SEMANTICS-BASED ACCESS CONTROL
Ontologies and Feasibility Study of Policy Enforcement Function

Anton Naumenko
Department of Mathematical Information Technology, University of Jyväskylä, P.O.Box 35, FIN-40014, Finland

Keywords: Access Control, Policy Enforcement Mechanism, Semantic Web, Ontology.

Abstract: The current Web evolves to the Web 2.0 that is an intermediate step towards Semantic Web. Conventional security measures fall short to serve both, emerging technologies and innovative web-based information systems. The paper presents our research and development results towards adoption Semantic Web standards for the creation of unified view on the access control area that enables flexible, collaborative and distributed management of access control based on semantic relations amongst relating concepts. The integration of Semantic Web and access control disciplines leads to the elaboration of new more intelligent, flexible and reusable access control mechanisms and tools. The paper has practical orientation, evaluating research results and ideas with the development and testing of the prototype for the enforcement of access control policies based on the ontologies.

1 INTRODUCTION

The current Web evolves to the Web 2.0, which is an intermediate step towards Semantic Web (Berners-Lee, Hendler, and Lassila, 2001), by adding new unique advanced features (O'Reilly, 2005). Ubiquitous and autonomic computing, RFID technologies, and ambient intelligence ultimately leads to the "Internet of things". Web-based information systems become more complex, dynamic, heterogeneous, pervasive, nomadic, and open. Conventional security measures fall short to serve both, emerging Internet technologies and innovative web-based information systems. This slows down or even blocks the adoption of innovative Internet and Web technologies.

We present our research and development results towards Semantics-Based Access Control (SBAC). SBAC aims at the adoption of Semantic Web standards for the creation of unified view on the access control area that enables flexible, collaborative and distributed management of access control based on semantic relations amongst relating concepts. SBAC research and development targets are mathematical models (Naumenko 2006), ontologies (Gruber, 1993), specification of functionality for enforcement and administrative functions, algorithms, abstract designs (Naumenko 2006, Naumenko and Luostarinen 2006), reference implementations, concrete designs for different domains with different ICTs (Naumenko et al 2005, Naumenko and Luostarinen 2006, Naumenko et al 2007). SBAC is to provide means for the management of access control on the abstract level, in order to allow flexible ontology-based policy management for open and dynamic environments. This paper addresses the use of ontologies and evaluates our research results with the development and testing of the prototype for the enforcement of access control policies based on ontologies.

The remainder is organised as follows. Related work is presented in section 2. Section 3 addresses ontologies in SBAC. Section 4 describes the study of feasibility of the SBAC enforcement mechanism. Section 5 provides conclusions.

2 RELATED WORK

The presented in this paper research lies on the intersection of Semantic Web and access control research areas. There are number of ongoing efforts to apply the Semantic Web standards to different aspects of access control.

Yagüe et al initially introduced Semantic Access Control (SAC) model (2005) and constantly publish results of their research on applying SAC (2003) in different environments. SAC uses XML (Yergeau et al., 2004) inheriting limitations of XML-based efforts (see below).

Naumenko A. (2007).
SEMANTICS-BASED ACCESS CONTROL - Ontologies and Feasibility Study of Policy Enforcement Function.
In *Proceedings of the Third International Conference on Web Information Systems and Technologies - Internet Technology*, pages 150-155
DOI: 10.5220/0001265601500155

The Concept-level Access Control (Qin and Atluri, 2003) introduces the model based on 4-tuple (object, operation, positive or negative sign, subject) for the specification of authorizations over Semantic Web data.

Rei is the rule-based policy language represented using RDF. This language originally was oriented to specify policy rules for individual subjects, targets and actions. However, it permits specification of policies based on roles, groups and entities despite the fact, that notions for roles, groups and entities have not been specified in the basic Rei ontology (Tonti et al., 2003).

KAoS is an approach to the ontology-based policy representation language. It is based on KAoS Policy Ontology (KPO) that uses OWL (Tonti et al., 2003). KAoS policies authorize actions that restrict subjects and objects of access further in their annotations. Thus the KAoS overstates the importance of actions comparing to subjects and objects. Policies may target individual concepts, classes, groups, etc.

Finally, there are number of ongoing industrial efforts to produce access control languages and standards based on XML, like Extensible Access Control Markup Language (Moses, 2005), Web Services Security (Nadalin et al., 2006), Extensible Rights Markup Language (Wang et al., 2002), etc. XML-based solutions intersect in ideas and concepts with ontology-based approaches. However, they do not concentrate on semantic features and thus do not fully gain benefits of Semantic Web. The main limitation of these efforts is that knowledge representation models standardized as part of the Semantic Web activity is much more generic and expressive than the representation based on tailored XML schemas.

