Complex Systems Informatics and Modeling Quarterly
CSIMQ, Article 62, Issue 11, June/July 2017, Pages 1–19
Published online by RTU Press, https://csimq-journals.rtu.lv
https://doi.org/10.7250/csimq.2017-11.01
ISSN: 2255-9922 online

Modeling and Recognizing Policy Conflicts with Resource Access Requests on Protected Health Information

Raik Kuhlisch*

Institute of Computer Science, University of Rostock,
Albert-Einstein-Str. 22, 18059 Rostock, Germany

raik.kuhlisch@uni-rostock.de

Abstract. This article discusses potential clashes between different types of security policies that regulate resource access requests on clinical patient data in hospitals by employees. Attribute-based Access Control (ABAC) is proposed as a proper means for such regulation. A proper representation of ABAC policies must include a handling of policy attributes among different policy types. In this article, we propose a semantic policy model with predefined policy conflict categories. A conformance verification function detects erroneous, clashing or mutually susceptible rules early during the policy planning phase. The model and conflicts are used in a conceptual application environment and evaluated in a technical experiment during an interoperability test event.

Keywords: Hospital intra-enterprise policy conflict, policy compliance verification, information security, knowledge representation.

1 Introduction

Hospitals as crucial facilities of public health systems are categorized as critical infrastructure and are also constrained by additional regulation due to the sensitivity of the processed medical data. In acknowledging this strong correlation between safety and security, public administrations merged formerly separated safety and information security programs into consolidated guidelines that are considering information security as a fundamental corner stone of maintaining the availability, safety, and proper functioning of such critical infrastructure.

The information security of the processed data, applications, and health information technology (IT) systems is safeguarded through an information security management (ISM). Its defined objectives and controls assure a proper regulation of access to facilities that process information as well as the disclosure of protected (health) information [1].

* Corresponding author

Reference: R. Kuhlisch, “Modeling and Recognizing Policy Conflicts with Resource Access Requests on Protected Health Information,” Complex Systems Informatics and Modeling Quarterly, CSIMQ, Issue no. 11, pp. 1-19, 2017. Available: https://doi.org/10.7250/csimq.2017-11.01

The rules for any legitimate system access or information disclosure are documented in regulatory, compliance, and enterprise constraints, however, are usually not immediately machine processable. Access Control Systems (ACS) are utilized to formalize and combine any relevant constraints into commonly processable rulesets [2]. Attribute-based Access Control (ABAC) is one model capable of implementing a top-down-driven ISM in a hospital environment with the inherent advantage of explicitly concentrating on formalizing, combining, and processing access and disclosure regulation through sets of policies [3]. The latter contain legally binding rulesets, enterprise objectives, and general information security concerns in a formalized and processable technical representation. The eXtensible Access Control Markup Language (XACML) [4] perfectly fits to express fine-grained ABAC policies. Due to multiple attributes, context-sensitive access requests are more precise than, e.g. requests derived from the Role-based Access Control (RBAC) model which is quite common in hospital information systems.

In the ABAC model, a permission to a protected information object or resource is described with a number of subject, object, or environmental attributes. Each attribute is represented by key-value pairs and might be gathered at runtime to evaluate a dynamic authorization decision. During such evaluation a security context is created and anchors approvals to access an information object [5, p. 18]. In the RBAC model, the authorization decision to an object or a resource relies solely on an assigned role to a subject.

However, the syntactic and semantic correctness of all attributes must be ensured in both information security policies and access control policies. The multitude of simultaneously applied policies with varying degrees of abstraction and an increasing use of IT in a hospital lead to an ever-growing risk of individual policies conflicting with each other. Those collisions may paralyze the ACS ability to decide on the legitimacy on access requests and cause a denial of service situation or a confidentiality breach, effectively crippling the health IT ability to function and perform properly.

Therefore, assessing how policy conflicts can be identified and mitigated in order to prevent illegitimately assigned access rights or violating any data disclosure laws are being observed in this work. Previous approaches to policy conflict detection cannot be transferred easily to a hospital information system because their characteristics do not consider policies of different types. Usually, individual information systems are safeguarded locally with isolated access rules. There is a lack of visibility into the different types of policies for managing patient data access, which takes into account the information management of a hospital. For this reason, it is necessary to work out the relationships and characteristics of the different policy concepts (e.g. impacts on safeguards for health information through a patient consent and rules of the operational and organizational structure). This forms the basis for the analysis of policy realms.

This contribution presents a policy model and related specific conflict categories. Both are applied in a conceptual application environment and evaluated afterwards. By using above mentioned IT artifacts, clashes in and between access control policies, patient privacy policies, and information security policies can be recognized, effectively enabling a focused mitigation or ideally a correction. Therefore, the management of a hospital information system is supported on a tactical and operational levels.

The remainder of the article is structured as follows: Section 2 gives an overview of the research approach of the presented work. The Section 3 presents policies that control the resource access demands on patient data and discusses the related work. The Section 4 outlines a semantic policy model and introduces policy conflict categories. The Section 5 is concerned with the presentation of the policy management system which uses both the model and the conflict definitions. The Section 6 provides a discussion considering the evaluation of the developed model and conflicts and outlining implications for practice. Finally, a concluding section summarizes the key findings of this article.

2 Research Approach

The starting point for the investigation of the problem areas outlined in Section 1 is how policy-based information security management can be used to identify and deal with conflicts between rulesets in a computer-assisted manner. The derived research questions are:

- *Access context for protected health information:* What are the mandatory rules for operational access to patient data in a hospital and which policy concepts represent these rules?
- *Policy representation:* How can policies be formalized to enable a computer-assisted verification?
- *Policy conflict detection:* How can conflicts be classified with policy concepts and what are the impacts of individual policy concepts on other policies?
- *Policy conflict resolution:* How can tool support be used to test policies for conflicts and correct them if necessary?

The following overall research objectives intend to address the stated problems:

- Legally binding regulations as well as internal and regulatory requirements must be combined in order to differentiate the policy concepts. The nationally applicable orientation guide for hospital information systems ”Orientierungshilfe Krankenhausinformationssysteme” (OH-KIS) [6] of the German Working Group for Health, Social Affairs and Technology of the Federal and State Privacy Commissioners provides a significant foundation and relevant information for the mapping of data protection-compliant accesses to the hospital’s service processes. This information must be used for the differentiation of the policy concepts.
- In addition to the analysis of the dependencies of regulations and guidelines in a hospital, a semantic model for such policies is to be defined.
- Based on the policy model, the conflicts between the policies are analyzed.
- Policy model and policy conflicts are to be integrated into an application concept in order to demonstrate the concrete implementability. A prototypical application underlines the evaluation of the application concept, including the policy model and policy conflicts.

The used research approach for our work is Design Science Research (DSR) [7] which aims to describe an information system problem closely, to develop iteratively a solution to the problem (expressed as built design or IT artifacts) and finally to evaluate it. According to the phase model of Peffers et al. [8] this DSR research approach can be outlined as follows (phases are indicated with italic):

1. The first phase starts with *problem identification* and *motivation*. The development of the IHE[1] white paper on access control in healthcare scenarios [2] revealed that the handling of policy conflicts of different types in distributed environments is very complex. The concrete contents of the different policy types are not elaborated in this white paper. Initial research questions could be derived from this and raised, among other things, the question of how violations of information security can be recognized by an investigation of policy conflicts.

[1] IHE stands for the initiative ”Integrating the Healthcare Enterprise” and is an organization for the profiling of existing industry standards for the implementation of defined scenarios of a digitized healthcare system.

