

**Adaptable Access Control Policies for
Medical Information Systems:
requirements analysis and case studies**

Tine Verhanneman

Liesbeth Jaco

Bart De Win

Frank Piessens

Wouter Joosen

Report CW 363, August 2003



Katholieke Universiteit Leuven
Department of Computer Science

Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

Adaptable Access Control Policies for Medical Information Systems: requirements analysis and case studies

Tine Verhanneman

Liesbeth Jaco

Bart De Win

Frank Piessens

Wouter Joosen

Report CW363, August 2003

Department of Computer Science, K.U.Leuven

Abstract

IT enforced access control policies in medical information systems have to be fine-grained and dynamic. We justify this observation on the basis of legislation and on the basis of the evolution within the healthcare domain. Consequently, a reconfigurable or at least adaptable implementation of access control facilities has become extremely important. For this purpose, current technology provides insufficient support. We highlight a basic solution to address shortcomings by using interception techniques. In addition, we identify further research that is required to address the challenges of dynamic and fine-grained access control in the long run.

Keywords : access control, medical information systems

CR Subject Classification : K6.5, J3, H2.0

Adaptable Access Control Policies for Medical Information Systems: requirements analysis and case studies

Tine Verhanneman, Liesbeth Jaco, Bart De Win,
Frank Piessens and Wouter Joosen

Abstract

IT enforced access control policies in medical information systems have to be fine-grained and dynamic. We justify this observation on the basis of legislation and on the basis of the evolution within the healthcare domain. Consequently, a reconfigurable or at least adaptable implementation of access control facilities has become extremely important. For this purpose, current technology provides insufficient support. We highlight a basic solution to address shortcomings by using interception techniques. In addition, we identify further research that is required to address the challenges of dynamic and fine-grained access control in the long run.

1 Introduction

The healthcare industry spends increasingly more resources on information technology. This evolution is driven by both the need to manage healthcare costs and by the changing structure of healthcare organizations. A side effect is a growing concern regarding the privacy and security of health care information([1, 2]).

In this paper, we characterize the requirements of access control in medical information systems. Because of these sophisticated demands, this is an interesting case study for complicated access control ([9, 17]). First, we explain why access control policies are not (and will never be) static. Secondly, we discuss the importance of fine-grained IT enforced policies.

Access control policies are dynamic and two driving factors play a crucial role in such an evolution: legislation on the one hand and system changes on the other hand.

There are certain rules (ethical rules, enforced by legislation) that describe the way medical data should be handled. First of all, these laws can change, so access control policies are by nature not static. Second, the changing interpretation of the legislation must be considered. As will be shown in section 2, the applicable law in the USA states that to meet some of the requirements a “reasonable effort” should be done. The concrete interpretation of this term depends on the context, like e.g. the size of the organization and on case law. A court decision for one specific case can lead to the fact that many healthcare organizations need to tighten their access control.

Another cause of evolving access control policies are system changes. One kind of system change is the extension of functionality, for instance when not only hospital personnel but also patients and relatives can login onto the system to access their own clinical information. Another system change is the effective IT enforcement of access control. This topic will drive the examples that we discuss in this paper. IT enforced access control policies must be *fine grained*. We will give examples of access control rules in healthcare, and we show that they may rely on context or application state, as well as on subject, object and operation to be performed.

In effect, applications that implement medical information systems have to include reconfigurable and/or adaptable access control technology. This is an essential part of the security components of such an application.

We now define these terms more precisely. Security policies are said to be reconfigurable if the deployer of the application is able to set the security measures without having to dig into the source-code written by the programmer. Access control is set through the interpretation of a configuration file, written in a “non-programming language”, such as XML for example. Ideally, the format and language of this configuration file are not too complicated, so that non-programmers are able to draw it up. In the literature, this is often referred to as *declarative security*. Declarative security is often opposed to programmatic security, in which access control policies are hard-coded in the application.

Adaptability is a weaker requirement than reconfigurability. A system that is highly adaptable requires little effort to be transformed into a system that satisfies new requirements. Adaptability implies that all the concerns are expressed somehow separately in a modular structure. This not necessarily means that the access control policy can be changed at deployment time or at run time.

