SPECIFYING SECURITY POLICIES FOR ELECTRONIC HEALTH RECORDS

Felix Apitzsch
Fraunhofer Institute for Open Communication Systems FOKUS, Kaiserin-Augusta-Allee 31, Berlin, Germany

Stefan Liske, Thomas Scheffler, Bettina Schnor
Department of Computer Science, Potsdam University, Potsdam, Germany

Keywords: Electronic Health Record (EHR), Security Policies, Digital Rights Language, Privilege Management and Access Control (PMAC).

Abstract: Sensitive data in electronic health records needs marking for special handling in order to maintain privacy. Person-centred records need mechanisms for individual and flexible marking. Policy mechanisms currently applied with shared health records in integrated care environments lack the ability to model complex privacy requirements. The paper examines two state-of-the-art policy languages for distributed processing environments such as web-services and digital rights management and describes how they can be applied with XML health records. Furthermore, it highlights the abstract concepts that need to be adopted and presents a distributed policy enforcement model.

1 INTRODUCTION

One of the major problems with integrated care depending on distributed care processes and distributed health IT is data interoperability between different IT systems. Centralised health care limited to one hospital and a single episode of illness can be handled by a single IT system or at least systems operated by a single authority. With integrated care, health data needs to be securely available at different locations in the care process.

The following paper describes how existing standards for the expression of data access policies for XML data can be applied to health IT-systems to protect Electronic Health Records (EHRs). We examine different EHR standards with respect to their support for access policies. Afterwards we analyse two policy languages that can be used to describe access control for distributed private medical data. [1]

Traditionally, security policies are first and foremost bound to a system, not to the resources. Even though there exist concepts to bind a security policy primarily to a resource (Wang, 2005), this approach has not been taken with EHRs yet. Existing access control methods are tailored towards a centralised data storage model. Consequently, medical records are stored in a medical database that is being accessed from authenticated users. Access control to the database items would be mediated by the database itself according to specific access control policies.

An important new aspect, that is specific to the domain of distributed health IT and the use of access control schemes, is the fact that there is no central policy management in place. Instead, data owners themselves should be able to set their own privacy preferences with the support of default settings and exemplary templates. These preferences then need to remain with the data as *sticky policies* (Mont et al., 2003)(Karjoth et al., 2003) throughout the ongoing data distribution process.

In a truly distributed system, medical data can migrate freely (e.g. on a patients smartcard (BMG, 2006)) and no centralised mechanism exists to protect access to the data. Therefore, the data has to be self-contained, which implies that it incorporates the necessary policies and protection mechanisms. A com-

[1] The security model proposed in this paper tries not to ignore or reinvent security or confidentiality concepts (Blobel et al., 2006) already described in ISO 22600, ISO 27799, ISO 21000, ISO 21731, EN 13606 and EN 13608, but to combine their views and to present some important common or extending concepts on an abstracting level.

Apitzsch F., Liske S., Scheffler T. and Schnor B. (2008).
SPECIFYING SECURITY POLICIES FOR ELECTRONIC HEALTH RECORDS.
In *Proceedings of the First International Conference on Health Informatics*, pages 82-90

mon security architecture has to be applied by any system involved in the care process and the relevant components have to comply with well-defined standards. We propose to mediate distributed data access with the help of explicit policy description languages.

The necessary authentication mechanisms as well as the mechanisms for the protection of the EHR (e.g. through encryption) are outside the scope of this paper, but the latter have been addressed in previous work (Apitzsch, 2007).

2 INTEGRATED CARE USE CASE

In this section we describe an integrated care use case that serves as the basis for the policy examples described in section 4. We present informal privacy requirements for the use case, as well as possible EHR XML-structures representing its data.

