# Representing Access Control Policies using Semantic Web

Project plan

Steven van Beek (10292527)
*Supervisor:* Milen G. Kebede

03-04-2020

# 1 Context

In modern healthcare there is often a need for organisations to share information about patients in order to provide treatment ~~or for use~~ in medical research. This medical information is considered sensitive and government regulations exist to prevent unnecessary dissemination.

In order to comply with these regulations organisations create access control policies to restrict access to sensitive information. Moreover, both policies and technical implementations of these policies differ across organisations. As a consequence, it is hard to reason whether organisations are compliant with government regulations and access is properly restricted. This in turn could inhibit sharing of important medical information across organisations.

Therefore, it could be helpful to look in to improving means for healthcare organisations to analyse and verify access control policies.

# 2 Literature

## 2.1 Access Control Models

Access control is typically represented as a subject, object, and a permission. The permission determines whether a subject is authorised to access an object. A subject can be seen as a user or any process or program acting on behalf of the user. An object is specified as any item in connection to information technology to which access needs to be restricted.

These relations are represented in Access Control Models (ACM). Access Control Policies are typically built on an Access Control Model. The most relevant Access Control Models today are briefly discussed.

### 2.1.1 Discretionary access control

Discretionary Access Control (DAC) is an access control model commonly used in modern operating systems such as Microsoft Windows and UNIX-based systems. The defining feature of DAC is that the owner (a subject) of an object has discretion over what other subjects have what permissions on that object.

The main advantages of using DAC are the flexibility and fine-grained control the model offers. The model allows the owner to set the least required privilege for each object. DAC is also considered intuitive to use.

One of the drawbacks of DAC is that maintaining and verifying policies is hard to accomplish due to the fact that users can alter permissions. Another downside is that users can lose control over who has access to information when another users with permission to read said information copies it. [1]

### 2.1.2 Mandatory access control

Mandatory Acess Control (MAC) is an access control model where permissions for subjects to access information is administered centrally. MAC is popular

with organisations requiring high levels of confidentiality such as military and intelligence agencies.

The model most frequently used is called the *Bell-LaPadula Confidentiality Model*. This model assigns classifications to subjects and objects and applies two principles to ensure information can not flow to a lower classification it currently has. Firstly, a subject is not allowed to access an object with a higher classification. Secondly, a subject is not allowed to write to a lower classification. Furthermore, limits are typically placed on changing classifications during operations of the system.

MAC is considered a good a model to use when the need for confidentiality or risk of attack is high. However, MAC can only be applied to a part of a system requiring the rest to be considered *trusted components*. Furthermore, the strict rules that MAC enforces often hamper productivity. [1]

### 2.1.3 Role based access control

In Role Based Access Control (RBAC) permissions are assigned to roles as opposed to a subject directly. This allows administrators bundle a set of related permissions into a role and assign the role to multiple users. RBAC is considered a mix of DAC and MAC where fine-grained access control is possible but the ability of users to alter permissions is removed. Several variations of RBAC exist. Notably, hierarchical RBAC where roles are allowed to inherit from other roles creating an hierarchy. The main benefit of RBAC is ease of administration and verification of security policies.[1][3]

### 2.1.4 Attribute based access control

As the name suggests Attribute Based Access Control (ABAC) allows for attributes to be used in determining authorisation. These attributes can be related to the subject, object, or the environment. The environment contains attributes unrelated to the subject or object. For example, the time of day. An advantage of using ABAC are that information can more easily shared across organisations. Moreover, it allows for deciding access right based on a variety of factors other than the subjects identity and roles. This enables more fine-grained control access control. The main downside of ABAC is the difficulty administering and verifying access, especially when the number of attributes and the complexity of the rules increase. [4]

## 2.2 Semantic web

Semantic web used to describe a set of standards created by the W3C able to order information as web of knowledge. Presenting information in a semantic web allows computers to more easily interpret and reason about it. One standard of particular interest is the Web Ontology Language (OWL). This language can be used to declaratively create an ontology using classes, properties, instances

of classes, and constraints. Moreover, this ontology can then be used to reason its about properties and relations.[2][3]

### 2.3 Representing ACM in OWL

Finin *et al* showed multiple methods to describe a simple RBAC and ABAC based Access Control Policy (ACP) using OWL. This allowed for reasoning technology to be used to query for permissions. Furthermore, after adding some constraints the paper show the ability to perform security analysis on the ACP defined in OWL. The constraints mentioned refer to the inability to prove properties of the policy when rules are added or removed.[3]