Supporting reconfigurability or adaptability often limits the access control policies that can be implemented, as expressive power is limited, either because the configuration language is limited, or because

the module that implements access control does not have access to all information that is required to enable detailed decision making. Expressiveness must be treated with caution, since the more expressive the language the more complex the enforcement mechanism will be.

If access control policies are expected to be dynamically evolving and fine-grained by nature, application servers preferably should support reconfigurable access control in a high level policy language that is sufficiently expressive.

The rest of this paper consists of two parts. In a first part, we analyze the requirements of medical information systems with regards to access control. This analysis is based on the legal and regulatory framework for privacy and security of medical data in both the European Union and the United States, and also on the evolution of the organization of healthcare. This will be discussed in section 2. We illustrate the access control requirements with a more specific example in section 3. In a second part of the paper, we discuss the impact of our observations when building application using state-of-the-art platforms such as J2EE and .NET. Obviously, the current support for declarative access control is insufficient, and we outline improvements in section 4. Section 5 identifies future research directions and we conclude in section 6.

2 Access control policies in healthcare

All existing threats against privacy and security of health information, violate certain generally accepted rules, that express a concern for patient rights and the well-functioning of health care. The ethical basis of those rules is the Hippocrates oath, describing the right of confidentiality. Nowadays, the rules for the use of medical information are extended with the right to dispose of data with guaranteed integrity and availability. Laws are constituted, describing the rights and duties imposed when processing medical data and defining the potential sanctions applied to misuse. On this basis, healthcare institutions formulate policies, containing both organizational and technical security measures.

The legislation actually provides for two kinds of rights and duties.

First, the law prescribes the circumstances for medical data to be collected, stored and used, and the authorization rules to access the data. This is input for the access control policy that a healthcare organization should manage. The HIPAA Privacy Rule is the prime example of such a legislation ([20]).

Second, the law also sets some standards on how well the policy should be *enforced*. In other words, if a healthcare organization stores

and processes medical data, and outsiders (or malicious insiders) manage to get unauthorized access to the data, the organization could still be prosecuted and convicted if it could be shown that the data was not appropriately protected against unauthorized access. The HIPAA Security Rule is the prime example of this kind of legislation ([22]).

A detailed discussion of the first kind of legislation would lead us too far but some examples of access control rules will be given in section 3.3. But the second kind of legislation is important from the point of view of IT enforcement. Hence we give a short survey of the relevant legislation.

2.1 Legislation in the European Union

Considering the protection of health information in the EU, the Data Protection Law (Directive 95/46/EC) should be mentioned first ([11]). It does not only apply to personal identifiable data in general, but also to personal identifiable medical data, and to both automatic and manual processing. Article 17 requires:

Member States shall provide that the controller must implement *appropriate* technical and organizational measures to protect personal data against accidental or unlawful destruction or accidental loss, alteration, unauthorized disclosure or access, in particular where the processing involves the transmission of data over a network, and against all other unlawful forms of processing. Having regard to the state of the art and the cost of their implementation, such measures shall ensure a level of security *appropriate* to the risks represented by the processing and the nature of the data to be protected.

Recommendation R(97)5 ("on the Protection of Medical Data") ([12]) provides some further guidance for healthcare providers. Recommendations have no legally binding character for the member states, but are incentives for certain behavior. The text of the recommendation contains the following part:

9.1 Appropriate technical and organizational measures shall be taken to protect personal data - processed in accordance with this recommendation - against accidental or illegal destruction, accidental loss, as well as against unauthorise access, alteration, communication or any other form of processing. Such measures shall ensure an appropriate level of security taking account, on the one hand, of the technical state of the art and, on the other hand, of the

sensitive nature of medical data and the evaluation of potential risks. These measures shall be *reviewed* periodically.

We argue that the emphasis on *appropriate* measures and periodical review necessitates flexibility and reconfigurability of the IT enforced access control policy.