2.1 Shared Use of Medical Data

The following example of a fictional medical history is constructed to outline relevant characteristics for a study of privacy requirements. It shows how data of different sensitivity might be combined in one health record and different privacy requirements need to be be addressed by respective privacy policies.

```
Demographic Information:
  Name:            John Doe
  Date of Birth:   04.09.1977
  Place of Birth:  Berlin
Medical Entries:
01.06.1988 - (Dr. AB / XY Clinic)
  Diagnosis:  Cancer (Leukaemia)
  Treatment:  Planned chemo therapy
10.06.1988 - (Dr. AB / XY Clinic)
  Treatment:  Chemo therapy
  Result:     Everything OK
  Planned screening every year
01.07.1989 - (Dr. AB / XY Clinic)
  Treatment:  Cancer screening
  Result:     Everything OK
  [...]
30.08.2005 - (Dr. CD)
  Diagnosis:  Gonorrhoea infection
  Prescription: CIPROFLOXACIN 500 mg
15.04.2006 - (Dr. GH / UV Centre)
  Diagnosis:  Rheumatism
  Treatment:  Spinal X-ray
  Image:      <Attached X-ray>
  Prescription:  Novalgin
19.07.2006 - (Dr. IJ / ER)
  Treatment:  X-ray after traffic accident
  Image:      <Attached X-ray>
  Diagnosis:  Potential spinal injury
```

Listing 1: Exemplary Health Record.

Left out in the middle of the report are repetitive entries about the cancer aftercare. The rheumatism treatment includes a spinal X-ray that might be helpful for the medical opinion on a later potential spinal injury resulting from a car accident.
The health record addresses the patient's current situation with the latest entry for a potential spinal injury and shows useful X-rays from previous medical examinations. The further treatment process needs to add new data to the health record and needs to access parts of the old data.

Reflecting the idea of integrated care, the entries to the health record were made from different physician at different and independent institutions, all contributing to the ongoing treatment process for the patient. Accordingly, each involved computer system needs to access and process "foreign" data.

When Dr. IJ takes an X-ray to verify a potential spinal injury he might want to compare it to a prior spinal X-ray taken in the context of a rheumatism treatment. This raises the question how he knows about the prior data and how he can access it. A straightforward approach would imply that all medical documentation from every individual system is exported to a common EHR storage, allowing every party to access any data they need.

Since data from different systems is combined and leaves the individual security domains, this directly leads to the question whether everyone should have access to all medical data and how privacy of medical data can be preserved.

2.2 Confidentiality Requirements

The following parts of the paper will discuss technical details of a proposed security solution. Therefore, this section gives an informal description of exemplary security targets for the use case that need to be addressed in order to preserve privacy of shared medical data.

The listed confidentiality requirements are chosen to illustrate potential policy issues and even though commonly reasonable, do not necessarily offer maximum data protection.

Policy requirements might be different for every individual patient and under different situations. Consequently, data access privileges specified by the patient for the EHR are not static, but need to be dynamically extended or restricted.

In the example, the patient has no privacy concerns regarding the use of demographic data, rheumatism and the traffic accident, that can consequently be divulged to everyone. Data about cancer and the gonorrhoea infection are seen as sensitive and should not

have unlimited access. This requires the ability to effectively restrict data processing for everyone as chosen by the "owner" of the data, which usually is the patient.

In this use case example it is therefore not necessary to restrict the use of the rheumatism X-ray data. The sensitive data about the gonorrhoea infection, however, should only be visible to the EHR owner (patient). For the attending physician Dr. CD its visibility should be limited to the time period of treatment.

2.3 Data Representation in Electronic Health Records

Even though health records represented as free text are still in predominant use, standards for structured data representations in EHRs have been developed (e.g. (CEN/TS-15211, 2006)). Adding extra information on a meta-level, they not only ease data collection, data mining and reuse (cf. (Giere, 1986)) but additional allow the specification of security aspects at this level.

2.3.1 Genuine XML

Using XML to serialise the medical history described in Listing 2, the EHR would be structured by tags:

```
<healthRecord>
  <demographicInformation>
    <family_name>Doe</family_name>
    <given_name>John</given_name>
    <dayOfBirth>1977-09-04</dayOfBirth>
  </demographicInformation>
  <medicalHistory>
    <event date="1988-06-01" time="16:35:27">
      <diagnosis>Cancer</diagnosis>
      <treatment>Chemo therapy</treatment>
      <practitioner>DR. AB</practitioner>
    </event>
    [...]
    <event date="2005-08-30" time="10:35:27">
      <diagnosis>Gonorrhoea infection</diagnosis>
      <treatment>Ciprofloxacin 500 mg</treatment>
      <practitioner>DR. CD</practitioner>
    </event>
  </medicalHistory>
</healthRecord>
```

Listing 2: Health Record XML encoded.