3 SEMANTICS-BASED ACCESS CONTROL ONTOLOGIES

The main research proposal is to use ontologies instead of mathematical access control and domain models. The SBAC ontologies consolidate and formally specify knowledge of the access control domain in machine-interpretable form. This means that SBAC ontologies mainly represent and organize knowledge that was already formalized in different existing access control models. The SBAC ontologies formally serialize the model-theoretic semantics of SBAC (Naumenko 2006) which uses the model-theoretic semantics of OWL (Patel-Schneider, 2004).

For the specification of SBAC ontologies we use the abstract syntax of OWL (McGuinness and Harmelen, 2004). Purpose of abstract syntax is informal specification of ontologies that facilitates analysis of concepts and relations.

A regular OWL ontology consists of annotations, axioms, and facts. Annotations carry information about authorship, versioning and other data associated with an ontology and concepts. Facts and axioms provide information about classes, individuals and properties that form main content of an ontology. An ontology can have name that is intended to be the address where it can be found, although this is out of formal semantics.

Semantics-based security (SBS) ontology is a stub of upper ontology. The SBS ontology defines three classes and three individual-valued properties with explicit definition of their names (note: OWL allows defining anonymous concepts). Specification of classes and properties consists of axioms that associate concepts' identifiers with the specification of their characteristics, for example that sbs:subject, sbs:predicate and sbs:object properties have sbs:SecurityStatement class as their domains. The class of security statements and three relations define a generic structure for specification of statements related to security e.g. privileges, prohibitions, obligations for access control, trace statements for logging and audit, reputation statements and trust agreement statements for trust management, and other. The scope of this paper encompasses semantics of access control statements. The main feature of the semantics of access control statements and the whole SBAC is that above mentioned security-related statements are specified between classes instead of individuals.

```
Ontology(sbs:ontology
Class(sbs:SecurityStatement)
ObjectProperty(sbs:subject
domain(sbs:SecurityStatement))
ObjectProperty(sbs:predicate
domain(sbs:SecurityStatement))
ObjectProperty(sbs:object
domain(sbs:SecurityStatement)))
```

An ontology property owl:imports gives the extra effect of importing the contents of target ontology into the current ontology (Patel-Schneider, 2004). The SBAC ontology imports the SBS ontology in order to specialize the security statement and three relations. The introduced class for access control statements is a subclass of security statements. Subject, operation and object relations of access control statements are subproperties of corresponding relations of the SBS ontology. The SBAC ontology also defines restrictions on these relations that their values must be classes of resources and operations, respectively. For this

purpose there are two sub classes of the owl:Class concept that denote the class of resources and class of operations. A resource is an entity of physical or digital world that is a subject or an object of access. Definition of the resource as a set for subjects and objects gives more flexibility in access control rights specification because it is hard to separate resources on passive and active in environments where artificial resources play active roles and their relations to human users are weak or are not present. Individual operations could be actions, transactions, access modes, etc. Finally, there is an axiom defining relation of precedence between specialisations of access control statements, like privileges, prohibitions, etc (see description below).

```
Ontology(sbac:ontology
Annotation(owl:imports sbs:ontology)
Class(sbac:ClassOfResources partial
owl:Class)
Class(sbac:ClassOfOperations partial
owl:Class)
Class(sbac:AccessControlStatement
partial sbs:SecurityStatement
restriction(sbac:subject
allValuesFrom(sbac:ClassOfResources)
)
restriction(sbac:operation
allValuesFrom(sbac:ClassOfOperations
))
restriction(sbac:object
allValuesFrom(sbac:ClassOfResources)
)))
ObjectProperty(sbac:subject
super(sbs:subject)
domain(sbac:AccessControlStatement)
range(sbac:ClassOfResources))
ObjectProperty(sbac:operation
super(sbs:predicate)
domain(sbac:AccessControlStatement)
range(sbac:ClassOfOperations))
ObjectProperty(sbac:object
super(sbs:object)
domain(sbac:AccessControlStatement)
range(sbac:ClassOfResources))
ObjectProperty(sbac:precedes))
```

The SBAC privilege and prohibition ontologies import the SBAC ontology in order to extend it with class axioms that define the class of privilege statements or the class of prohibition statements, respectively. These classes are subclasses of the abstract class of access control statements. The individual privileges and prohibitions are positive and negative authorizations.

```
Ontology(sbacpriv:ontology
Annotation(owl:imports
sbac:ontology)
Class(sbacpriv:Privilege partial
sbac:AccessControlStatement))

Ontology(sbacproh:ontology
Annotation(owl:imports
sbac:ontology)
Class(sbacproh:Prohibition partial
sbac:AccessControlStatement))
```

A privilege is an authorization of resources to access other resources using some operations. A decision of access granting or prohibiting depends on classification of subjects, operations and objects. The decision algorithm evaluates types of subjects, operations and objects taking into account partial order of classes.

