is only imported into Indivo when the user takes an explicit action to do so

5.2.3 Comments on the rainbow

Only one study participant selected the rainbow as favorite. It was the least favorite of 6 people. This was also the design it took most participants the longest to get started on. Many started clicking the menu and it took a while to realize that the rainbow was clickable. The one task most people struggled with was task 2 (physical therapist) mainly because when nothing is selected in the rainbow it looks like a solid grey band that clearly does not communicate "click me" to most people.

Some specific comments made on the rainbow:

- "The signing in stuff was on the right and the actual work was on the left was opposite of what I expected."
- " If this was just a checklist would be better." (Suggest checkboxes by the menu instead)
- "(..) require additional instructions about clicking on a box multiple times in order to change status."
- "Should look like buttons so it's apparent that it is clickable. But still it's a double click for each choice whereas this (list) is just a toggle on and off."
- "That's a nice way to avoid repetition."
- "All right little box how do I get you to have words on you? (..) How do I start?"
- "This was the least clear because there were no directions on what to do, just a grey bar, but once you clicked around you realized oh I have to click on that. But it was clear in that it said edit/read (..)"
- "It's not just a color but the color and edit so it's more clear. Don't have to remember which one is edit and which one is read."
- "I didn't even notice them. It doesn't stand out like the others. It makes more sense to me to have it to the right."
- "If it was blank someone may not necessarily click there. Once I selected a template it was clear."
- "Had to figure out that edit includes read."
- "Liked just moving up and down (as opposed to cube where you have to move around."
- "Im more of a visual person so it took me a whole to figure out what to do, because there was less visual ques in my opinion."

5.3 Access profiles

On the topic of templates most people stated that it made sense to have them. However some very interesting comments were made on the authority of templates, including:

> "This kind of thing to my mind has so much authority, that going to it I would wonder well: is that what I'm supposed to tell my physician, is that what they require me to tell my physician or is that what my physician said he needs to know or what."

This is clearly something that needs to be considered carefully. A template is not just for convenience, but may also be perceived as a suggestion made by someone with authority in the matter (in this case the system and/or whoever made it), and people may choose to use the template because they reason that it was created by someone who knows best.

The functionality of the demo was such that the participants could choose to either use one of the templates unchanged, create a profile from scratch or change one of the existing templates. If they chose to create a new or made changes in an existing they were prompted to provide a name when they clicked "save". This new profile would then show up in the list of available templates. Some users had issues with this. Some commented that "create new access profile" should be clearly labeled as functionality and not presented in the list of templates as it is a very different thing.

> "I guess I would expect this function would not be in the same list as other things that are not like it. If it was a separate place to say create a new. It's a different sort of thing (because it's a function)."

The demos were made to automatically switch to "create new.." whenever a user made a change in an existing profile. This was intended to make the system more usable, but turned out to confuse many. Some users selected a profile, made some changes and upon realizing that the profile they had wanted was no longer selected then re-selected it, which meant that they lost all the changes they had made. Clearly this switch should not be done without notifying the user explicitly. Most users who encountered this issue also commented that when they selected a profile and changed it, they expected it to be saved with the same name and different content. In other words they expected to be able to update the profile itself, rather than creating a new one based on the existing profile.

Some users also expressed surprise that the profiles they created showed up in the list of profiles afterwards. However, even if this was unexpected for some they also thought it was a good idea. One user suggested that maybe the list

should be divided into two parts: the profiles suggested by the system and the ones you have created yourself.

On the question of who to share with, most simply answered their provider, any other clinician they have a treatment relationship with, and family. They were probably guided by what templates were available in the demo. Most also stated that as long as they were healthy or didn't have any illnesses they considered embarrassing or otherwise sensitive, they were not that concerned about sharing. But most of them also said that this could change if their health status changed.

5.4 General and unexpected comments

This section summarizes useful comments that were made in the summarizing interview.

5.4.1 Save vs. share

Several users expressed concern or confusion that the button to click after selecting/creating a profile said "save". Some users said they thought they had just created a profile and that the button should say "share" to make it explicit that that is actually what you are doing.