The structure of the XML based EHR document needs to comply with an XML schema definition (Walmsley, 2004). This way, not only the structure of an EHR can be unified for simpler post processing, but specific tags or structures can be associated with semantic concepts.

Moving from free text towards structured data representation, the segment `<diagnosis>Cancer</diagnosis>` could be replaced with a specific tag `<cancerDiagnosis/>` or with another XML element of the type "cancerDiagnosis". This approach is used by HL7 CDA and EN13606 which represent the most commonly used and the latest development of EHR schemata, respectively.

2.3.2 Comments on HL7 CDA

HL7's Clinical Document Architecture (HL7, 2005) is not designed to support longitudinal records that cover complete accumulation of reports over time similar to the use case example above. A number of single documents could be used to represent the use case.

As mentioned for generic XML, CDA supports a standard XML schema defined structure as well as specific element tags associated with semantic concepts. CDA rel. 2 defines seven linked XML schema definitions, segmenting the XML structure into header and body, sections and entries. The semantic foundation for the contained elements is the HL7 Reference Information Model RIM (ISO/HL7-21731, 2006), complemented with fix vocabulary domains.

The information from the use case example can be represented by instances of the RIM classes Person, Procedure and Observation. A set of meta-data, e.g. a confidentialityCode, can be attached to each of these objects. Attached to an observation, the attribute specifies confidentiality rules limited to the object itself. When present with a person object, it refers to all entries related to the person. To allow for hierarchical confidentiality rules, a confidentialityCode may be specified at header, body, section, or entry level, each overwriting the more general.

Confidentiality rules are expressed using the vocabulary domain Confidentiality. It contains the following codes:

- low / normal / restricted / very restricted
- business / clinician / individual / substance abuse related / HIV related / psychiatry related / sexual and domestic violence related
- celebrity / sensitive / taboo

The vocabulary domain is a straightforward approach, but fairly incomplete, as it contains e.g. "HIV related", but no "cancer related" code. And although it defines the confidentiality intentions associated to the codes, they do not provide sufficient information to serve as machine processable security policies on their own. Nevertheless, they can be referred to as content types by higher level policies.

Additionally the RIM supports the use of roles, e.g. Patient, Employee, or special access roles. These might be referred to as types, similar to the confidentiality codes.

2.3.3 Comments on En 13606

Even though EN 13606 is a communication standard, it models the structure of the EHR with a two level approach, a reference model (EN-13606-1, 2007) and an archetype model (EN-13606-2, 2005). Using adequate archetypes, all data from the use case example can be represented and serialised in XML conforming to respective schemata (cf. openEHR (openEHR, 2007) which is an XML-based standard implementation close to EN 13606). In addition, using archetypes, which are bound to ontological concepts, semantic meta-information is available for all entries. This might aid the specification of security policies, as e.g. a cancer-related entry from the use case can be recognised as such, because it might be represented as a cancer-archetype.

On top of the reference and archetype model, EN 13606 defines its own security model (EN-13606-4, 2007). Using the concepts previously introduced in part 1 and part 2, part 4 defines an *access policy archetype*. (Medical) data that is represented as one or more *COMPOSITIONS* is complemented within a dedicated *Access policies FOLDER*. Each of these access policies itself is represented as a single *COMPOSITION*, whose archetype must conform to the specifications in part 4 of the standard. Additionally, a number of security categories are introduced with the security model:

- A Private entries shared with General Practitioner
- B Entries restricted to sexual health team
- C Entries accessible to administrative staff
- D Entries accessible to clinical support staff
- E Entries accessible to direct care teams
- F Private entries shared with several named parties
- G Entries restricted to prison health services

These categories can be used to mark data in an EN 13606 EHR at different levels of abstraction and thereby to assign respective policies, cf. Figure 1. Even though, this pragmatic approach gains some extra expressiveness through the use of flexible and extensible archetypes, it is still very limited in the kind of policies that can be specified. Although EN 13606 is still partly work in progress, it is to be doubted, that all confidentiality requirements of the use case described above, e.g. the delegation of rights, could be expressed without the integration of more comprehensive security standards. Therefore, a more general and flexible security model is described in the following section.

