# Access Control over the Medical Informatics Platform (MIP)

Despoina Trivela, Tassos Venetis, Giorgos Stoilos, and Vasilis Vassalos.

Department of Informatics, Athens University of Economics and Business,
Athens, Greece.

## 1 Role based access control using semantic technologies

Information sharing is a major challenge for many modern applications. The functionality of sophisticated and intelligent applications based on infrastractures such as data integration systems, grid computing, web services, relies on the exchange of information among autonomous systems. However, such systems must protect their resources and negotiate information sharing. Therefore, modern applications require to determine policies for secure sharing of resources. Access control systems have been proposed to satisfy the policy requirements of the various application domains. Research in the area has resulted in several access control models [1, 6] and languages [11].

Role-based access control (RBAC) [8] has been recognized as an efficient access control model [8] and has been approved as a standard by the American National Standards Insitute (ANSI) [1]. It restricts access to system resources w.r.t. to user roles that are granted specific permissions. Web applications, healthcare systems, workflow-management systems require separation of duty among users. RBAC model groups users into roles and assigns permissions according to their position within an organization.

An access control mechanism depends on the structure of an organization. For example, an organization in the context of an application in the healthcare domain is a set of hospitals and people (users of the application). Users can be grouped into roles according to several criteria (do they work in the hospital, who is their supervisor). Once the structure and the subunits of the organisation, as well as the roles of each application user are determined, the requirements concerning the management of information resources must be specified. In the last decade the use of ontologies to describe the complex structure of applications data has gained great importance. Towards this direction, ontologies can be used in the context of an access control mechanism to express the strucure of the organization, the role of each user, the relationship between people working in the organization. Several access control systems have been proposed that make use of ontologies [3, 10, 7, 5, 11]. Systems can also use existing vocabularies to define their domain specific ontology. For example, in healthcare there are several vocabularies already used to describe medical data, such as the JuBrain Atlas, the Allen Atlas that are used for brain modeling.

### 1.1 State of the art

An ontology used for applying access control was presented is the SecurOntology presented in [3]. It is an ontology oriented to web services. It describes concepts such as the *Resources* of the system, the *Owners* of the resources, the *Role* of an owner (administrator, etc.), the *Permissions* of an owner over a resource for *read, write* and *execution.* The ontology also contains the class *ResourceAndPermission* that relates a resource to a permission. Properties are defined in the ontology to establish the relations among the classes of the ontology.

In [3] the architecture of an access control system is described. It uses the SecurOntology expressed in OWL to describe the elements of the domain and rules expressed in SWRL to describe the access control policy. The ontology is translated into SWRL facts that are processed along with the access rules by the access control manager module. The ontology facts and policies are transformed to actual permissions on resources related to Web applications (such as content management systems, knowledge management systems, applications with financial data and sensitive data). The access control rule manager is a reasoning engine that applies the rules on the ontology instances to result to an access control desicion. Notice that the inference process is applied on data (owl triples translated into SWRL facts) and not on ontology axioms.

Another access control system for web services is described in [10]. The authors propose the use of semantic rules expressed in SWRL to protect data that can be accessed from a web-service endpoint. The information being protected is defined by an ontology expressed in OWL. The filtering of information occurs by taking into account axioms specificaly defined within the KB. The evaluation of the rules is performed by an OWL-DL reasoning engine that is extended to handle SWRL rules. A user can be granted either full, or limited, or no access depending on the rules.

In [7] an access control system is described that applies directly on rdf stores, that is the access control rules specify the actions (described by the systems ontology) that an agent can perform directly on the rdf graph. Other access control systems applied to rdf stores have been proposed [4] . . .

A complete security framework was presented in [9]. It proposes a solution to access control to xml documents. It involves an RBAC system that uses an OWL ontology to describe the domain information and access rules expressed in SWRL by using the vocabulary of the ontology. At runtime the xml files are mapped to the ontology and as a result each element and attribute of an xml file is represented by an instance of an OWL class or property. For example, consider the following ClassMapping:

<ClassMapping>
<xmlItem>PatientRecord/Patient/PatientInformation</xmlItem>
<owlItem>PatientInformation <owlItem>
< /ClassMapping>

where the xmlItem PatientRecord/Patient/PatientInformation is mapped to the owl class PatientInformation. Once the instance mapping is complete, an Semantic Access Control Query is formulated from every mapping object and send to the reasoning engine. The semantic access control queries are triples of the form (*individual1, property, individual2*) where *property* is the "hasAccessToInformation" OWLObject property of systems ontology. The results are stored (in xml items) and are used to remove elements and attributes from the xml file. Notice that in this approach access control is performed on xml files and that the reasoning process that performs the access control is performed on instance information and not on the general concepts of the ontology.