2.2 Legislation in the US

Contrary to the European Union, in the US there is no explicit constitutional recognition of privacy. Therefore, there is no comprehensive legislation, but rather a patchwork of laws, directed to a certain domain. Drawback of this approach is the large amount of laws enacted, advantage is that specific issues of a certain domain are dealt with in its very own way.

The specific law concerning the protection of individually identifiable health information is the Health Insurance Portability and Accountability Act of 1996, also known as HIPAA.

HIPAA is considered the most significant healthcare legislation passed in years. The law contains several sections, including rules on electronic transactions, national identifiers, patient privacy and data security. It obliges healthcare organizations to use information and communication technology to increase efficiency, but it also addresses the problems of deploying these technologies. Therefore all healthcare organizations that maintain or transmit electronic health information must comply, and there are severe civil and criminal penalties for those that do not.

In the context of this paper, two rules of the comprehensive HIPAA regulation are important, namely the Privacy Rule ([20]) and Security Rule ([22]). The Security Rule applies to protected health information in electronic form only, whereas the Privacy Rule applies to protected health information in any form. The latter sets forth what uses and disclosures are authorized or required and what rights patients have with respect to their health information.

The relationship between the Privacy Rule and the Security Rule can be summarized by saying that the former sets the policy to which personal health information should be subjected, while the latter specifies what implementation is obligatory for enforcement of this policy or what reasonable efforts should be done. It describes the necessity for standards at all stages of transmission and storage of electronic health care information to ensure integrity and confidentiality of the records at all phases of the process, before, during and after electronic transmission. It defines administrative, physical and technical

safeguards to protect the confidentiality, integrity and availability of electronic protected health information.

Regarding access control, a rewording in the Final Security Rule ([22]) in comparison with the Proposed Rule ([21]) can be noticed:

There was no intent to limit the implementation features to the named technologies and this final rule has been reworded to make it clear that use of any appropriate access control mechanism is allowed. Proposed implementation features titled “Context-based access,” “Role-based access”, and “User-based access” have been deleted and the access control standard at Sec. 164.312(a)(1) states the general requirement.

Features such as context-based, role-based and user-based access control are no longer explicitly mentioned in the rule. Instead, the requirement is that *appropriate* access control should be provided. Like in the European legislation, the emphasis is on the fact that technical enforcement should be appropriate with respect to risk. In the following section, we illustrate how this requirement leads to dynamic access control policies that must be enforced by IT.

3 IT enforced policies

3.1 Towards more IT enforcement

A security policy can be enforced in many ways: through organizational measures (e.g. assigning responsibility for security of data in a clear way), through technical measures (e.g. using firewalls and access control in applications). In some cases, no explicit enforcement is needed and employees are trusted to conform to the security policy.

In healthcare, there is an old tradition of trusting the care provider. Because of the increased specialization of care providers, and the increased complexity of care procedures, the size of the team of care providers that deals with one patient grows. Teams of ten to fifty are common. Obviously the purely trust-based model does not work.

Besides, the increased use of IT makes technical measures to enforce the security policy unavoidable. Many hospitals have already reached the maturity-point where hospital wards are distributed in different, separated buildings. Because each ward has its own administration, data is no longer centralized and communication networks outside the physical boundaries are used to share information. More evolved healthcare organizations offer remote access, which allows doctors or patients to access clinical information from off-site locations.

Many people have increasing (potential) access to personal clinical information of a large number of patients. Therefore IT enforcement becomes essential, organizations rely less on trust ([3]). In this context the term *regular policy* refers to the policy a healthcare organization wishes to impose, independent of the enforcement mechanism. *IT enforced policy* refers to a policy that is actually enforced by IT-based technical measures.

3.2 IT enforced policies are dynamic

In this paper, the focus is on the IT enforced access control policy. It is unreasonable to suppose that these IT enforced access control policies will remain the same over long periods of time, for the following reasons.

1. *Changes in the regular policy* can be caused by changes in legislation, changes in the interpretation of legislation, or because the hospital management decides that a stricter policy will lead to a competitive advantage as patient awareness about privacy increases.
2. *Appropriate security is necessarily dynamic.* As discussed in section 2, the law states that enforcement of the access control policy should be reviewed regularly.
3. There is a shift of the trust based model towards *more IT enforcement.*
4. *IT enforced policy evolves with system changes.* As system functionality increases, the security rules need to be enforced in more and more situations.