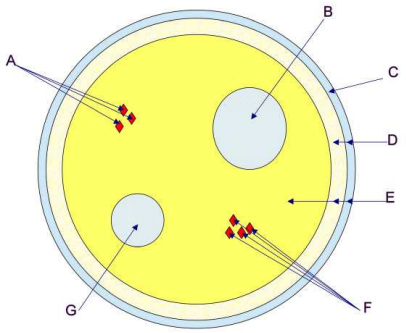

Figure 1: Access domains within 13606 EHRs(EN-13606-4, 2007).

3 EHR SECURITY MODEL

It is considered good practice to separate policies and mechanisms for access control and make the policy explicit. This allows independent implementation changes to enforcement mechanisms and opens the policy for external analysis and composition.

3.1 Policy (Description) Languages

From a mathematical point of view an access policy can be regarded as the set of all possible access decisions over the sets of subjects and objects supported by this policy. A definition of such an access policy is given by Woo (Woo and Lam, 1993): An authorisation policy is the 4-tuple (P^+, P^-, N^+, N^-) where each component is a subset of $\{(r,s,o) \mid r \in R, s \in S, o \in O\}$ over the set of subjects S, objects O and access rights R. P^+ and N^+ record the rights that are explicitly granted or denied. Whereas P^- and N^- record the rights that should not be explicitly granted or denied and are needed to define the semantics of policy composition.

Policy $A = (P^+, P^-, N^+, N^-)$ defines three authorisation relations for an authorisation request (r,s,o).

$$\begin{aligned} A grants\,(r,s,o) \quad &\text{iff} \quad (r,s,o) \in P^+ \\ A denies\,(r,s,o) \quad &\text{iff} \quad (r,s,o) \in N^+ \\ A\,fails\,(r,s,o) \quad &\text{iff} \quad (r,s,o) \notin P^+ \cup N^+ \end{aligned}$$

The policy representation as thus becomes irrelevant, as long as it can guarantee to be set-theoretical equivalent to the access matrix.

Policy description languages, such as the eXtensible Access Control Markup Language XACML

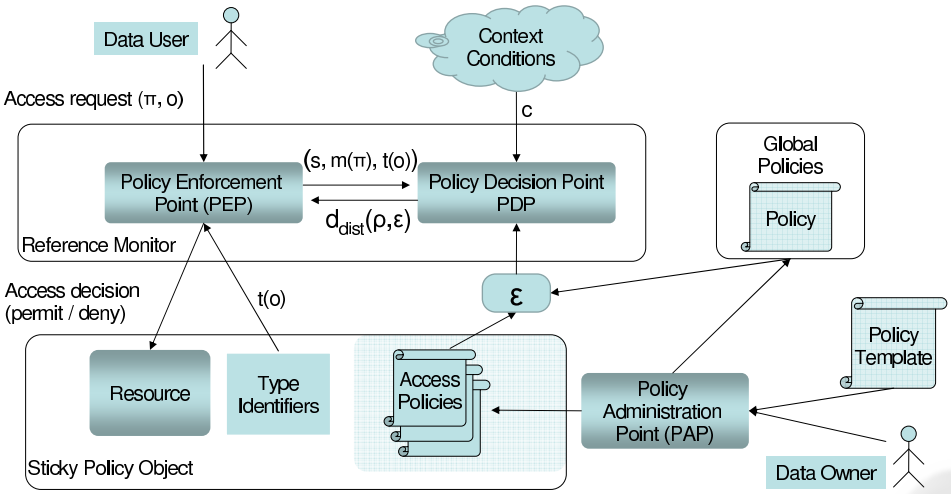

Figure 2: EHR Processing Model.

(XACML-2.0, 2005) or the eXtensible rights Markup Language XrML (ContentGuard, 2001) go one step further in their expressiveness and maintainability than simple *Access Control Lists* (ACL).

The ability of these languages to logically group objects and subjects, as well as the ability to evaluate environmental conditions, such as access-time, allow the creation of concise policies. These policies must be evaluated at the time of access in order to determine the access decision.

It is to be noted that with the use of these higher level policy languages the meaning of the term access decision is being extended. Since the rights that can be granted by a policy can reference complex processing concepts rather than simple access modes (read-/write), access control can be replaced by processing control. Consequently, these terms will be used synonymously, referring to the idea of processing control via a reference monitor.