> "So if I didn't wanna change it save means share.
> But if I do wanna change it save means save."

5.5 Information content

Several users were concerned that they didn't know exactly what information was contained in the categories. Some said that they would like to be able to browse their record and look up information and then go back to the sharing interface to continue, but they were afraid that they would loose their changes if they did that.

5.6 On staying informed

Many users were concerned about what would happen after they had shared with someone and how they would know what those people were doing. Comments and suggestions made include:

- "(..) if I'm giving someone permission to edit my information I would want to know that they have changed something, get a notification. Otherwise I'm not comfortable giving them that permission. Even though it makes sense for them to do it. I would want to verify it."
- "The other part that's not clear here. I would like the healthcare provider to fix errors they see or to update this information. If I say they can't edit it....then. I want them to be able to edit but I'm not comfortable giving them edit permission unless the loop is completed (it comes back to me - I'm notified)."
- "Also: if I give someone permission to edit I would want a notification and be able to see what has been changed. (Like track changes.) Maybe colors to see changes made by different people."

5.7 On being allowed to make changes

The main feature of these demos was that they actually allowed the user to change profiles before assigning them to someone. The participants were not told to do so - because an important part of the usability test was to see if the designs themselves communicated to the users that they could change them. Some users took a bit longer than others to realize this. The users who got to do the tasks on the list style interface first got it immediately. It was a bit more difficult for the ones who started out with the cube, and of the ones that started out with the rainbow, several did not realize they could make changes until they got to the next demo. It was very interesting to note that once they realized that they could make changes, every user did that for every task. This indicates that given the opportunity to exercise control, the users would like to do so.

Some specific comments made by user upon realizing that they could make changes:

- "That's nice that it actually... Cause I was kind of making it specific to me so..."
- "When I looked at it I felt I wanted to change it."

6 Discussion

With the launching of both Microsoft Health Vault and Google Health within the last year it is becoming obvious that PCHRs or PCHR-like applications is no longer just an idea. While the potential for patient empowerment is great, so are the security risks for the patient. The patient being able to actually be in control relies on the security, transparency and usability of these applications. Security and usability is often presented as being at complete odds. However, usability as a term was introduced in a security and safety context. To minimize the risk of a plain crashing because of pilot error the controls have to be designed in an intuitive, usable manner. So is the case for access controls in PCHRs. As developers we cannot rely on the technological skills of the users, as the user could be anyone.

The main contribution of the study summarized here is on the users' reactions to being allowed total control over sharing and on the comparison of three proposed visual interfaces for sharing. The main lessons learned include that

given the opportunity, most users would like to exercise detailed control – and that we should consider the authority of system-defined templates.

We cannot conclude that one interface is "perfect". The list style is the one that is most intuitive. How important intuitiveness is depends on how often the functionality is to be used. For functionality that is used often, a slight learning curve can be acceptable because the frequency of use ensures that once the users have learned how to use it, they will remember it. For functionality that is used infrequently, intuitiveness is more important. Sharing your health data with new people is probably not something you do very often and as such intuitiveness is important for a PCHR.

However, as noted several users commented on the level of details. In this demo the users could only set permissions on information categories. The test users expressed wishes to review what information was in one category, and that maybe they would grant a permission on some information but not the entire category. The issue with the list style is that it would "explode" in size if every document in the record should have an entry in the list. If the list could no longer fit on the screen so the users would have to scroll through it, usability would likely be much lower. A possible solution would be to make the list expandable/collapsable (like a menu) to allow the user to examine one category in detail at a time.

The main advantage of the rainbow is that it makes it possible to compare profiles and also that it could allow the user to browse through the record and always see the permissions. The main problem of the rainbow was that it did not appear clickable. It could be interesting to explore merging the two by using checkboxes alongside the menu. This would also eliminate the need to duplicate the information in the menu.

A few simplifications were adapted for the demos. In a PCHR the permission set would be more complex, at least including read, edit and append. In the demos we only used read and edit since testing the users' understanding of the permissions and what they mean were not part of our goals for this study. Including more permissions would make the interfaces more complex and would require a study focusing on the effects of that.