3.3 An example of fine grained and context sensitive access control

The concept of an IT enforced security policy is introduced above. In practice, this requires the availability of a structured, computer-interpretable medical record. Record definitions must be standardized according to a reference model and match the semantics of the target domain. HIPAA regulates the improvement of standards and their implementation. One of the designated standardization organizations within this context is Health Level Seven (HL7) [15]. Their Reference Information Model (RIM) is a static model of health and health care information. We will use this model to illustrate our findings.

It should be noticed that the RIM model is a relatively detailed domain model and the result of an in depth analysis. The illustrations

that we show in this section are based on RIM, and hence inherit some peculiarities that cannot be discussed within the scope of this paper. Figure 1 gives an overview of the model. One can identify the six core classes: Entity, Role, RoleLink, Act, ActRelationship and Participation. Each class has certain (mandatory) attributes to specify more details about the meaning and purpose of it. Since these classes are quite high-level, subclasses are defined when additional characteristics are necessary.

Figure 1: HL7 Reference Information Model

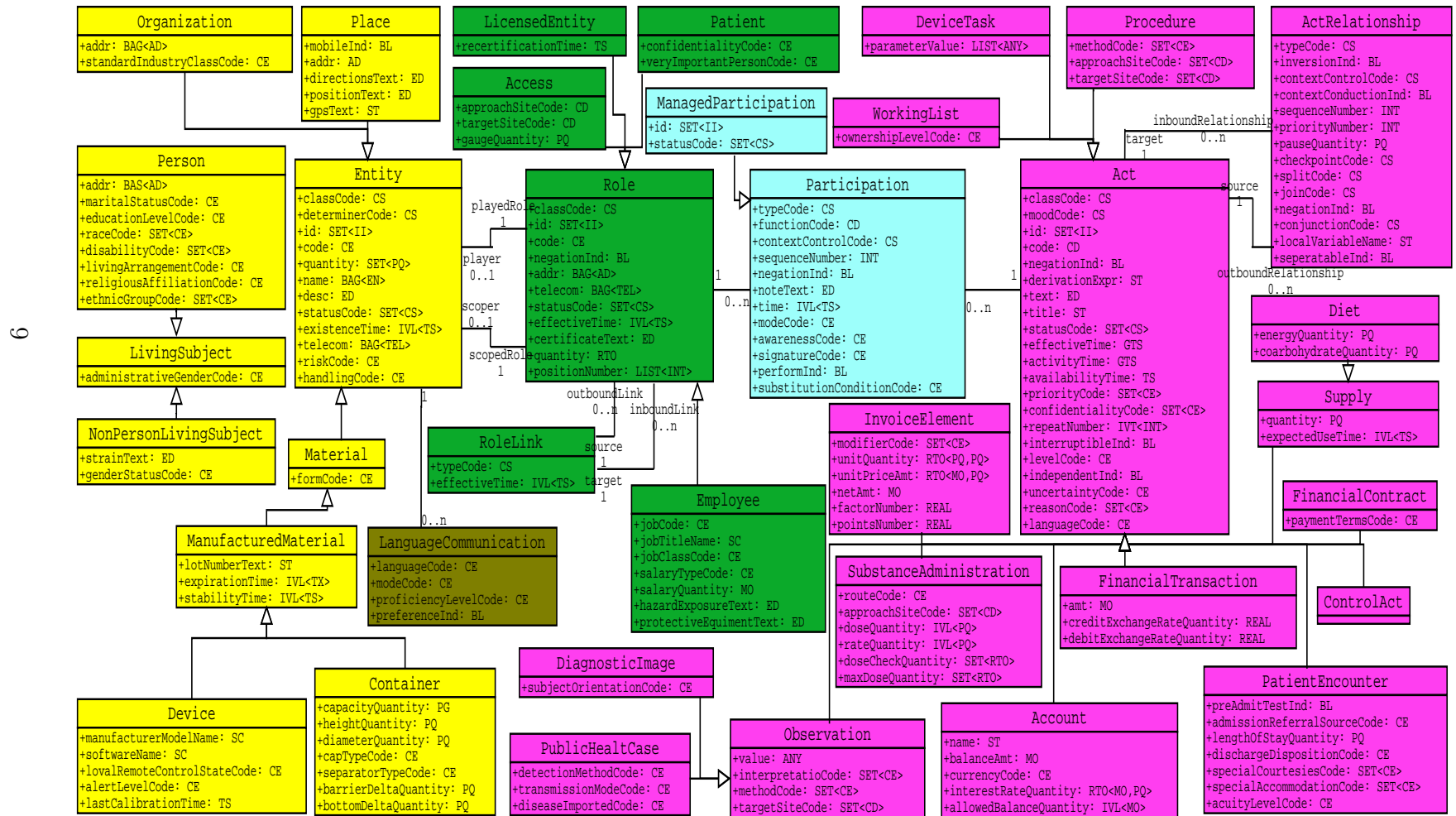
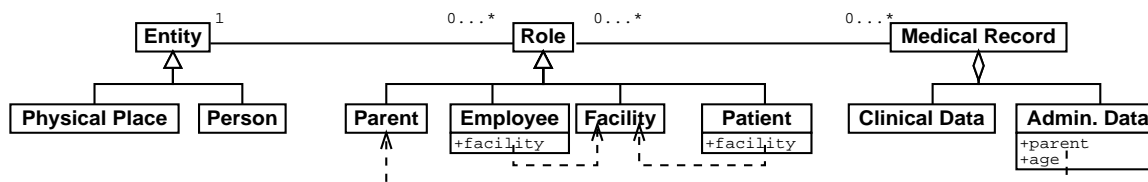