In [2] the relationship between the RBDA model and the OWL is studied. OWL can express the components of the RBAC model and its details and hence is a suitable language to describe access rules in the lines of RBAC. OWL has also been used to define policy languages such as Rei [5] and Kaos [11].

The authors propose two different approaches for modeling RBAC by using OWL. Both approaches support the variations of the RBDA model (flat, hierarchical, constrained, symmetric) [1]. They both use a basic RBAC ontology and each one uses an ontology that models a specific domain. The basic RBAC ontology describes the concept of an *Action* that has exactly one *Subject* and *Object.* Each action is either a *PermittedAction* or a *ProhibitedAction.* (example goes here). According to the first approach a role in terms of RBAC is an OWL-class. For example, a *User* is a subclass of the class *Role.* Role activation and seperation of duties among users are accomplished by using rules. According to the second approach the roles in terms of RBAC are expressed as values. For example, a "User" is an instance of class *Role.* In this case, inheritance of roles cannot be described by OWLsubclass axioms and is accomplished by rules.

In [5] the Rei policy language is described. Rei allows to specify the policy of an entity that is, what information an entity of an organization will make available to other entities. It includes three constructs (*policy objects*, *speech acts* that affect the policy objects and the *metapolicies* that prevent conflicts caused by policies). Rei is implemented in Prolog. It includes a domain independent ontology (basic ontology) and accepts domain dependent ontologies. The basic ontology includes concepts describing the policy objects, actions, and speech acts such as revocation, delegation, etc. The domain specific ontologies describe the entities of the system and include domain concepts such as person and files.

### 1.2 Role based access control in MIP

The Medical Informatics Platform operates over a data integration system collecting data from different hospitals. Each hospital defines its own policy that determines the resources that will be made available through MIP services. Different users of the platform may be assigned to different permissions over MIPs resources (data or services). We are interested in defining a centralized access control system that meets the platform requirements concerning availability of resources.

**Design of the access control mechanism** We have designed a mechanism that embodies the basic concepts of RBAC; users are assigned to roles, and each role is assigned to permissions. According to RBAC model a user can be assigned to many roles and a single permission can be assigned to more than one roles. Moreover, a user is able to exercise at the same time permissions corresponding to more than one roles. The RBAC model involves the following 5 basic elements [1]:

1. User. Users are human beings or machines, networks, agents etc.
2. Role. A role is a job assigned to a user that is associated with specific authority and responsibity.
3. Permission. Permissions are the approvals to a User to perform an operation.
4. Operation. An opertation involves the execution of some function by the user.
5. Object. An object is an entity that contains or receives information/ is a system resource (files, tables, rows within a database table).

We follow the approach proposed in [2] and use a basic ontology to describe the core elements of the RBAC model. This ontology includes the concept *Action* that can be either a *PermittedAction* or a *ProhibitedAction* capturing the notion of the Permission related to the Operation element of the core RBAC. *Action* is related to *Subject* and *Object* concepts via the properties *subject, object* respectively. A user in the context of RBAC is an instance of the class *Subject* or *Object*.

Next, we describe the domain specific ontology (MIP ontology) that describes the structure of the platform. Its concepts are subclasses of the classes occuring in the basic ontology. A role in the context of RBAC is captured by the concept *Role* in our ontology. *Role* is a subclass of the class *Subject* that occurs in the basic ontology. The MIP user groups (described in IP Web UI-User Guidelines) are represented in the ontology as subclasses of the *Role* concept. Therefore, we define concepts *Clinician, Researcher, Statistician, ScientificDev, PlatformDev, MedicalReseaarchWriters, GeneralPublic*. The services and data of MIP are subclasses of the *Object* class.

Once we have described the structure of the application and its users roles by an ontology we can determine the access control rules by using the vocabulary of the ontology.

*Example 1.* The following rules are used to describe that a Researcher can access the diagnostics table of the local database and that a Researcher can assign a general user with access restrictions.

$$
\begin{aligned}
Action(x) &\leftarrow PermittedAction(x)\\
PermittedAction(x) &\leftarrow PermittedAccessToDiagnostics(x)\\
PermittedAccessToDiagnostics(x) &\leftarrow AccessToDiagnostics(x) \wedge subject(x,y)\\
&\wedge Researcher(y)\\
PermittedAccessToDiagnostics(x) &\leftarrow AccessToDiagnostics(x) \wedge subject(x,y)\\
\wedge GeneralUser(y) \wedge &\, grantAccessToDiagnostics(z,y) \wedge Researcher(z)
\end{aligned}
$$