7 Conclusion and Future Work

This paper summarizes findings from a usability study of the sharing interface in a PCHR. The study shows that users would like to exercise detailed control given the opportunity to do so. None of the evaluated interfaces were completely satisfactory and they were also a bit simplified. Further work should focus on enhancing the list style and including functionality that allow the users control on a per-document level.

8 Acknowledgments

Removed for blinding.

References

[1] *Google Health*, last accessed: June 8th 2008. https://www.google.com/health.

[2] *Microsoft HealthVault*, last accessed: June 8th 2008. www.healthvault.com.

[3] Connecting for health: The personal health working group final report. Technical report, Markle Foundation, July 1 2003.

[4] K. D. Mandl, P. Szolovits, and I. S. Kohane. Public standards and patients' control: how to keep electronic medical records accessible but private commentary: Open approaches to electronic patient records commentary: A patient's viewpoint. *BMJ*, 322(7281):283–287, 2001.

[5] W. W. Simons, K. D. Mandl, and I. S. Kohane. The ping personally controlled electronic medical record system: Technical architecture. *J Am Med Inform Assoc*, 12(1):47–54, 2004.

Bibliography

Connecting for health: The personal health working group final report. Technical report, Markle Foundation, July 1 2003.

Unified modeling language: Superstructure. Technical report, Object Management Group (OMG), August 2005. http://www.omg.org.

The asgaard project. URL: http://www.asgaard.tuwien.ac.at. Last accessed: June 18th 2008.

Google health. URL: https://www.google.com/health. Last accessed: June 18th 2008.

Ieee std 1471-2000 ieee recommended practice for architectural description of software-intensive systems. Technical report, IEEE, 2000.

Microsoft healthvault. URL: http://www.healthvault.com. Last accessed: June 18th 2008.

Nou 1997:2 pasienten først (the patient first). Technical report, Ministry of Health and Care Services, 1997.

St. olavs hospital - medical ward. URL: http://www.stolav.no/stolav/Virksomhet/behandling/medisin/index.htm. Last accessed: June 18th 2008.

M. A. Al-Kahtani and R. Sandhu. Rule-based rbac with negative authorization. *Computer Security Applications Conference, 2004. 20th Annual*, pages 405–415, 6-10 Dec. 2004. ISSN 1063-9527. doi: 10.1109/CSAC.2004.32.

I. Alexander. Misuse cases: Use cases with hostile intent. *IEEE Software*, 20(1): 58–66, 2003a.

I. Alexander. Misuse cases help to elicit non-functional requirements. *Computing & Control Engineering Journal*, 14(1):40–45, 2003b.

I. Alexander. Initial industrial experience of misuse cases in trade-off analysis. In *IEEE Joint International Conference on Requiremenst Engineering*, Essen, Germany, 2002a. IEEE.

I. Alexander. Modelling the interplay of conflicting goals with use and misuse cases. In *Goal-Oriented Business-Process Modeling (GBMP) 2002*, volume 109, London, UK, 2002b. CEUR Workshop Proceedings.

I. Alexander. Modelling the interplay of conflicting goals with use and misuse cases. In *International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ) 2002*, Essen, Germany, 2002c.

I. Alexander and N. Maiden. *Scenarios, Stories, Use Cases: Through the Systems Development Life-Cycle*. John Wiley & Sons, 2004. ISBN: 0470861940.

R. J. Anderson. *A security policy model for clinical information systems*. Proceedings of the 1996 IEEE Symposium on Security and Privacy. IEEE Computer Society, 1996.

E. Barka and R. Sandhu. Framework for role-based delegation models. In *Annual Computer Security Applications Conference (ACSAC)*, pages 168–176, 2000.

J. Barkley, K. Beznosov, and J. Uppal. Supporting relationships in access control using role based access control. In *RBAC '99: Proceedings of the fourth ACM workshop on Role-based access control*, pages 55–65, New York, NY, USA, 1999. ACM. ISBN 1-58113-180-1. doi: http://doi.acm.org/10.1145/319171.319177.