Figure 2: A simplified RIM diagram



To illustrate the use of this RIM model in the context of access control, the class diagram in figure 2 is used. This is a simplified (yet consistent) model of how the concept of a medical record and its concerns can be expressed. The RIM model can be recognized as it contains Entities, Roles and Acts. For the pedagogical purpose of this illustrations, we have omitted an explicit design of the associations that are represented in the classes RoleLink, Participation and ActRelationship.

Two examples of access control rules are given below. We show a collaboration diagram for both cases.

The first example of an access control rule is: “A parent may lookup the clinical data of its child if the child’s age is less than 18”. Figure 3 shows that access is granted on the basis of information contained in Administrative Data, because this class knows the patient’s parent, and whether or not the child is a minor. So access control is performed when an inspect-operation is carried out on the Clinical Data-class.

The second example is “An administrative servant may perform administrative actions only for patients located in the user’s facility”(figure 4). Here the access control is performed when the administrative servant performs an operation on the Administrative Data. The input for the access control decision is retrieved from the Patient-class, which contains the facility where the patient is hospitalized.

The state-of-the-art model for IT-enforced access control is the Role Based Access Control (RBAC) model ([13]). A role in this context is a job or responsibility related grouping of permissions, e.g. doctor-role and nurse-role. Users are then assigned to roles, and as such acquire the privileges associated with the role. But as illustrated above, healthcare applications will need many types of information in addition to the role of the user to make authorization decisions in order to comply with patient information disclosure requirements ([25]).

- instance based information, as illustrated in the first case above.

Figure 3: interaction diagram illustrating the lookup by a parent

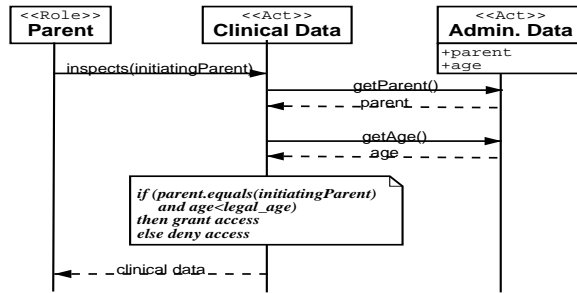
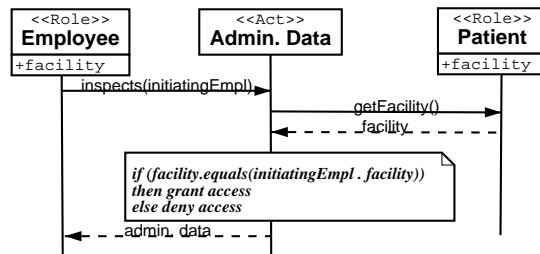