3.2 Distributed Processing Control Architecture

A *Reference Monitor* is an instance that decides whether a process associated with the user $s \in S$ may execute a procedure $\pi \in \Pi$ on a resource $o \in O$ under the current system context $c \in C$. Therefore, it computes its decision for a request $\rho = (s, \pi, o, c) \in P = S \times \Pi \times O \times C$ as shown in equation (1).

$$d : P \rightarrow \{\text{grant, deny}\} \tag{1}$$

In a distributed environment a reference monitor may not know S, Π, O, and C in advance. So, locally, d cannot be fully be defined. A decision d_{dist} needs to be based on an appropriate security (access) policy. To make this local decision for the same request ρ in a distributed environment, a rights expression ε needs to be evaluated together with the decision request. The expression ε may describe a number of policies that explicitly or implicitly address $S_o \times \Pi_o \times o \times C$ for a specific resource object o. A rights expression might use the concept of rights R instead of procedures Π. R represents generic action concepts, e.g. "view" or "amend", whereas Π contains specific program code blocks. Hence, a mapping $m : \Pi \rightarrow R$ needs to be defined for the actual execution context. Similarly, with $t : O \rightarrow T$ the rights expression may refer to types of objects $t(o)$ rather than the object o itself. Given a request $\rho \in P$ and a rights expression $\varepsilon \in E$, the reference monitor needs to compute

$$d_{dist} : P \times E \rightarrow \{\text{grant, deny}\} \tag{2}$$

by matching the decision request against the represented policies.

In Figure 2 we show the complete processing model for distributed policy evaluation. The Data Owner creates a *Sticky Policy Object* at the *Policy Administration Point* (PAP), which he can also use to create global policies for all of his EHR data. The sticky data object combines EHR data with the actual Access Policy into a single XML file. It should be possible to support the policy creation process through the use of generic policy templates.

Data access from the Data User is only granted via the Reference Monitor which uses a modularised design, separating the *Policy Enforcement* from the *Policy Decision* component.

3.3 Sources of Authority

Only authoritative policies may be considered for the computation of d_{dist}. This means that not everyone is allowed to set permissions for a resource by defining and issuing arbitrary policies. Only the owner of a resource may author and authorise policies for it. Nevertheless, this authority could be extended by the owner to a third party, e.g. by making the extension part of a policy that grants the right to grant rights. The determination of authoritativeness of a policy is more complicated when privilege delegation or delegation of policy authoring is supported by the distributed processing control architecture.

Let $\varepsilon = (p_1 ... p_n)$ be a rights expression that consists of a set of policies by different authors $a(p_i)$ referring to a resource o. Further, let a_{owner} be the owner of o and let $v(\varepsilon)$ be a boolean function that is true, if and only if

$$\begin{array}{l} \forall p_i \in \varepsilon : a(p_i) = a_{owner} \vee \\ \exists p_{i-i} : p_{i-1} \text{ grants } a(p_i) \text{ to issue } (p_i) \end{array} \quad (3)$$

Then v is called verification of authority for ε. If $v(\varepsilon)$ is true, ε is called an authoritative rights expression and may be used to compute d_{dist}. With this definition the meaning of the term source of authority becomes clear, as all policy parts in an authoritative rights expression derive their authority from a single source a_{owner}. Any policy delegating rights or granting the right to issue new policies needs to be included in ε to allow the policy decision point to compute $v(\varepsilon)$.

4 XML POLICY LANGUAGES

This section compares two prominent policy description languages from the viewpoint of their applicability for the application domain.

4.1 XACML

The eXtensible Access Control Markup Language (XACML) is a declarative access control policy language and a processing model that is standardised by OASIS (XACML-2.0, 2005). The current version 2.0 was ratified in 2005.

4.1.1 Language Elements

PolicySet. The `<PolicySet>` element contains a set of `<Policy>` or other `<PolicySet>` elements and a *PolicyCombiningAlgorithm* to determine the joint evaluation of different elements.

Policy. The `<Policy>` element contains a set of rule elements and a *RuleCombiningAlgorithm* to determine the joint evaluation of the rules of the policy. It is the basis of an authorisation decision.