Although the topic field (security) policy is extensively addressed in the literature, policy conflicts in the health IT environment are not considered in detail. The published state of the art is largely restricted to similar policies, such as network policies or authorization policies. The subject area policy conflicts in the health care system was limited to the hospital. A carried out case study confirms that operational access requests are not justified by information security policies [9].

2. From these activities, requirements for the solution have been developed and possible *design artifacts defined* that addressed the problem adequately (cf. [10]). The research questions were refined and the *objectives defined* in the subsequent second phase.
3. After the definition of objectives, the actual process could start to *develop the design artifacts* (cf. [10], [11]). By separating the scope of tasks for policies, an exact analysis of potential conflicts in access requests to protected health data is achieved. Starting from established frameworks of information security, policy concepts were successively linked with further concepts into a policy model. This was done by incorporating amongst other things the policy concepts of the IHE white paper [2] and the orientation guide for hospital information systems [6].

   Contributions to policy conflicts from the literature were applied to the policy model. Further specific policy conflicts (dedicated to the hospital scope) could be defined as well.
4. The policy model and the conflict definitions were integrated and transferred to an application concept. This *demonstrates* how these artifacts might be applied (cf. [3]).
5. The subsequent *evaluation* phase clarifies the extent to which the research objectives fit into the results of the evaluation. This article summarizes findings of previous work that is described in the phases before and presents also these results.

3 Characteristics of Resource Access Demands on Patient Data

This section deals with the description of the access context for protected health information and outlines our policy model that has been developed (cf. Section 4). In addition, it presents related work in security policy management in healthcare and conflict resolution techniques.

The goal of a policy organization is to combine various policy concepts thus making them comparable and identifying clashes so that the policy concepts can ultimately be implemented. Policy concepts in healthcare environments include [2]

- rules for protecting medical data from illicit disclosure (policy concept *compliance*),
- patient's constraints given in a consent (policy concept *patient consent*), and
- constraints derived from the intended use of a certain healthcare system (policy concept *purpose of use*).

A proper information security management (ISM) puts a stop to illegitimately disclosure of protected health information. Preferably, access demands derived from the policy concepts described above are directed at compatible security policies setup. It is to be noted that such policies must be implemented and enforced adequately and in context of their individual criticality. The Unified Modeling Language (UML) class diagram in Figure 1 depicts the placement of policy types to the information security management and illustrates the policy-based protection of health information:

- The ISM starts with an *Information Security Policy* that holds general principles, fundamental objectives and responsibilities for the protection of any information being processed in an organization.
- The *Information Access Policy* is the logical part that details the lawful access to any protected health information in accordance with terms of compliance and regulatory mandates of an organization. Access demands are managed by dedicated subsystems of the entire hospital information system (HIS).

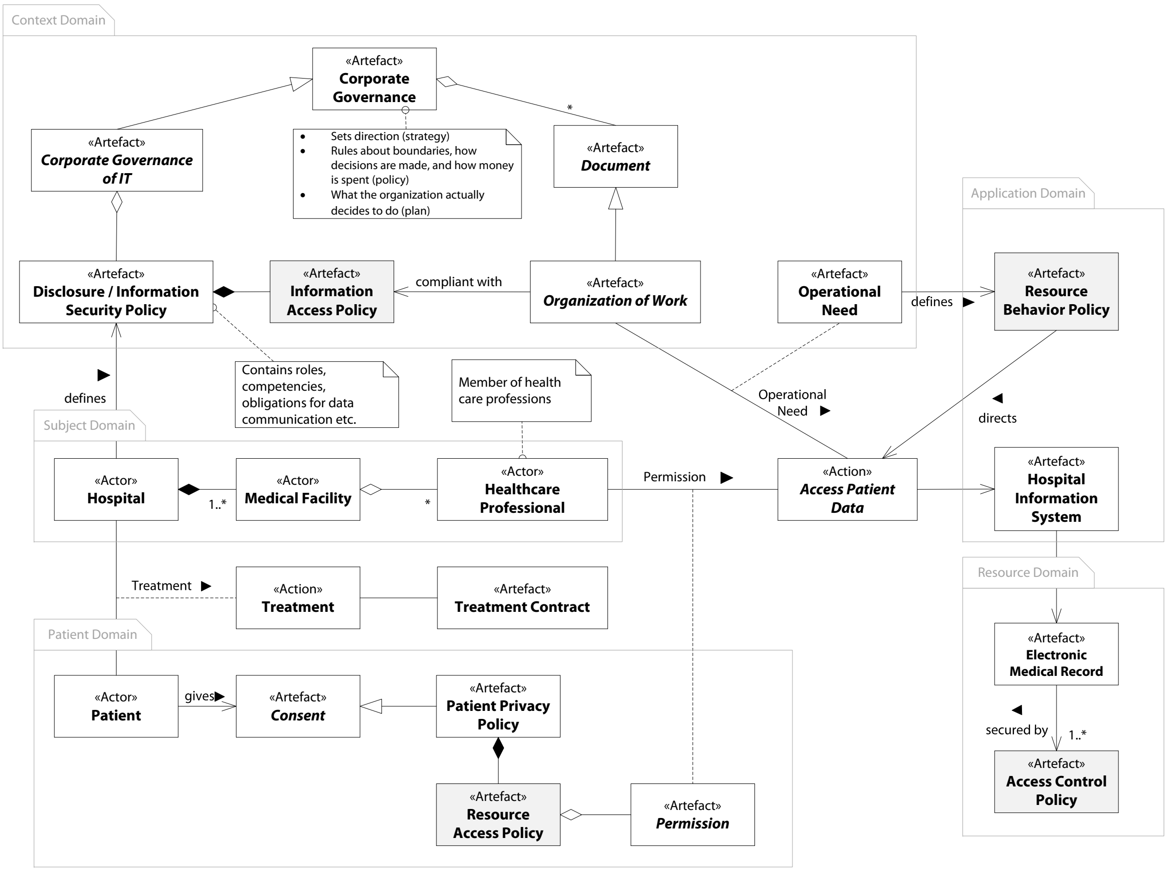
Context Domain
«Artefact» Corporate Governance
«Artefact» Corporate Governance of IT
Sets direction (strategy)
Rules about boundaries, how decisions are made, and how money is spent (policy)
What the organization actually decides to do (plan)
«Artefact» Document
*
«Artefact» Disclosure / Information Security Policy
«Artefact» Information Access Policy
compliant with
«Artefact» Organization of Work
«Artefact» Operational Need
defines ▶
Application Domain
«Artefact» Resource Behavior Policy
◀ directs
defines ▶
Contains roles, competencies, obligations for data communication etc.
Member of health care professions
Operational Need ▶
Subject Domain
«Actor» Hospital
1..*
«Actor» Medical Facility
*
«Actor» Healthcare Professional
Permission ▶
«Action» Access Patient Data
«Artefact» Hospital Information System
Resource Domain
Treatment ▶
«Action» Treatment
«Artefact» Treatment Contract
«Artefact» Electronic Medical Record
◀ secured by
1..*
«Artefact» Access Control Policy
Patient Domain
«Actor» Patient
gives▶
«Artefact» Consent
«Artefact» Patient Privacy Policy
«Artefact» Resource Access Policy
«Artefact» Permission

Figure 1. Policy types involved for accessing protected health information

- A *Resource Behavior Policy* reflects the operational need or purpose of use. That is, certain subjects might act on certain resources which are to be processed by information systems, such assignments are an essential part of this type of a policy.
- A *Patient Privacy Policy* is a fundamental prerequisite for giving permissions for processing health data of a patient. It is derived from the patient consent and yields potential access permits.
- Viewed logically, a *Resource Access Policy* holds specific characteristics of authorizations to certain individuals and organizations in order to use relevant applications (e.g. Electronic Medical Record [EMR] system) to the extent of the agreed purpose-of-use in the given patient consent and organization of work of the facility.
- Essentially, permissions are access rights that are assigned to subjects or roles in a discrete access context. Considering that, an additional *Access Control Policy* must be defined that holds these assignments. Access demands by hospital's employees intersect with potential permissions yielding the need-to-know realm.