K. Beznosov. *Requirements for access control: US Healthcare domain*. Proceedings of the third ACM workshop on Role-based access control. ACM Press, Fairfax, Virginia, United States, 1998. ISBN: 1581131135.

B. Blobel. Authorisation and access control for electronic health record systems. *International Journal of Medical Informatics*, 73(3):251–257, 2004. ISSN: 1386-5056.

N. K. Denzin and Y. S. Lincoln. *Handbook of Qualitative Research*. Sage Publications, Inc, 2 edition, 2000. ISBN: 0761915125.

T. Dingsoyr and N. Moe. The process workshop: a tool to define electronic process guides in small software companies. *Software Engineering Conference, 2004. Proceedings. 2004 Australian*, pages 350–357, 2004. doi: 10.1109/ASWEC.2004.1290488.

W. Essmayr, S. Probst, and E. Weippl. Role-based access controls: Status, dissemination, and prospects for generic security mechanisms. *Electronic Commerce Research*, 4(1-2):127–156, 2004. ISSN 1389-5753.

M. Evered and S. Bögeholz. *A case study in access control requirements for a Health Information System*. Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation - Volume 32. Australian Computer Society, Inc., Dunedin, New Zealand, 2004.

D. Ferraiolo and R. Kuhn. Role-based access control. In *Proceedings of 15th National Computer Security Conference*, 1992.

D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli. Proposed nist standard for role-based access control. *ACM Trans. Inf. Syst. Secur.*, 4(3):224–274, 2001. ISSN 1094-9224. doi: http://doi.acm.org/10.1145/501978.501980.

D. F. Ferraiolo, D. R. Kuhn, and R. Chandramouli. *Role-Based Access Control*. Computer Security Series. Artech House Publishers, Boston, 1 edition, 2003. ISBN: 1580533701.

D. F. Ferraiolo, S. Gavrila, V. Hu, and D. R. Kuhn. Composing and combining policies under the policy machine. In *SACMAT '05: Proceedings of the tenth ACM symposium on Access control models and technologies*, pages 11–20, New York, NY, USA, 2005. ACM. ISBN 1-59593-045-0. doi: http://doi.acm.org/10.1145/1063979.1063982.

M. J. Field and K. N. Lohr. Clinical practice guidelines: Direc-tions for a new program,. Technical Report The National Academy of Sci-ences, 1990.

D. G. Firesmith. Security use cases. *Journal of Object Technology*, 2(3):53–64, 2003.

C. K. Georgiadis, I. Mavridis, G. Pangalos, and R. K. Thomas. Flexible team-based access control using contexts. In *SACMAT '01: Proceedings of the sixth ACM symposium on Access control models and technologies*, pages 21–27, New York, NY, USA, 2001. ACM. ISBN 1-58113-350-2. doi: http://doi.acm.org/10.1145/373256.373259.

D. Gollmann. *Computer Security*. John Wiley & Sons, 1999. ISBN: 0471978442.

W. R. Hersh. Medical informatics: Improving health care through information. *The Journal of the American Medical Association (JAMA)*, 288(16):1955–1958, 2002.

R. Hillestad, J. Bigelow, A. Bower, F. Girosi, R. Meili, R. Scoville, and R. Taylor. Can electronic medical record systems transform health care? potential health benefits, savings, and costs. *Health Affairs*, 24(5):1103–1117, 2005.

P. Hope, G. McGraw, and A. I. Anton. Misuse and abuse cases: Getting past the positive. *IEEE Security & Privacy*, 2(3):90–92, May/June 2004.

V. C. Hu, D. A. Frincke, and D. F. Ferraiolo. The policy machine for security policy management. In *ICCS '01: Proceedings of the International Conference on Computational Science-Part II*, pages 494–506, London, UK, 2001. Springer-Verlag. ISBN 3-540-42233-1.

A. N. S. Institute. American national standard for information technology: Role based access control. Technical Report ANSI INCITS 359-2004, 2004.

S. Jajodia, P. Samarati, V. S. Subrahmanian, and E. Bertino. A unified framework for enforcing multiple access control policies. *SIGMOD Rec.*, 26(2):474–485, 1997. ISSN 0163-5808. doi: http://doi.acm.org/10.1145/253262.253364.