Combining Algorithms. XACML allows explicit positive and negative evaluation of rules (permit/-deny), as well as the combination of policies from different sources in a PolicySet for distributed policy generation. Combining algorithms are an essential part of the language specification. They are needed to derive an authorisation decision from potentially conflicting individual rules and policies. Standard combining algorithms are:

- Deny-overrides
- Permit-overrides
- First-applicable
- Only-one-applicable

Rule. The `<Rule>` element contains a policy expression that can be evaluated in isolation and provides the basic unit of policy management. The main components of a rule are its effect (permit/deny), target and potentially a condition that refines the applicability of the rule.

```
<Rule ...  Effect>
<Target>
...
</Target>
<Condition>
...
</Condition>
</Rule>
```

Target. The set of resources, subjects and actions to which rules and policies apply is called a target in XACML. Targets in policy elements define the scope of this element. If no restrictions have been made here the policy will have global scope.

Issuer/Delegation. As of version 2.0, XACML provides no mechanisms to describe a delegation policy as well as an issuer of a policy/delegation. In the current standard these have to be specified externally. Version 3.0 is currently in preparation and will add generic attribute categories and a policy delegation profile to the XACML specification.

4.1.2 Use Case Example

The XACML approach strictly separates authorisation policies and resources. Within XML-based re-

sources policies can be included and referenced via XPath (DeRose, 1999).

```
<HealthRecord>
 <Policy RuleCombiningAlgId="deny-overrides">
   <Rule Effect="Permit">
     <Target>
      <Subject><AnySubject/></Subject>
      <Resources>
        <ResourceMatch MatchId="xpath-node-equal">
        /HealthRecord/fileData </ResourceMatch>
      </Resources>
      <Action>view</Action>
     </Target>
   </Rule>
   <Rule Effect="Permit">
     <Target>
      <Subject>Dr. CD</Subject>
      <Resources>
        <ResourceMatch MatchId="xpath-node-equal">
    /HealthRecord/medHistory/event[Diagnosis="Gonorrhoea"]
        </ResourceMatch>
      </Resources>
      <Action>view</Action>
      <Condition FunctionId="date-less-than-or-equal">
        <Apply FunctionId="date-one-and-only"> </Apply>
        <AttributeValue>2002-03-22</AttributeValue>
      </Condition>
     </Target>
   </Rule>
 </Policy>
 <fileData>
    <demographicInformation>
   [...]
 </fileData>
 <medHistory>
   <event date="1988-06-01 16:35:27">
     <diagnosis>Cancer</diagnosis>
     [...]
   </event>
   <event date="2005-08-30 10:35:27">
      <diagnosis>Gonorrhoea infection</diagnosis>
      [...]
   </event>
 </medHistory>
</HealthRecord>
```

Listing 3: XACML Policy protected EHR.

4.2 XrML / MPEG-21

The origin of XrML is research on a "digital rights property language" (DPRL) by Stefik (Stefik, 1996). Version 2 of XrML developed at ContentGuard introduces a more generic approach to rights specifications. A revised edition was adopted by ISO as part 5 of the MPEG-21 standard. All subsequent statements refer to the MPEG-21 Version of XrML. Rights and other properties are represented by abstract concepts that are not bound to any context domain. Using XML namespaces, this basic XML structure can be extended with domain specific language elements replacing the abstract concepts. Therefore, it can be assumed that XRML can be adapted to the context of EHRs, a hypothesis that will be substantiated below.

4.2.1 Language Elements

License. A license in terms of XrML is a collection of grants allowing individuals to perform actions on specific resources. The `<license>` tag is the root of the XML tree and brackets all other relevant tags including `<grant>` or `<grantGroup>`, `<issuer>` and `<inventory>`. It does not contain any processible information itself. Alternatively, the non-obfuscated child nodes of a license can be replaced by an `<encryptedLicense>` which contains the same information, but needs to be decrypted for further processing.

Issuer. The `<issuer>` tag encloses a set of issuer-specific details about the circumstances under which he issues the license and a digital signature (Eastlake et al., 2002) for the license. With respect to the use case it could be the patient as a single source of authority signing the license.

Inventory. The `<inventory>` tag marks a part of a license that can be used to store anything referred to by a grant. By placing it in the inventory, redundancy, e.g. multiple principal or resource specifications, can be avoided when multiple grants refer to the same items. With respect to the use case, the inventory would be the place to include XML fragments of the EHR within `<digitalResource>` subsections, or to give an URI reference to an external EHR resource.