In summary, all these types of policies presented in Figure 1 implement an Attribute-based Access Control (ABAC). As mentioned before, ABAC policies inject additional attributes from all relevant information systems such as subject, object or environment conditions into an authorization decision process. With that said, syntactic and semantic correctness of these attributes must be ensured in all policy types in order to justify and finally affect an authorization decision. We developed a proper policy representation approach and management of attribute values discussed in Section 4 that tackles the syntactic and semantic correctness of policy attributes. This handles potential interoperability issues among attributes in contrast to atomic values normally used in ABAC [3].

Related Work

A security policy class hierarchy was already modeled by the Health Level Seven (HL7) Security Work Group [12]. Different viewpoints detail business and engineering aspects in a domain analysis model dedicated for healthcare environments. From a security policy point of view, the domain analysis model is highly focusing on access control. Role-based Access Control (RBAC) [13] is considered as the access control paradigm of choice in this domain analysis model which is along with the previously mentioned orientation guide for hospital information systems [6]. RBAC states that authorization decisions on the access to certain objects rely on pre-defined roles which are assigned to subjects in user sessions. We, however, regard RBAC policies as not entirely suitable for representing actual workflows in hospitals and propose the ABAC model [3]. A top-town information security management as well as the identification of policy clashes are not addressed by the domain analysis model. However, we adopted the flavors of access control policies to our model (authorization policy, constraint policy, delegation policy, refrain policy, and obligation policy).

Our developed policy model is based on Semantic Web technologies. The policy languages Rei [14] and KAoS [15] use that approach as well. These languages are rather domain-independent and cope with authorization policies. We apply our policy model to the hospital's organization of work plus security and privacy rules. That is why our definitions of policy conflicts are more extensive and comprehensive. Policy clashes occur on different levels, not only with authorization policies.

Mouelhi et al. [16] introduce a Model-driven Engineering (MDE) process that transforms an access control policy into security components. These components can be integrated into executable program code and are actually the Policy Enforcement Point (PEP) as well as the Policy Decision Point (PDP). These two components regulate the authorization request and decision regardless of the business logic of a software application. An access control policy is specified in a generic meta model. With this meta model, RBAC policies can be described. A verification routine checks the conformity of the specific policy to the meta model as well as existing conflicting rules for permission, prohibition or obligation. The MDE process also provides for the generation of Java code, which generates XACML program code for a PEP as well as a PDP. This procedure provides only rudimentary checks on policy conflicts. It is only able to uncover contradictory permissions and obligations. Although the policy meta model is generic the support of other policy types such as a patient privacy policy is missing. Also, no statements are made about how policies can be related to each other to determine dependencies between different policy types [16].

4 Semantic Policy Model

In a previous work [17], we introduced a semantic policy model that expresses the policy concepts through the policy types from Section 3: We rely on ontologies as a means of a proper policy representation. An important advantage to this knowledge representation approach is the linking of knowledge with inference and constraints. These are rules for the preparation and logical reasoning of new contexts and links (and thus the derivation of new knowledge) as well as to ensure their validity.

The policy ontologies are engineered according to the frames concept [18]: Classes (*class frames*) stand for a lot of similar objects. These classes can be assigned properties and relations in the form of attribute sets (*slots*). With the help of further features (*facets*), slots can be filled and controlled with values (*filler*). Classes can be instantiated (instance frames) and represent individual concepts or objects of the class. Through the supported concept of specialization, Instance Frames inherit all the properties of Class Frames and can also include additional properties [19].

The fundamental design goal of the policy model is to use strictly externalized values of policy attributes since this at least simplifies the conflict handling if not preventing it altogether. As a modeling approach, the frame paradigm is used because the filler concept can be linked perfectly with terminology services.

The CTS2 – linked data edition (CTS2-LE) [20] is an implementation of the Common Terminology Services 2 (CTS2) [21] from the Object Management Group (OMG). The CTS2 specification simplifies the definition of identifiers and interrelations of terminology concepts via a standardized interface. Initially intended for representing medical code sets, CTS2 provides various means to represent arbitrary code sets and value sets. The CTS2-LE terminology service enables the definition of multiplicities as well as constraints on properties of resources of the Resource Description Framework (RDF) [22] via so-called RDF signatures. RDF and the RDF Schema [23] describe structural constraints according to the CTS2 information model. The underlying rule engine ensures that the frames paradigm is properly applied to the RDF signatures. Thus, instances of RDFS classes in RDF graphs (i.e. representation of terminology concepts) are in accordance with class frames and fillers.

We adopted the CTS2-LE implementation with the frames-oriented rule engine and enhanced it with RDF signatures of our policy types presented in Section 3. Each policy type is written with the RDF syntax ”Turtle” and holds specific policy attribute types in a respective ontology (in total five). Inheritance structures of classes and multiplicities are used in the RDF signatures as well. Figure 2 shows a graphical representation of a sample policy rule from a Resource Access Policy. The policy statement *”I hereby authorize physicians at Clinic A to use the ’Historical Database’ application in order to access all my lab data for the purpose of medical treatment”* consented by a patient can be linked with a terminology server (”controlled vocabulary” layer). Different layers organize a proper linkage of policy attributes with coded attributes that are managed by that server. The RDF signatures of our different policy types belong to the ”policy schema” layer. A proper separation of policy types leads to cross-linked attribute definitions. The next layer ”restriction policies” represents normative policies with RDF instances of the ”policy schema” layer. These policies are in accordance with the organization of work and tasks which are to be expressed by means of a Resource Behavior Policy and an Information Access Policy. New policy instances are ranged in the last layer ”policy instances”.

In summary, Figure 2 reveals possible clashes of a Resource Access Policy with other policy types. Thus, Resource Behavior Policies and Information Access Policies also need to be in a suitable alignment with the above policy statement.

Using this policy model and the proposed linkage with a terminology server, coded attributes in ABAC policies get more semantics. A proper attribute definition is important for both activation of a policy and decision on a policy.