Figure 4: interaction diagram illustrating the lookup by an administrative servant



- state information. This allows to let access control decisions depend on certain values of fields in the request. This has been illustrated in the second case described above.
- contextual information, which allows to distinguish between the set of roles assigned to a user at *registration time*, and the set of roles that can be activated at *session time*. The latter depends on the working context of a particular user, combined with dynamic exclusion rules. This working context can be a (physical) place or time. The following rule is an example: “*on-call physician can only be activated during night time*”. In addition, context information allows to dynamically adapt permission assignment for a particular role dependent on specific user context constraints, for example: “*no write access to therapy data if a ward physician is outside his default ward*”.

Clearly, the language to formulate the policy needs to be expressive to describe sufficiently fine-grained and context-aware access control rules.

4 Support for flexible access control policies

4.1 The need for expressiveness

Access control rules describe conditions for subjects to access resources. It is obviously not feasible to write a separate rule for any object or subject in the system. Hence, a good policy language should support the grouping of objects with similar access control requirements and the grouping of subjects with similar rights.

Grouping of objects is typically achieved through *policy domains*. Each of the objects in the system belongs to one or maybe several domains. This does not always depend on the class of the object alone, but also on instance properties, the relations the object is involved in, the location where the object instance is deployed, ... A powerful policy language should support fine-grained description of policy domains as well as short, generalizing definitions.

Besides the grouping of objects, some support of grouping *subjects* will be required. An obvious example of this kind of grouping is the notion of a role, as discussed in 3.3. A more advanced example is implicit grouping as described in [14], where roles are assigned (dynamically) on the basis of properties of the subject.

A subject may only have limited access to an object. In object-oriented systems, there is a tendency to consider a method as the Unit

of *privilege*. However, in the majority of the applications in Medical IS, it seems that privileges are more naturally expressed in terms of access to and modification of data. This is hard to realize, however, since it cannot always be determined beforehand whether or not a method will access and/or modify a given data-unit.

Last, since the outcome of an access control decision is often influenced by *context information* (section 3.3), such as time, location, etc., a policy language should have access to this kind of context information.

Given all these concerns, it becomes clear that it is in fact the *interaction* in its whole, that must be taken into consideration when describing and executing the access control decision process. An *interaction* is the chain of method invocations that are caused by an invocation of the client on the application server. An interaction is always carried out on behalf of a subject on a set of target-objects in a specific context. This subject can assume one or more roles, which can be activated when and if needed. An interaction can also be tagged with some additional contextual information. Interactions should be identified and grouped into policy domains, in a fine-grained and also flexible manner. Domains then allow for the association of conditions with each individual interaction. This is the ultimate requirement for a policy language.

4.2 An assessment of J2EE and .NET

Modern application platforms such as J2EE and .NET provide support for activating infrastructural services such as access control in a declarative way. As a consequence, access control policies are reconfigurable at deployment time or even at run time. But the expressiveness of such declarative access control policies is rather limited. For more complex, fine grained policies, access control is enforced in the code, leading to policies that are harder to adapt.

J2EE offers a minimal support for RBAC as defined in section 3.3. Security roles can be defined (in principle by the developer) by grouping users in categories. The mapping of these security roles to security identity is done at deployment time by the application deployer. The permissions assigned to the security roles are based on method-invocations. Each method specification needs to describe the roles that have permission to invoke the method. A J2EE container allows by default all users ("AllUsers") to invoke all methods, so access permissions definitely need to be restricted.