*Example 2.* The following rules describe that a biological signature of a disease (described by concept $BSignature(x)$) can be uploaded, deleted or updated by the platform developer.

$$\begin{aligned}
PermittedAction(x) &\leftarrow PermittedUploadBSignature(x)\\
PermittedAction(x) &\leftarrow PermittedUpdateBSignature(x)\\
PermittedAction(x) &\leftarrow PermittedDeleteBSignature(x)\\
PermittedUploadBSignature(x) &\leftarrow UploadBSignature(x)\wedge\\
&\wedge subject(x,y) \wedge PlatformDev(y)\\
PermittedUploadBSignature(x) &\leftarrow UdateBSignature(x)\wedge\\
&\wedge subject(x,y) \wedge PlatformDev(y)\\
PermittedUploadBSignature(x) &\leftarrow DeleteBSignature(x)\wedge\\
&\wedge subject(x,y) \wedge PlatformDev(y)
\end{aligned}$$

*Example 3.* The following rules are used to describe that only psychiatrists and neurologists users can obtain data related to brain scans.

$$\begin{aligned}
Role(x) &\leftarrow Clinician(x)\\
Clinician(x) &\leftarrow Neurologist(x)\\
Clinician(x) &\leftarrow Psychiatrist(x)\\
Action(x) &\leftarrow PermittedAction(x)\\
Action(x) &\leftarrow AccessToBrainScan(x)\\
PermittedAction(x) &\leftarrow PermittedAccessToBrainScan(x)\\
PermittedAccessToBrainScan(x) &\leftarrow AccessToBrainScan(x)\wedge\\
&\wedge subject(x,user) \wedge Neurologist(user)\\
PermittedAccessToBrainScan(x) &\leftarrow AccessToBrainScan(x)\wedge\\
&\wedge subject(x,user) \wedge Psychiatrist(user)
\end{aligned}$$

The domain specific ontology used for ac includes concepts that are related to resources that must be protected. It can be extended with any ontology that describes MIPs resources according to the system functionality and requirements. Access rules can be dynamically defined to satisfy the system's policy.

## References

1. David F. Ferraiolo, Ravi S. Sandhu, Serban I. Gavrila, D. Richard Kuhn, and Ramaswamy Chandramouli. Proposed NIST standard for role-based access control. *ACM Trans. Inf. Syst. Secur.*, 4(3):224–274, 2001.

2. Timothy W. Finin, Anupam Joshi, Lalana Kagal, Jianwei Niu, Ravi S. Sandhu, William H. Winsborough, and Bhavani M. Thuraisingham. R*OWL*bac: representing role based access control in *OWL*. In *13th ACM Symposium on Access Control Models and Technologies, SACMAT 2008, Estes Park, CO, USA, June 11-13, 2008, Proceedings*, pages 73–82, 2008.
3. Ángel García-Crespo, Juan Miguel Gómez Berbís, Ricardo Colomo Palacios, and Giner Alor-Hernández. Securontology: A semantic web access control framework. *Computer Standards & Interfaces*, 33(1):42–49, 2011.
4. Amit Jain and Csilla Farkas. Secure resource description framework: an access control model. In *Proceedings of the eleventh ACM symposium on Access control models and technologies*, pages 121–129. ACM, 2006.
5. Lalana Kagal, Timothy W. Finin, and Anupam Joshi. A policy language for a pervasive computing environment. In *4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2003), 4-6 June 2003, Lake Como, Italy*, page 63, 2003.
6. Ninghui Li and John C Mitchell. Datalog with constraints: A foundation for trust management languages. In *International Symposium on Practical Aspects of Declarative Languages*, pages 58–73. Springer, 2003.
7. Pavan Reddivari, Tim Finin, and Anupam Joshi. Policy-based access control for an rdf store.
8. Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.
9. Brian Shields and Owen Molloy. Securing systems intelligently: The logical approa. In *Information Systems Development, Challenges in Practice, Theory, and Education Volume 2 [Proceedings of ISD 2007, National University of Ireland, Galway, Ireland. August 29-31, 2007].*, pages 753–766, 2007.
10. Brian Shields, Owen Molloy, Gerard Lyons, and Jim Duggan. Using semantic rules to determine access control for web services. In *Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23-26, 2006*, pages 913–914, 2006.
11. Gianluca Tonti, Jeffrey M. Bradshaw, Renia Jeffers, Rebecca Montanari, Niranjan Suri, and Andrzej Uszok. Semantic web languages for policy representation and reasoning: A comparison of kaos, rei, and ponder. In *The Semantic Web - ISWC 2003, Second International Semantic Web Conference, Sanibel Island, FL, USA, October 20-23, 2003, Proceedings*, pages 419–437, 2003.