Grant or GrantGroup. Multiple subsections marked by `<grant>` tags may be present. A grant is the part of the license that specifies information relevant to decide whether a sub-procedure within a computer program should be executed with respect to the license (issuer's intention) or not. Like the `<license>`, the `<grant>` tag is only of syntactical nature. The relevant information is contained in the quartet of child nodes for principal, right, resource and condition, which all are conceptually abstract.

For each grant, additional pattern and delegation-control information can be stored in `<forAll>` and `<delegationControl>` tags respectively.

Therefore and with respect to the use case, the confidentiality requirements from Section 2.2 can be represented in the grant sections, including the intended delegation of privileges.

Principal, Right, Resource, Condition. Within each grant domain specific tags represent the abstract concepts principal, right, resource and condition. This means that there are no <principal> or <right> tags, but that these concepts can be substituted with domain specific tags, e.g. <hpcHolder> (for one specific holder of a health professional card) or <compareImage> (indicating the X-rays from the use case may be compared). These extensions to the abstract concepts are assembled in domain-specific XML-namespaces.

4.2.2 Use Case Example

The following code lists in an abridged form the required XrML language elements for the EHR use case example.

```
<license>
  <inventory>
    <digitalResource licensePartId="demoInfo">
      <XML>
        [Demographic Info]
      </XML>
    </digitalResource>
    [Cancer01]
    [...]
    [Gonorrhoea]
    [...]
    [Traffic Accident]
  </inventory>
  <grant>
    <export/>
    <digitalResource
     licensePartRef="demoInfo">
  </grant>
  [...]
  <grant>
    <[Dr. CD]/>
    <view/>
    <digitalResource
     licensePartRef="gonorrhoea">
    <notAfter>2006-02-30</notAfter>
  </grant>
  [...]
  <issuer> [Patient] </issuer>
</license>
```

Listing 4: XrML Policy protected EHR.

4.3 Comparison of XACML and XrML

Policy description languages differ in their ability to express certain concepts directly and efficiently as part of their language. Table 1 compares the support for different concepts in XACML and XrML.

Any XrML license is always granting. There are no denying syntax elements in the language. Therefore, the absence of a license never leads to false positive granting. This might be an advantage in distributed systems. XACML Policies instead have the ability to express negative authorisations and therefore can define explicit Policy/RuleCombiningAlgorithms for the inclusion of policies from different sources.

Table 1: Comparison between XACML and XrML.

	XACML	**XrML**
Explicit issuer	No element	Element
Condition support	Extensive	Extensive
Rights delegation	No	Yes
X500 naming	Supported	Not mandatory
X509 identities	Not supported	Supported
X509 attributes	Not supported	Not mandatory
Rule-signing	Indirectly through XML signatures	Directly supported
Encryption of content	Indirectly through XML encryption	Directly supported
Environment	Yes (time, etc.)	Yes (time, ticket, etc.)
Deny rules	Yes	No
Insertion of access rules	Easy [1]	Easy
Deletion of access rules	Easy [1]	Easy
Policy template support	Yes, through the use of `PolicySets`	No
Type identifiers	Not directly	Yes, (includes pattern matching based on XPath)

[1] might lead to changes in the policy (grant/deny rules)

Any requirement from Section 2.2 can be expressed with XrML and XCAML, because there is no limit to the integration of domain specific concepts. Domain independent requirements, e.g. the delegation of privileges, are featured by XrML itself.

Furthermore, the languages can express all elements of rights expression, as defined in the general security model in Section 3.2. The computation of $m(r)$ is in the responsibility of the policy enforcement point, referring to $t(o)$ is directly supported by XrML. e.g., it can refer to HL7 RIM attributes or EN13606 archetype using *resourcePatterns*.

5 CONCLUSIONS

In this paper we have examined the possibility to use existing XML policy languages that were developed

for digital rights management and the description of access policies for the protection of EHR data.

We foresee a need to mediate distributed data access, where data is stored, accessed and processed in a truly distributed fashion without the help of centralised policy mechanisms. Distributed data access, however, also requires a dedicated access control architecture, which we presented in Section 3 as a general model for access control in distributed processing environments, e.g. the medical IT environment described in the use case. Any concrete implementation of an policy enforcement mechanism can be analysed and compared with respect to this model.