4.1 Policy Conflict Categories

Given the policy model potential clashes between the policy types can be analyzed. Previous work on policy management shows that concise conflict categories are in place. We applied these categories on our policy model and defined further categories that indicate conflict ability. The following policy conflict categories are defined for the policy model:

- *Goal conflicts:* Policy subjects, resources and actions can overlap. For instance, special constellations may lead to a conflict of duties, conflict of interests or self-management [24].
- *Modality conflicts:* Authorization policies contradict with each other when permissions are set in two ways: One assigned permission to a subject permits access to a resource whereas another one denies it. The same servers for obligations (i.e. actions that must be

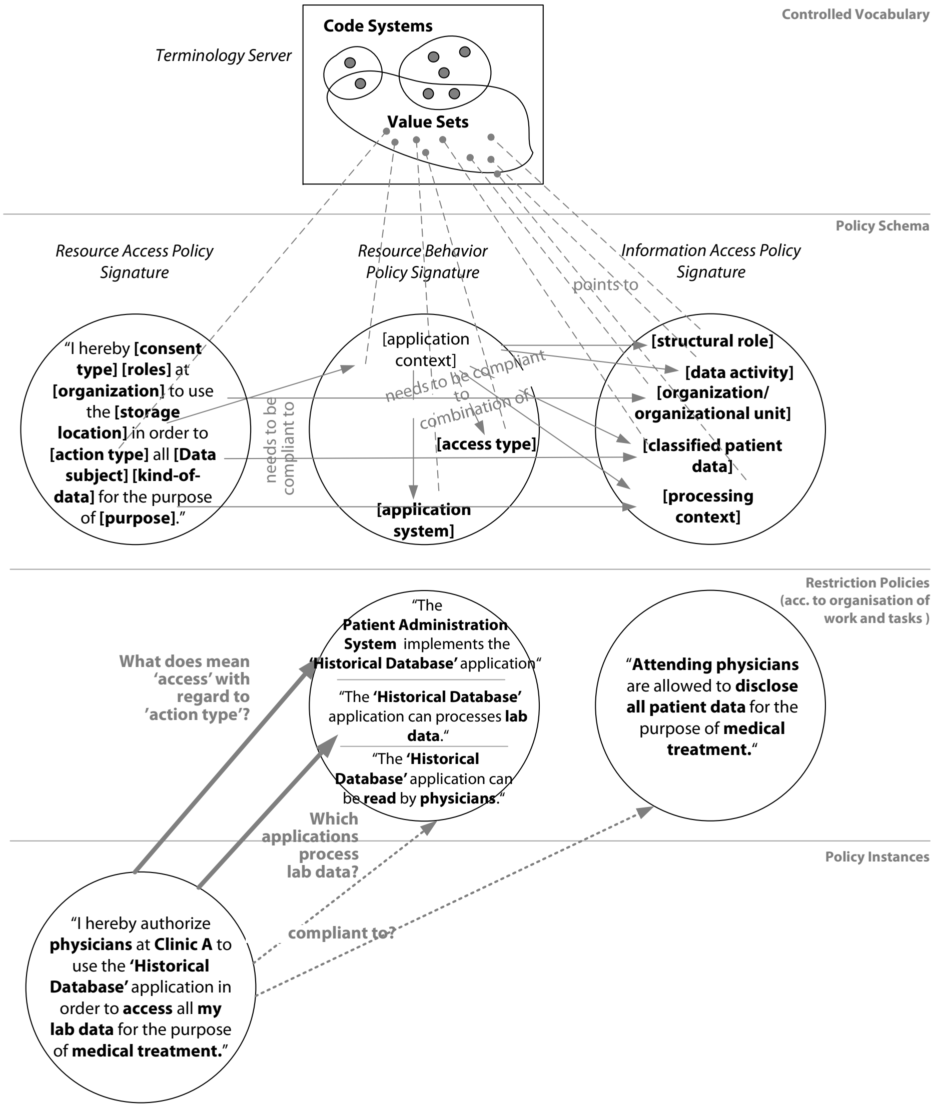
Controlled Vocabulary
Code Systems
Terminology Server
Value Sets
Policy Schema
Resource Access Policy Signature
Resource Behavior Policy Signature
Information Access Policy Signature
points to
"I hereby [consent type] [roles] at [organization] to use the [storage location] in order to [action type] all [Data subject] [kind-of-data] for the purpose of [purpose]."
needs to be compliant to
[application context]
needs to be compliant to combination of
[access type]
[application system]
[structural role]
[data activity]
[organization/ organizational unit]
[classified patient data]
[processing context]
Restriction Policies (acc. to organisation of work and tasks)
"The Patient Administration System implements the 'Historical Database' application"
"The 'Historical Database' application can processes lab data."
"The 'Historical Database' application can be read by physicians."
"Attending physicians are allowed to disclose all patient data for the purpose of medical treatment."
What does mean 'access' with regard to 'action type'?
Which applications process lab data?
compliant to?
Policy Instances
"I hereby authorize physicians at Clinic A to use the 'Historical Database' application in order to access all my lab data for the purpose of medical treatment."

Figure 2. Policy relationships on the basis of a terminology service

executed in conjunction with an authorization or action that are refrained from execution) [24].

- *Domain conflicts:* This semantical conflict category treats with functional dependencies among policy attributes. Since coded attributes are used for describing a policy state, the codes might contradict when different policy types apply simultaneously. To tackle code relations, subsumption relationships can be defined so that
$$CodedValue_{PolicyTypeA} \sqsubseteq CodedValue_{PolicyTypeB}$$
applies.
- *Data conflicts:* Data conflicts can be significantly mitigated or even avoided by using the CTS2 terminology service. Policy attributes with stored code systems or value sets can be linked with additional codes via the CTS2 Map Services to organize semantically same attributes. Thus, different identifiers (i.e. designators) and different scales of a code value are available.
- *In-congruent policy concerns:* Erroneous rules may occur with respect to different policy concerns. For instance, if no patient consent is given to a particular resource access, physicians must be refrained from accessing the patient data. This requires that subjects, resources and actions must be in accordance with the rules from compliance and the defined purpose of use of dedicated information systems and applications.

- *Special conflicts:* Based on the policy types, specific policy conflicts can be defined. For instance, if a patient limits access to his or her data for the purpose of treatment in a Resource Access Policy, but this is not compatible with the existing processing purposes (and associated accesses to application systems) because no intended recipient of information from an Information Access Policy can ensure treatment under these conditions.

  Another policy clash might occur if the access to an application system is implemented correctly (amongst other things via an Information Access Policy) but no authorization is given. In this case a conflict between a Resource Behavior Policy and an Access Control Policy exists.

4.2 Recognizing Policy Conflicts

As a query language, the SPARQL Protocol and RDF Query Language (SPARQL) [25] is predestined for RDF support. Since the policy model is in RDF, queries about policy conflicts can also be formulated implementation independently with SPARQL in order to recognize them.

All defined policy conflicts can be identified using this approach. For instance, the following query recognizes a conflict of duty (i.e. an overlapping between subjects and objects among two policies). Other conflicts can be identified analogously:

```
PREFIX bp: <urn:policy:signatures:basic#>
PREFIX prop: <urn:policy:signatures:properties#>
SELECT ?p1 ?p2
WHERE {
   ?p1 a bp:PolicyDescription .
   ?p2 a bp:PolicyDescription .
   # same subjects
   ?p1 prop:subject [
      prop:code ?subject
   ] .
   ?p2 prop:subject [
      prop:code ?subject
   ] .
   # same objects/targets
   ?p1 prop:rule [
      prop:target [
         prop:code ?target
      ]
   ] .
   ?p2 prop:rule [
      prop:target [
         prop:code ?target
      ]
   ] .
   # different actions
   ?p1 prop:rule [
      prop:action [
         prop:code ?action1
      ]
   ] .
   ?p2 prop:rule [
      prop:action [
         prop:code ?action2
      ]
   ] .
   FILTER ( ?p1 != ?p2 ) .
   FILTER ( ?action1 != ?action2 ) .
}
```