S. Jajodia, P. Samarati, M. L. Sapino, and V. S. Subrahmanian. Flexible support for multiple access control policies. *ACM Trans. Database Syst.*, 26(2):214–260, 2001. ISSN 0362-5915. doi: http://doi.acm.org/10.1145/383891.383894.

M. I. Kim and K. B. Johnson. Personal health records: Evaluation of functionality and utility. *J Am Med Inform Assoc*, 9(2):171–180, 2002.

I. S. Kohane, K. D. Mandl, P. L. Taylor, I. A. Holm, D. J. Nigrin, and L. M. Kunkel. Medicine: Reestablishing the researcher-patient compact. *Science*, 316 (5826):836–837, 2007.

A. Kumar, N. Karnik, and G. Chafle. Context sensitivity in role-based access control. *SIGOPS Oper. Syst. Rev.*, 36(3):53–66, 2002. ISSN 0163-5980. doi: http://doi.acm.org/10.1145/567331.567336.

K. D. Mandl, P. Szolovits, and I. S. Kohane. Public standards and patients' control: how to keep electronic medical records accessible but private commentary: Open approaches to electronic patient records commentary: A patient's viewpoint. *BMJ*, 322(7281):283–287, 2001.

J. McDermott. Abuse case models for security requirements analysis. In *Symposium on Requirements Engineering for Information Security (SREIS)*, Indianapolis, USA, 2001.

J. McDermott and C. Fox. Using abuse case models for security requirements analysis. In *Annual Computer Security Applications Conference*, Phoenix, Arizona, 1999.

G. McGraw. *Software Security - Building Security In.* Addison-Wesley Software Security Series. Addison-Wesley (Pearson Education), Boston, 1 edition, 2006. ISBN: 0321356705.

P. H. Meland, L. Røstad, and I. A. Tøndel. How to mediate between information security and patient safety. In *Proceedings of the Eight International Conference on Probabilistic Safety Assessment and Management (PSAM8)*, New Orleans, 2006.

S. Miksch, Y. Shahar, and P. Johnson. Asbru: A task-specific intention-based, and time-oriented language for repre-senting skeletal plans. In *7th Workshop on Knowledge Engi-neering: Methods and Languages (KEML-97)*, 1997.

S. Na and S. Cheon. *Role delegation in role-based access control*. Proceedings of the fifth ACM workshop on Role-based access control. ACM Press, Berlin, Germany, 2000.

S. Osborn, R. Sandhu, and Q. Munawer. Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Trans. Inf. Syst. Secur.*, 3(2):85–106, 2000. ISSN 1094-9224. doi: http://doi.acm.org/10.1145/354876.354878.

M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R. Greenes, R. Hall, P. Johnson, N. Jones, A. Kumar, S. Miksch, S. Quaglini, A. Seyfang, E. Shortliffe, and M. Stefanelli. Comparing computer-interpretable guideline models: a case-study approach. *JAMIA*, 10(1):52–68, 2003.

D. Povey. *Optimistic security: a new access control paradigm*. Proceedings of the 1999 workshop on New security paradigms. ACM Press, Caledon Hills, Ontario, Canada, 2000. ISBN: 1581131496.

R. Sandhu. Roles versus groups. In *RBAC '95: Proceedings of the first ACM Workshop on Role-based access control*, page 7, New York, NY, USA, 1996. ACM. ISBN 0-89791-759-6. doi: http://doi.acm.org/10.1145/270152.270163.

R. N. Shiffman, Y. Liaw, C. A. Brandt, and G. J. Corb. Computer-based guideline implementation systems: A systematic review of functionality and effectiveness. *JAMIA*, 6:104–114, 1999.

F. Siewe, A. Cau, and H. Zedan. A compositional framework for access control policies enforcement. In *FMSE '03: Proceedings of the 2003 ACM workshop on Formal methods in security engineering*, pages 32–42, New York, NY, USA, 2003. ACM. ISBN 1-58113-781-8. doi: http://doi.acm.org/10.1145/1035429.1035433.