The analysis of current EHR standards has shown that they are not ideally suited for reliable data protection and patient-controlled access restrictions. Instead, they should be used in combination with dedicated policy languages.

Section 4 presents two dedicated policy description languages that might be used to specify data access policies for EHR. A structural analysis and shortened example explains how these languages could be used. Even though a full policy description representing the use case could not be given for reasons of readability and length, their general applicability is shown. The two languages are compared face to face, outlining important differences when used for EHR protection.

An open issue and potential basis for further work is the formulation of a generic set of actions, rich enough for the fine-grained control over medical data in the workflow and simple enough for the patient to reliably apply in EHR policies.

REFERENCES

Apitzsch, F. (2007). Digital Rights Management for Electronic Health Records. In *Proceedings of CeHR International Conference 2007 (to appear)*.

Blobel, B., Nordberg, R., Davis, J., and Pharow, P. (2006). Modelling privilege management and access control. In *International Journal of Medical Informatics*, volume 75, pages 597–623.

BMG (2006). Die Spezifikation der elektronischen Gesundheitskarte. Bundesministerium für Gesundheit, Version 1.1.0, http://www.dimdi.de/static/de/ehealth/karte/index.htm.

CEN/TS-15211 (2006). Health informatics - Mapping of hierarchical message descriptions to XML. European Committee for Standardisation, http://www.cen.eu.

ContentGuard (2001). eXtensible rights Markup Language (XrML) 2.0, Specification.

DeRose, J. C. S. (1999). XML Path Language (XPath). W3C Recommendation, http://www.w3.org/TR/1999/REC-xpath-19991116.

Eastlake, D., Reagle, J., and Solo, D. (2002). RFC3235: Extensible Markup Language - XML-Signature Syntax and Processing. http://www.rfc-editor.org/rfc/rfc3275.txt.

EN-13606-1 (2007). Health informatics - Electronic health record communication - Part 1: Reference model. European Committee for Standardisation, http://www.cen.eu.

EN-13606-2 (2005). Health informatics - Electronic health record communication - Part 2: Archetypes. European Committee for Standardisation, http://www.cen.eu.

EN-13606-4 (2007). Health informatics - Electronic health record communication - Part 4: Security. European Committee for Standardisation, http://www.cen.eu.

Giere, W. (1986). *BAIK - Befunddokumentation und Arztbriefbeschreibung im Krankenhaus*.

HL7 (2005). HL7 Clinical Document Architecture, Release 2.0, Normative Edition.

ISO/HL7-21731 (2006). Health informatics - HL7 version Reference information model Release 1).

Karjoth, G., Schunter, M., and Waidner, M. (2003). Platform For Enterprise Privacy Practices: Privacy-enabled Management Of Customer Data. In *2nd Workshop on Privacy Enhancing Technologies (PET2002)*, volume Lecture Notes in Computer Science 2482, pages 69–84. Springer Verlag.

Mont, M. C., Pearson, S., and Bramhall, P. (2003). Towards Accountable Management of Identity and Privacy: Sticky Policies and Enforceable Tracing Services. In *Proceedings of the 14th International Workshop on Database and Expert Systems Applications*, page 377. IEEE Computer Society.

openEHR (2007). openEHR Release 1.0.1. http://www.openehr.org.

Stefik, M. (September 18th, 1996). The Digital Property Rights Language, Manual and Tutorial, Version 1.02. Technical report, Xerox Palo Alto Research Center, Palo Alto, CA.

Walmsley, D. C. F. P. (2004). XML Schema. W3C Recommendation, http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/.

Wang, X. (2005). Desing Principles and Issues of Rights Expression Languages for Digital Rights Management. In *Proceedings SPIE, Conference on Visual Communications and Image Processing*, volume 5960, pages 1130–1141.

Woo, T. Y. C. and Lam, S. S. (1993). Authorizations in Distributed Systems: A New Approach. *Journal of Computer Security*, 2(2-3):107–136.

XACML-2.0 (2005). eXtensible Access Control Markup Language (XACML). OASIS-Standard, http://www.oasis-open.org/committees/xacml.