4.3 Support for Relationships Among Coded Policy Attributes

A special handling of domain conflicts is required. Policy attributes (e.g. policy actions) can be distributed taxonomically across multiple policy types. This can cause inconsistencies if contradictory attribute values are used in policies. We define subsumption relationships between policy actions with a dedicated RDF signature:

```
@prefix : <urn:policy:signatures:conflicts#> .
@prefix bp: <urn:policy:signatures:basic#> .
@prefix rdfs: <http: //www.w3.org/2000/01/rdf−schema#> .
@prefix sig: <urn:negros:signatures#> .
:Subsumption
  a rdfs:Class ;
  sig:propertyConstraint [
    sig:onProperty :superior ;
    sig:range bp:Action ;
    sig:min 1 ;
    sig:max 1 ;
  ] ;
  sig:propertyConstraint [
    sig:onProperty :subordinate ;
    sig:range bp:Action ;
 ] .
```

An instance of this RDF signature is given in Figure 3 which shows the graph structure in a UML-like notation (generalization is indicated with dashed lines whereas associations are shown with solid lines). The data activity ”USE” is defined as a superior action of the access types ”READ” and ”EXECUTE” dedicated for describing accesses to healthcare information systems. The access type ”READ” includes a specific access control action of a healthcare information system (in this case ”RegistryStoredQuery”). Each attribute is used in different types of policies and linked with a terminology service.

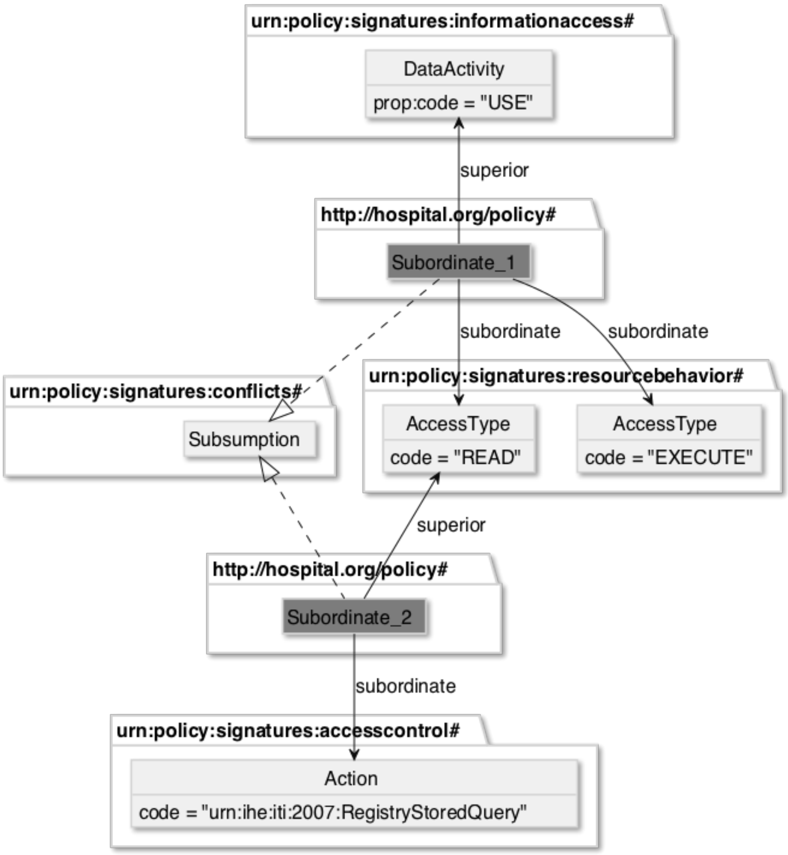
urn:policy:signatures:informationaccess#
DataActivity
prop:code = "USE"
superior
http://hospital.org/policy#
Subordinate_1
subordinate
subordinate
urn:policy:signatures:conflicts#
Subsumption
urn:policy:signatures:resourcebehavior#
AccessType
code = "READ"
AccessType
code = "EXECUTE"
superior
http://hospital.org/policy#
Subordinate_2
subordinate
urn:policy:signatures:accesscontrol#
Action
code = "urn:ihe:iti:2007:RegistryStoredQuery"

Figure 3. Object model for subsumption relationships

Another SPARQL query identifies the policy attribute codes that are used in a candidate policy but do not have any higher-level code relationship. By means of a stringent subsumption definition injuries would be visible:

```
PREFIX conf: <urn:policy:signatures:conflicts#>
PREFIX prop: <urn:policy:signatures:properties#>
SELECT DISTINCT ?policy ?code
WHERE {
   FILTER NOT EXISTS {
      ?x conf:subordinate [
         prop:code ?code
      ] .
   }
   GRAPH ?policy {
      ?z prop:action [
         prop:code ?code
      ] .
   }
   # exclude normative policies
   FILTER ( ?policy != <urn:policy:normative:informationaccess> )
   FILTER ( ?policy != <urn:policy:normative:resourcebehavior> )
}
```

5 Conceptual Application Environment

According to the DSR approach, a demonstration of the utility of the design artifacts ”policy model” and ”policy conflicts” is necessary. For this purpose, we introduce a Policy Management System (PMS). This system is the central database for policies and provides functions to detect policy conflicts (design artifact of the type ”instantiation”). In addition, this system provides basic policy management tasks which include querying, as well as providing and storing policies. It is based on a standard policy vocabulary configured in a Common Terminology Service based on CTS2-LE.

The overview in Figure 4 shows how the individual logical components of the PMS are integrated into the hierarchical architecture design with the three layers (1) external interface (Web Service Endpoint), (2) application core (Modules), and (3) data management (RDF Store). The entire application logic is divided into the seven components, which have a common interface for the users of the Policy Repository Service:

- *Query Module:* This component realizes query functionality on RDF graphs to find policies. Here the broad spectrum of Semantic Web technologies is provided by SPARQL.
- *Import Module:* This component enables the permanent storage of policies. A transformation logic can generate from policies in the representation format XACML policies in RDF. This is determined by means of the media type (*application/xacml+xml*). Furthermore, this component uses the Semantic Validation Module to identify conflicts with currently stored policies.
- *Export Module:* This module loads stored policies and outputs them in XACML or pure RDF format.
- *Semantic Validation Module:* This component uses SPARQL templates to check whether there are conflicts (e.g. with the codes used) with other policies. It continues to provide normative policies, which are the basis for examining policy instances.
- *CTS2 Module:* This component provides the interface to the CTS2 system and allows the communication of coded policy attributes.
- *RDF Signatures Validation Module:* All policies meet a schema. The test includes conformity tests regarding the frame concept.
- *Persistence Module:* Finally, the policies of this component are stored as RDF graphs and are retrievable for higher layers.

The services of the PMS are provided by means of the ”SPARQL 1.1 Graph Store HTTP Protocol” [26] as well as the ”SPARQL 1.1 Protocol” [27]. REST-based web services published via the Web Application Description Language (WADL) assure a consistent usage of all PMS services.

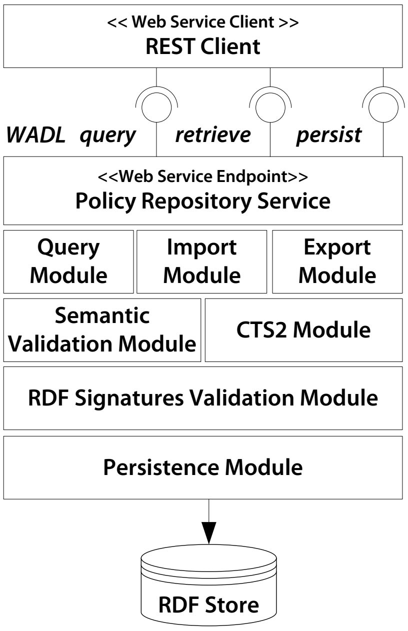

Figure 4. System architecture of the Policy Management System

Policy Conflict Handling

It is the primary interest to work out an application concept with simple and effective treatment strategies. That is why we consider the static analysis of policy conflicts during policy design rather than dynamic conflict resolution techniques, e.g. based on priorities or heuristics. We aim at creating an integrated database for later queries in contrast to solely evaluate a policy decision. In addition, the complexity of code systems in an ABAC-oriented policy scheme can only be managed practically with policies of the same type.