### 2.4 Existing representation of Access Control Policies in XACML

Extensible Access Control Markup Language (XACML) is an XML-based declarative language used to express Access Control Policies. The language focuses on the ~~Attribute Based Access Control~~ model but is also applicable to ~~Role Based Access Control~~ based policies. XACML can be used to combine rules in policies or policy sets which state whether a subject should given permission given attributes of the subject, object, and environment. [5]

## 3 Research question

Considering the current situation in healthcare information sharing and the state-of-the-art leads me to pose the following research question:

*Can semantic web be used to represent Access Control Policies described in XACML?*

In order to fully answer this question the following sub-questions have been formulated:

1. *How do Access Control Models in common use today compare?*
2. *How can Web Ontology Language be used to accurately represent a Access Control Policy specified in XACML?*
3. *What are limitations and incompatibilities translating XACML to an OWL ontology?*

In addition, the following items will be delivered in support of the thesis:

1. An OWL ontology representing an Access Control Policy specified in XACML.
2. The source code of a conversion tool to convert an XACML Access Control Policy to an OWL ontology.

# 4 Methods

## 4.1 Defining a policy and researching access control

In order to answer this question a (hypothetical) policy representative to healthcare will need to be defined. In light of this, research will need to be done on what a representative ontology would look like. Furthermore, a literature study will need to be done in order to adequately choose an ACM.

## 4.2 Describing a policy in OWL

An ontology should be created based on this policy using methods similar to Finin *et al.* [3]. Research will need to be done on how the to implement this ontology using OWL.

## 4.3 Converting the policy

After the ontology is finished a tool should be implemented that is able to convert. In case conversion is not, or only partly, possible the causes of failure should be noted as results.

# 5 Planning

## 5.1 Timeline

The tentative planning is represented in a Gantt chart, which is included in this document. In this Gantt chart the intensity of the work is indicated by the brightness of the colors.

## 5.2 Failure of delivering an adequate ontology

The most critical deliverable of this thesis project is the ontology. If it proves impossible to create an adequate ontology within the given time frame one of the following options should be taken:

1. A simpler Access Control Model could be chosen and new policy definition should be created, limiting time spent on the converter tool.

2. The research could take a theoretical turn looking into why the creation of the ontology failed.

| | Week 14 | Week 15 | Week 16 | Week 17 | Week 18 | Week 19 |
|---|---|---|---|---|---|---|
| **Deadlines** | Project plan | | | | Draft (May 1) | |
| **Research** | | | | | | |
| Access control | | | | | | |
| XACML | | | | | | |
| OWL | | | | | | |
| Converter | | | | | | |
| **Implementation** | | | | | | |
| Ontology | | | | | | |
| Converter tool | | | | | | |
| **Writing** | | | | | | |
| Project plan | | | | | | |
| Literature | | | | | | |
| OWL | | | | | | |
| Converter | | | | | | |
| Results | | | | | | |
| Improvements | | | | | | |
| | | | | | | |

| Week 20 | Week 21 | Week 22 | Week 23 | Week 24 | Week 25 |
|---|---|---|---|---|---|
| | | | First final | | Final + Delta document |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

# References


[1] R. Ausanka-Crues. Methods for access control: advances and limitations. *Harvey Mudd College*, 301:20, 2001.

[2] S. Bechhofer, F. Van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, L. A. Stein, et al. Owl web ontology language reference. *W3C recommendation*, 10(02), 2004.

[3] T. Finin, A. Joshi, L. Kagal, J. Niu, R. Sandhu, W. Winsborough, and B. Thuraisingham. R owl bac: representing role based access control in owl. In *Proceedings of the 13th ACM symposium on Access control models and technologies*, pages 73–82, 2008.

[4] V. C. Hu, D. Ferraiolo, R. Kuhn, A. R. Friedman, A. J. Lang, M. M. Cogdell, A. Schnitzer, K. Sandlin, R. Miller, K. Scarfone, et al. Guide to attribute based access control (abac) definition and considerations (draft). *NIST special publication*, 800(162), 2013.

[5] O. Standard. extensible access control markup language (xacml) version 3.0, 2005.