W. W. Simons, K. D. Mandl, and I. S. Kohane. The ping personally controlled electronic medical record system: Technical architecture. *J Am Med Inform Assoc*, 12(1):47–54, 2004.

G. Sindre and A. L. Opdahl. Capturing security requirements through misuse cases. In *Norsk Informatikkonferanse (NIK)*, Tromsø, Norway, 2001a.

G. Sindre and A. L. Opdahl. Eliciting security requirements by misuse cases. In *37th International Conference on Technology of Object-Oriented Languages and Systems (TOOLS-Pacific 2000)*, pages 120–131, Sydney, Australia, 2000.

G. Sindre and A. L. Opdahl. Eliciting security requirments with misuse cases. *Requirements Engineering*, 10(1):34–44, 2005.

G. Sindre and A. L. Opdahl. Templates for misuse case description. In *Seventh International Workshop on Requirements Engineering: Foundation of Software Quality (REFSQ'2001)*, Interlaken, Switzerland, 2001b.

G. Sindre, A. L. Opdahl, and G. F. Brevik. Generalization/specialization as a structuring mechanism for misuse cases. In *2nd Symposium on Requirements Engineering for Information Security (SREIS'02),*, Raleigh, NC, USA, 2002.

G. Sindre, D. G. Firesmith, and A. L. Opdahl. A reuse-based approach to determining security requirements. In *9th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'03)*, Klagenfurt/Velden, Austria, 2003.

I. d. Sørby, T. B. Røst, and Ø. Nytrø. Empirical grounding of guideline implementation in cooperative clinical care situations. In *AI Techniques in Healthcare: Evidence-based Guidelines and Protocols*, Riva del Garda, Italy, 2006.

F. Swiderski. *Threat Modeling*. Microsoft Press U.S., 2004. ISBN: 0735619913.

P. C. Tang, J. S. Ash, D. W. Bates, J. M. Overhage, and D. Z. Sands. Personal health records: Definition, benefits, and strategies for overcoming barriers to adoption. *Journal of the Ammerican Medical Informatics Association*, 2005.

R. K. Thomas. *Team-based access control (TMAC): a primitive for applying role-based access controls in collaborative environments*. Proceedings of the second ACM workshop on Role-based access control. ACM Press, Fairfax, Virginia, United States, 1997a.

R. K. Thomas. Team-based access control (tmac): a primitive for applying role-based access controls in collaborative environments. In *RBAC '97: Proceedings of the second ACM workshop on Role-based access control*, pages 13–19, New York, NY, USA, 1997b. ACM. ISBN 0-89791-985-8. doi: http://doi.acm.org/10.1145/266741.266748.

R. K. Thomas and R. S. Sandhu. Task-based authorization controls (tbac): A family of models for active and enterprise-oriented autorization management. In *Proceedings of the IFIP TC11 WG11.3 Eleventh International Conference on Database Securty XI*, pages 166–181, London, UK, UK, 1998. Chapman & Hall, Ltd. ISBN 0-412-82090-0.

W. Tolone, G.-J. Ahn, T. Pai, and S.-P. Hong. Access control in collaborative systems. *ACM Comput. Surv.*, 37(1):29–41, 2005.

R. Tuchinda. Access control mechanism for intelligent environments. *Bitstream: The MIT Journal of EECS Student Research*, 2002.

D. Verdon and G. McGraw. Risk analysis in software design. *IEEE Security & Privacy*, 2(4):79–84, 2004.

J. Viega and G. McGraw. *Building Secure Software: How to Avoid Security Problems the Right Way*. Addison Wesley, 2001. ISBN: 020172152X.

M. Wilikens, S. Feriti, A. Sanna, and M. Masera. A context-related authorization and access control method based on rbac. In *Symposium on Access Control Models and Technologies*, pages 117–124, Monterey, California, USA, 2002. ACM.

L. Zhang, G.-J. Ahn, and B.-T. Chu. *A role-based delegation framework for healthcare information systems.* Proceedings of the seventh ACM symposium on Access control models and technologies. ACM Press, Monterey, California, USA, 2002.