Therefore, the PMS is intended to be integrated into a workflow that implements the principle of ”security is a process”. This allows for a mitigation or ideally a correction of faulty states by a policy planner. A preventive treatment of policy conflicts by means of an iterative correction of a policy is ensured by recognizing these conflicts of the policy management system. The PMS’ persist module returns potential conflicts with other policies and makes sure that only valid policies are registered.

6 Evaluation

To evaluate the practical utility of the policy management system (and of course the model including the policy conflicts), it will be applied in a technical experiment (cf. DSR evaluation types proposed by Peffers et al. [8]). This means that patient privacy policies are tested with *Elektronische FallAkte* (EFA), which is a special kind of an electronic health record. The EFA principle is specified as a distributed treatment documentation that provides a standardized structured and integrated view of a patient’s medical data. Distributed means that different treating physicians view and update that data. The basic idea behind the EFA initiative launched by German hospitals and clinics in 2006 is to share a treatment case including the patient’s billing and treatment data with treatment facilities of a region. Thus, general practitioners as well as physicians of a hospital are able to access electronic documents relevant to treatment [28].

A consent is – in addition to the data protection requirement of obtaining permission – a basic construct, in order to open an EFA instance for a patient. The currently profiled patient privacy policy of the EFA (the so-called Enhanced Patient Privacy Consent – EPPC) consists of an HL7 Clinical Document Architecture (CDA) document [29] which in turn includes a computable policy (the so-called EPPC Patient Privacy Policy encoded with XACML). In this way, important metadata for a treatment case (CDA) is strictly separated from the actual authorization list (XACML) [30] (cf. Figure 5).

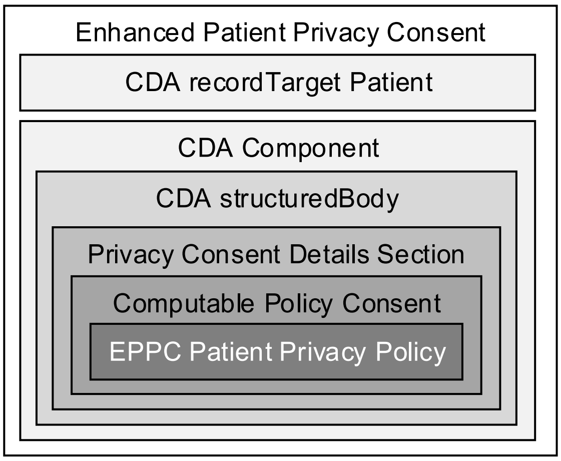
Enhanced Patient Privacy Consent
CDA recordTarget Patient
CDA Component
CDA structuredBody
Privacy Consent Details Section
Computable Policy Consent
EPPC Patient Privacy Policy

Figure 5. Structure of an Enhanced Patient Privacy Consent document [30]

6.1 IHE Connectathon

Several manufacturers of health-IT systems have agreed to test their current EFA implementations on the IHE Europe Connectathon in Bochum, Germany [31]. Three manufacturers tested the consent option with EPPC additionally to the EFA conformance tests. The five-day event, conducted annually by IHE, aims to interoperate with health information systems and issue certificates of conformity to IHE profiles. Compliance with the current specification for the EFA (almost an implementation profile based on different IHE profiles of the ”IT Infrastructure Domain”) is tested with regard to behavior and function.

A test case is carried out as follows: Two test partners have to find each other and configure their systems as a client or server system in a defined IP network. Depending on how a test case dictates, one system sends a SOAP message [32] to the other. All messages are checked by the so-called test monitors with regard to the specifications of IHE and are stored for verification purposes. This also means that the client system as well as the server system must be able to send and process specification-compliant messages in order to pass a test case. For the duration of the event, the systems can also be modified, for example, to repeat a failed test under other conditions.

6.2 Setup and Execution

All defined ontologies of our policy model are registered with the PMS’ validation module. Furthermore, the following states are pre-configured in the prototype:

- An attending physician can collect, process and use identification data as well as medical data within the scope of a medical admission of patient.
- An attending physician can access the application ”FallAkte”.
- The ”FallAkte” application uses the ”IHE XDS.b” application system to register and distribute diagnostic reports and images.
- Policy attribute relationships as they are defined in Section 4.3 are registered with the PMS.

In order to express the given configuration reference policies must be defined. The following Information Access Policy is registered as a policy instance within the PMS:

```
@prefix : <urn:policy:signatures:informationaccess#> .
@prefix prop: <urn:policy:signatures:properties#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://hospital.org/policy/efa-iapo>
  a :InformationAccessPolicy ;
  prop:note ”Information Access Policy for ’Elektronische FallAkte’”^^xsd:string ;
  prop:activated ”true”^^xsd:boolean ;
  prop:rule [
      a :Rule ;
      prop:action [
          prop:negationIndicator ”false”^^xsd:boolean ;
          a :DataActivity ;
          prop:code ”COLLECT”^^xsd:string ;
      ] ;
      prop:action [
          prop:negationIndicator ”false”^^xsd:boolean ;
          a :DataActivity ;
          prop:code ”USE”^^xsd:string ;
      ] ;
      prop:action [
          prop:negationIndicator ”false”^^xsd:boolean ;
          a :DataActivity ;
          prop:code ”PROCESS”^^xsd:string ;
      ] ;
      prop:domain [
          a :ProcessingContext ;
          prop:code ”ProfessionalTreatment”^^xsd:string ;
      ] ;
      prop:target [
          a :RecordType ;
          prop:code ”IdentificationData”^^xsd:string ;
      ] ;
      prop:target [
          a :RecordType ;
          prop:code ”MedicalData”^^xsd:string ;
      ]
  ] ;
  prop:subject [
     a :StructuralRole ;
     prop:code ”AttendingPhysician”^^xsd:string ;
  ] .
```

Furthermore, the following Resource Behavior Policy is registered as well:

```
@prefix : <urn:policy:signatures:resourcebehavior#> .
@prefix prop: <urn:policy:signatures:properties#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://hospital.org/policy/efa-rbpo>
  a :ResourceBehaviorPolicy ;
  prop:note ”Resource Behavior Policy for ’Elektronische FallAkte’”^^xsd:string ;
  prop:activated ”true”^^xsd:boolean ;
  prop:rule [
      a :Rule ;
      :informationAccessPolicy <http://hospital.org/policy/efa-iapo> ;
      prop:action [
          prop:negationIndicator ”false”^^xsd:boolean ;
          a :AccessType ;
          prop:code ”READ”^^xsd:string ;
      ] ;
      prop:target [
          a :Application ;
          prop:code ”FallAkte”^^xsd:string ;
          :implementedBy [
              a :ApplicationSystem ;
              prop:code ”IHE-XDS.b”^^xsd:string ;
          ] ;
      ] ;
  ] .
```

An EFA conformance test is passed as soon as a manufacturer successfully demonstrated a test case with at least three manufacturers. The relevant EFA test case is the one for opening a case record for a patient. A new record is opened by creating an IHE XDS[2] Folder and putting a patient consent (i.e. an EPPC document) to this folder at an EFA provider implementation which is grouped with the PMS. The following test steps must be carried out [33]:

1. The EFA client sends an *initializeEFA* message (SOAP message) to the EFA provider. A Schematron check tests the EFA compliance of the message.
2. When a successful check is performed, a further *initializeEFA* message with the same content data is sent to the EFA provider to test for an incorrect override of the EFA instance. A corresponding failure message should contain a policy violation error code.