..NET features are comparable with J2EE. A developer can tag methods with the roles that are required to execute the method by us-

ing the `PrincipalPermissionAttributes`. .NET also provides a bridge to the access control support that was supported in COM+. The COM+ access control technology allows the deployer to attach required roles to methods that are exposed by the application.

If one uses these declarative security systems, access control is reconfigurable, but fails to offer the expressive power as stated in section 4.1:

- The policy domain is determined by the type (class) of the target object only. This means that it is impossible to enforce access rules that are specific for instances.
- Grouping of subjects is only possible on the basis of roles that have to be assigned statically. Neither role-activation nor the enforcement of dynamic separation of duty is supported. In general, it is the application container that keeps track of the subject on whose behalf the invocation is made. There is no possibility to attach custom-defined data to the invocation.
- Access control rules are set up in terms of method-invocations. If the deployer wants to enforce a data modification access rule, the deployer needs to find out which methods may possibly modify the data.
- Further context information, such as external factors (context, time, ...) can not be used in the access control decision.

The only way to deal with the above mentioned limitations is by hard coding security into the application. But by using such *programmatic security*, reconfigurability and even adaptability are lost.

4.3 A solution based on extensible interceptor chains

A first step toward *interaction*-based access control consists of supporting *extensible interceptor chains*. An interceptor is a construct that intercepts method-invocations and that carries out some extra functionality before passing on, blocking, or redirecting the invocation. Depending on the considered technology, interceptor-instances can be deployed per container, class, object instance and/or context. It is fair to say that both J2EE and .NET provide some preliminary support.

For example, the open source J2EE-compliant JBoss application server tries to disentangle security code and application-logic by the definition of so-called security proxies. These proxies enforce the security policy outside the bean, but they have to be programmed in programmatic way.

.NET apparently has configurable interceptor chains built in already, but this is an undocumented feature rather than a solution. So-called context bound objects in .NET can be decorated with attributes that specify what interceptor should handle all method calls on the object. This interceptor can be programmed in any of the .NET languages.

Our team has been experimenting with application architectures that support flexible interceptor chains since about five years, and we have been applying this approach in security frameworks for electronic commerce[10]. Clearly, if one uses an interceptor to implement access control, adaptability can be achieved: the access control logic is encapsulated in the interceptor, and relatively easily replaceable (at least at development time) by another interceptor. This solution has its limitations: reconfigurability is not possible, and access control is certainly not specified at a high level of abstraction. Rules need to be programmed in a general purpose programming language with a reified method call as primary input. Expressiveness however is significantly better, since the interceptor can base its decision on object state, application state and available context information.

5 The way forward

Towards reconfigurable, expressive and high level access control.

Extensible interceptor chains as discussed above are only a first step towards achieving the goal of reconfigurable, expressive and high level access control. Some issues still need to be solved.

- Programming interceptors is too low-level: a more developer-friendly programming model is necessary. Most probably, the interceptor should be interpreting a high-level policy language. Therefore, research about policy languages ([8]) is relevant. Then the actual policy can be configured at run time.
- Current application containers do not pass much context information to the objects inside. For instance, context information about the current interaction, such as where the interaction was started (on a remote network or a local network) might influence an access control decision, but this information is not available to an interceptor.

Both issues are (at least in part) addressed by the Aspect Oriented Software Development (AOSD) community[4]. AOSD has as objective to modularize concerns, such as security, into slices of behavior. This modularization not only consists of the implementation of the required functionality but also the composition of these slices

into the overall application and the management of the relations between slices. A variety of techniques are being investigated, varying from the weaving of extra code at compile time (AspectJ [5]) to the (runtime) deployment of wrappers (Lasagne, JAC [18, 24]). The main difference between these AOSD systems and the previously described extensible interceptor chains is that there is considerably more support for the creation and composition of these slices of behavior. Often new language primitives are introduced to support the definition of these aspects. This addresses the first issue raised above.

In order to support the consistent activation of aspects, and to support client-specific views, Truyen et al. [24] have suggested to attach metadata to interactions to drive the activation. Such techniques can address the second issue raised above: contextual information about the interaction has to travel with the interaction as metadata.