The verification of an EPPC document includes, on the server side, the separation into a Resource Access Policy and Access Control Policy. After the separation, both artifacts are transferred into the policy model by means of the Extensible Stylesheet Language (XSL) [34] which then allows various checks to identify policy clashes (cf. Section 4.2 and Section 4.3). Finally, both policies are tested against reference policies.

6.3 Test Results

When the EPPC documents were tested by the PMS, it turned out that the activation of reference policies failed. This means that the test rules could not be applied because the Resource Access Policy and Access Control Policy of an EPPC document did not match with a suitable Resource Behavior Policy and Information Access Policy. The policy model required a modification in order to cope with structured data types for expressing subjects and objects as they are used in an EPPC document.

Objects are defined by an EFA purpose. The construct of the purpose binds a medical case to a patient – that is, all data that is available in a treatment context. In the case of EFA, this is implemented by so-called folder codes. A folder code, e.g. a coded diagnosis, describes an EFA instance in more detail and restricts the processing of a case record according to the intended purpose-of-use and need-to-know principle. If a patient agrees to the treatment, the agreed purpose-of-use must refer to one or more Access Control Policies. Finally, the *ihe:FolderCode* must be defined as a subtype of the purpose of an Access Control Policy in order to establish the necessary linkage to our policy model. The following RDF graph shows an example of structured data types used for subjects and objects:

```
@prefix acp: <urn:policy:signatures:accesscontrol#> .
@prefix hl7: <urn:hl7-org:v3#> .
@prefix iap: <urn:policy:signatures:informationaccess#> .
@prefix ihe: <urn:ihe:iti:xds-b:2007#> .
@prefix prop: <urn:policy:signatures:properties#> .
prop:subject [
   a ihe:HealthCareProfessional , iap:DataProcessor ;
   ihe:identifier [
      a hl7:InstanceIdentifier ;
      hl7:root "1.3.6.1.4.1.21367.2005.3.7" ;
      hl7:extension "6578946"
   ] ;
] ;
acp:object [
   a ihe:FolderCode ;
   ihe:code [
      a hl7:CodedValue ;
      hl7:code "K70.0" ;
      hl7:codeSystem "1.2.276.0.76.5.311"
   ]
] .
```

[2] XDS or more precisely XDS.b is an XML-based interoperability profile for sharing diagnostic reports and images defined by IHE.

After the necessary corrections re-tests revealed that all EPPC documents were semantically valid, i.e. no policy conflicts were present and the opening of a case record can be achieved.

The main limitation of the evaluation is that not all types of policy conflicts could be tested. Testing for comprehensive policy conflicts requires a certain number of policies that must be maintained in the PMS. However, the evaluation shows that security policies from practice can be analyzed by means of our policy model and policy conflict definitions.

7 Concluding Remarks

This work aims at providing contributions to a comprehensive information security management in the hospital by addressing policy conflicts when accessing protected health information. Security policies in a hospital can capture access demands from treating physicians and assistants. The recognition of policy clashes can make sure that information security constraints are brought to the access control level. The new approach presented in this work describes policies using terminology services. RDFS constructs allow for the definition and later querying of specific policy conflicts, such as domain conflicts. This complements an ABAC-based policy infrastructure.

An introduced policy management system in this work shows a comprehensive integration of the developed policy model and can detect policy clashes. Thus, it provides an advantageous verification tool for policy authors.

An exemplary patient privacy consent that inherently features a multi-dimensional policy was used for an evaluation of the policy management system. It presents how this policy type can be validated against and integrated with access control policies. However, the policy management system was evaluated with just a small number of policies. To use it in operation requires the conduct of additional performance tests. Especially with a large number of policies, querying the policy net needs a robust graph handling.

Further investigation is needed for providing support for common analysis on access demands. The policy model perfectly fits to handle structured queries (e.g. why is someone allowed to print a lab report in the laboratory information system). A frontend with a selection of predefined SPARQL queries that can be applied on a candidate policy should be developed. This hides complexity of RDF syntax from policy authors. Besides, a domain-specific language might ease the description of new policies instead of using RDF languages not known to policy authors.

References

[1] R. v. Solms, “Information security management (2): guidelines to the management of information technology security (GMITS),” *Information Management & Computer Security*, vol. 6, no. 5, pp. 221–223, 1998. [Online]. Available: https://doi.org/10.1108/EUM0000000004542

[2] J. Caumanns, R. Kuhlisch, O. Pfaff, and O. Rode, “IHE IT infrastructure white paper: Access control,” IHE International, Tech. Rep., 2009, rev. 1.3. [Online]. Available: http://www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_WhitePaper_AccessControl_2009-09-28.pdf

[3] R. Kuhlisch and S. Bittins, “Aligning ABAC policies with information security policies using controlled vocabulary,” in *Open Identity Summit 2016*, ser. Lecture Notes in Informatics, D. Hühnlein, H. Roßnagel, C. H. Schunck, and M. Talamo, Eds. Gesellschaft für Informatik e.V. (GI), 2016, vol. 264, pp. 181–191.

[4] T. Moses, "eXtensible Access Control Markup Language (XACML) Version 2.0," OASIS, Tech. Rep. oasis-access_control-xacml-2.0-core-spec-os, Feb. 2005.

[5] M. Benantar, *Access Control Systems: Security, Identity Management and Trust Models.* Springer Science+Business Media, 2006.

[6] Conference of the Data Protection Commissioners of the Federation and the Federal Länder, "Orientierungshilfe Krankenhausinformationssysteme – 2. Fassung," Tech. Rep., Mar. 2014. [Online]. Available: https://www.datenschutz.rlp.de/fileadmin/lfdi/Dokumente/dsb_info_kis.pdf

[7] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

[8] K. Peffers, M. Rothenberger, T. Tuunanen, and R. Vaezi, "Design science research evaluation," in *Design Science Research in Information Systems. Advances in Theory and Practice: 7th International Conference, DESRIST 2012, Las Vegas, NV, USA, May 14-15, 2012. Proceedings*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7286, p. 398–410. [Online]. Available: https://doi.org/10.1007/978-3-642-29863-9_29

[9] R. Kuhlisch and K. Sandkuhl, "Policy conflict handling as a monitoring activity of hospital information systems," in *Business Information Systems Workshops, BIS 2013 International Workshops Poznań, Poland, June 2013 Revised Papers*, ser. Lecture Notes in Business Information Processing, W. Abramowicz, Ed. Springer Berlin Heidelberg, 2013, no. 160, pp. 89–99. [Online]. Available: https://doi.org/10.1007/978-3-642-41687-3_10

[10] R. Kuhlisch, "A description model for policy conflicts for managing access to health information," in *Proceedings, 6th International Workshop on Information Logistics, Knowledge Supply and Ontologies in Information Systems (ILOG)*, ser. CEUR Workshop Proceedings, B. Lantow, K. Sandkuhl, and U. Seigerroth, Eds., vol. 1028. M. Jeusfeld c/o Redaktion Sun SITE, Informatik V, RWTH Aachen, 2013, pp. 44–55. [Online]. Available: http://ceur-ws.org/Vol-1028/paper-05.pdf

[11] R. Kuhlisch and J. Caumanns, "Engineering a CTS2-based system for healthcare security policy conflict checking," in *Proceedings, 3rd International Conference on Horizons for Information Architecture, Security and Cloud Intelligent Technology (HIASCIT 2015)*, 2015.