6 Conclusion

This paper has argued, on the basis of legislation and on the basis of the evolution of healthcare, that IT enforced access control policies in medical information systems will be fine-grained and dynamic. As a consequence a reconfigurable or at least adaptable implementation of access control is very important. Current application servers were shown to provide insufficient support for this, and some indications of how these shortcomings could be remedied were given.

References

- [1] R. J. Anderson, *A Security Policy Model for Clinical Information Systems*. IEEE Symposium on Security and Privacy, Oakland, CA, May 1996, pp 30-43
- [2] R. J. Anderson, *Security in Clinical Information Systems*. British Medical Association, Tech. Report, London, January 4, 1996.
- [3] R. J. Anderson, *Patient Confidentiality – At Risk from NHS Wide Networking*. Proceedings of Health Care 96, March 96
- [4] <http://www.aosd.net>
- [5] <http://aspectj.org>
- [6] S. Bodoff, D. Green, K. Haase, E. Jendrock, M. Pawlan, and Beth Stearns, *The J2EE Tutorial* Addison-Wesley, March 2002

- [7] Computer Science and Telecommunications Board, National Research Council *For the record, protecting electronic health information*, National Academy Press, July 1997
- [8] N. Damianou, *A Policy Framework for Management of Distributed Systems*, PhD thesis, Februari 2002
- [9] I. Denley, S. Weston Smith, *Privacy in Clinical Information Systems in Secondary Care*, British Medical Journal, 318:1328–30, May 1999
- [10] B. De Win, J. Van den Bergh, F. Matthijs, B. De Decker, and W. Joosen, *A security architecture for electronic commerce applications*, Information Security for Global Information Infrastructures (S. Qing and J. Eloff, eds.), Kluwer Academic Publishers, 2000, pp. 491-500
- [11] European Parliament and Council of Europe *Directive 95/46/EC, on the protection of individuals with regard to the processing of personal data and on the free movement of such data*, October 24, 1995
- [12] Council of Europe, *Recommendation R(97)5, On the protection of medical data*, February 12, 1997
- [13] D.F. Ferraiolo, R. Sandhu, S. Gavrila, D. Kuhn, R. Chandramouli, *Proposed NIST Standard for Role-Based Access Control*. ACM Transactions on Information and System Security, Vol.4, No. 3, August 2001, pages 224-274
- [14] R. Goodwin, S.F. Foh, F.Y. Wu, *Instance-level access control for business-to-business electronic commerce*, IBM Systems Journal, Volume 41, number 2, Januari 2002
- [15] Health Level Seven, <http://www.hl7.org>
- [16] K. Lieberherr, D. Lorenz, M. Mezini, *Programming with Aspectual Components*
- [17] W. W. Lowrance, *Privacy and Health Research, a Report to the U.S.*, Secretary of Health and Human Services. May 1997
- [18] R. Pawlak, L. Duchien, G. Florin, L. Seinturier *JAC : a Flexible Solution for Aspect Oriented Programming in Java*, Reflection 2001, Kyoto, Japan, Septembre 2001

- [19] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, C. E. Youman, *Role-Based Access Control Models*.IEEE Computer, vol 29, Num 2, February 1996, p38-47
- [20] Secretary of the Department of Health and Human Services, *Final Privacy Rule*, Federal Register, August 14, 2002
- [21] Secretary of the Department of Health and Human Services, *Proposed Security Rule*, Federal Register, December 28, 2000
- [22] Secretary of the Department of Health and Human Services, *Final Security Rule*, Federal Register, February 20, 2003
- [23] L. Taylor *Customized EJB security in JBoss. Separate your security policy from your business logic*, Java-World, February 15, 2002
- [24] E. Truyen, B. Vanhaute, W. Joosen, P. Verbaeten, B. N. Joergensen, *A Dynamic Customization Model for Distributed Component-Based Applications*, accepted for International Workshop on Dynamic and Distributed Multiservice Architectures (DDMA 2001)
- [25] M. Wilikens, S. Feriti, A. Sanna, M. Masera, *A context-related authorization and access control method based on RBAC: a case study from the health care domain*. SACMAT 2002: 117-124