[12] M. Davis, B. Blobel, J. Moehrke, R. Thoreson, S. Gonzales-Webb, I. Singureanu, and S. Versaggi, "Composite Security and Privacy Domain Analysis Model," Health Level Seven, Domain Analysis Model Report, May 2010. [Online]. Available: http://gforge.hl7.org/gf/download/frsrelease/655/7084/Security_DAM_v1_r2.pdf

[13] D. F. Ferraiolo and D. R. Kuhn, "Role-based access controls," in *Proceedings, 15th National Computer Security Conference*, 1992, pp. 554–563. [Online]. Available: http://csrc.nist.gov/rbac/rbacSTD-ACM.pdf

[14] L. Kagal, T. Finin, and A. Joshi, "A Policy Language for a Pervasive Computing Environment," in *Proceedings, 4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'03).* Washington, DC, USA: IEEE Computer Society, 2003, pp. 63–74. [Online]. Available: https://doi.org/10.1109/policy.2003.1206958

[15] A. Uszok, J. Bradshaw, R. Jeffers, N. Suri, P. Hayes, M. Breedy, L. Bunch, M. Johnson, S. Kulkarni, and J. Lott, "KAoS Policy and Domain Services: Toward a Description-Logic Approach to Policy Representation, Deconfliction, and Enforcement," in *Proceedings, 4th IEEE International Workshop on Policies for Distributed Systems and Networks, 2003. Proceedings (POLICY 2003).* IEEE, Jun. 2003, pp. 93–96. [Online]. Available: https://doi.org/10.1109/policy.2003.1206963

[16] T. Mouelhi, F. Fleurey, B. Baudry, and Y. Traon, “A Model-Based Framework for Security Policy Specification, Deployment and Testing,” in *Proceedings, 11th international conference on Model Driven Engineering Languages and Systems (MoDELS ’08).* Berlin, Heidelberg: Springer-Verlag, 2008, pp. 537–552. [Online]. Available: https://doi.org/10.1007/978-3-540-87875-9_38

[17] R. Kuhlisch and J. Caumanns, “Engineering a CTS2-based System for Healthcare Security Policy Conflict Checking,” in *Proceedings, 3rd International Conference on Horizons for Information Architecture, Security and Cloud Intelligent Technology (HIASCIT 2015)*, Sanremo, Italy, Jul. 2015.

[18] M. Minsky, “A Framework for Representing Knowledge,” in *The Psychology of Computer Vision*, P. H. Winston, Ed. McGraw-Hill, Jun. 1974, no. AIM-306, pp. 211–277. [Online]. Available: http://dspace.mit.edu/handle/1721.1/6089

[19] D. GaSevic, D. Djuric, and V. Devedzic, *Model Driven Engineering and Ontology Development*, 2nd ed. Springer, Berlin, Jun. 2009. [Online]. Available: https://doi.org/10.1007/978-3-642-00282-3

[20] A. Billig, “Utilizing Semantic Technologies for a CTS2 Store,” Jun. 2013, version 0.1. [Online]. Available: http://semantik.fokus.fraunhofer.de/WebCts2LE/main3/cts4omg.pdf

[21] Object Management Group, “Common Terminology Services 2, Version 1.2,” OMG, Tech. Rep. formal/2015-04-01, Apr. 2015. [Online]. Available: http://www.omg.org/spec/CTS2/1.2/

[22] D. Beckett, “RDF 1.1 XML syntax,” 2004, W3C Recommendation. [Online]. Available: http://www.w3.org/TR/2014/REC-rdf-syntax-grammar-20140225/

[23] D. Brickley and R. V. Guha, “RDF schema 1.1,” 2014, W3C Recommendation. [Online]. Available: http://www.w3.org/TR/2014/REC-rdf-schema-20140225/

[24] J. D. Moffett and M. S. Sloman, “Policy Conflict Analysis in Distributed System Management,” *Journal of Organizational Computing*, vol. 4, no. 1, pp. 1–22, 1994. [Online]. Available: https://doi.org/10.1080/10919399409540214

[25] S. Harris, A. Seaborne, and E. Prud’hommeaux, “SPARQL 1.1 Query Language,” W3C, Tech. Rep., Mar. 2013, w3C Recommendation. [Online]. Available: http://www.w3.org/TR/2013/REC-sparql11-query-20130321/

[26] C. Ogbuji, “SPARQL 1.1 Graph Store HTTP Protocol,” W3C, Tech. Rep., Mar. 2013, w3C Recommendation. [Online]. Available: http://www.w3.org/TR/2013/REC-sparql11-http-rdf-update-20130321/

[27] W3C SPARQL Working Group, “SPARQL 1.1 Overview,” W3C, Tech. Rep., Mar. 2013, w3C Recommendation. [Online]. Available: http://www.w3.org/TR/2013/REC-sparql11-overview-20130321/

[28] bvitg, IHE Deutschland, and EFA-Verein, “EFA Spezifikation v2.0,” Apr. 2016. [Online]. Available: https://wiki.hl7.de/index.php?title=cdaefa:EFA_Spezifikation_v2.0

[29] Health Level Seven International, “HL7 Standards Product Brief – CDA Release 2,” Health Level Seven International, Tech. Rep., 2017. [Online]. Available: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7

[30] J. Caumanns, T. Idris, and B. Kraufmann, “Enhanced Patient Privacy Consent for Germany (EPPC-G) Integration Profile,” Verein ”Elektronische FallAkte” and Bundesverband Gesundheits-IT and Integrating the Healthcare Enterprise, EFAv2.0 Supplement / IHE Cookbook Supplement Rev. 0.1, Apr. 2014, national Comment. [Online]. Available: http://wiki.hl7.de/images/EPPC-G_Draft_for_Comment_v04.pdf

[31] IHE Europe, “IHE Connectathon CAT 2016 Impressions,” Apr. 2016. [Online]. Available: http://ihe-service.net/connectathon/cat-2016

[32] N. Mitra and Y. Lafon, “SOAP Version 1.2 Part 0: Primer (Second Edition),” W3C, Tech. Rep., Apr. 2007, w3C Recommendation. [Online]. Available: http://www.w3.org/TR/2007/REC-soap12-part0-20070427/

[33] J. Caumanns, “EFA Projectathon 2016,” Apr. 2016. [Online]. Available: http://wiki.hl7.de/index.php?title=cdaefa:EFA_Projectathon_2016#Test_Case_2:_EFA_Initialization

[34] M. Kay, “XSL Transformations (XSLT) Version 2.0,” W3C, Tech. Rep., Jan. 2007, w3C Recommendation. [Online]. Available: http://www.w3.org/TR/2007/REC-xslt20-20070123/

Additional information about the article:

Author’s ORCID iD	Raik Kuhlisch – www.orcid.org/0000-0002-1719-1507
Article history	Received 4 January 2017 Last revision received 28 June 2017 Accepted 10 July 2017 Available online 31 July 2017
Article PII	S225599221700062X