Support of only positive authorizations in the form of privileges guaranties a conflicts free specification of access control policies. However, even in this case, the model has an implicit prohibition that everything is prohibited unless it is privileged. Introducing means for the specification of prohibitions in SBAC policies enhances expressivity of the policy language i.e. to make negative authorizations explicit.

It is evident that policies with privileges and prohibitions are not free from conflicts in an arbitrary case (Naumenko 2006). These policies require mechanisms to resolve conflicts and ambiguity for the guarantied decidability. Following the fundamental principle of access control for ensuring confidentiality, prohibitions always precede privileges. For example, block lists in mobile phones prohibit accepting calls from given phone numbers while there is a general implicit privilege to accept calls from everybody. Note, for this example prohibitions are mostly used to specify policies in the form of block lists. Although in the most cases policies will follow the fundamental principle, there is a need to specify the precedence between privileges and prohibitions to facilitate at least the explicit specification of the fundamental prohibiting principle with the further precedence of privileges.

Figure 1 illustrates the concepts of SBAC ontologies, importing mechanism amongst the ontologies and possible policies commitments to the different SBAC features introduced in the paper. Ontologies for the policies A and D are able to define only privilege and only prohibition statements respectively. The policies B and C may contain both types of statements. Privileges have precedence over prohibitions in the policy B and prohibitions precede privileges in the policy C.

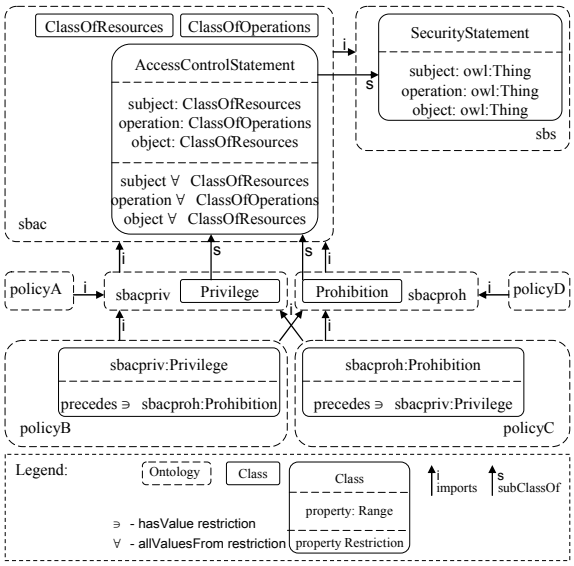

Figure 1: The SBAC ontologies.

SBAC interprets the facts, axioms and ontologies as defined by the OWL direct model-theoretic semantics. Notable and important interpretations of OWL for SBAC are provided briefly below. The OWL provides a possibility to specify classes using descriptions. Descriptions are axioms and they include class identifiers, restrictions and boolean combinations of other descriptions. Boolean combinations are union, intersection, and complement. Restrictions are placed on properties and called also facets. Descriptions allow flexible specification of access control policies for further inferring access control statements applicable to individual resources and operations based on their taxonomic and faceted classifications Another useful OWL feature for organizing access control statements is specification of an enumerated class by the explicit specification of all individual members.

Interpretation of ontologies is the key issue for evolution, consistency, reasoning and organising SBAC policies and domain knowledge in different ontologies separately. That is needed for flexible and joint further use with the high conceptual granularity. Annotation and ontology properties help to record a history of evolution of the SBAC and domain ontologies, policies, trust agreements, etc. The OWL standard (Patel-Schneider, 2004) defines conditions when an abstract OWL interpretation satisfies an OWL ontology. The definitions of when and how a collection of ontologies and axioms and facts is consistent and entails an ontology or axiom or fact provide background for reasoning and maintaining integrity of the SBAC data.

4 FEASIBILITY STUDY

The use of Semantic Web standards ensures automated reasoning over ontology-based access control policies. This also ensures the possibility to reuse existing Semantic Web tools and applications. The prototyping was conducted with the main purpose to test performance of the SBAC enforcement mechanism and to gather information for the feasibility study.

4.1 Development and Testing Environment

The development environment consists of several interrelated elements (figure 2). Java 2 standard edition development kit version 1.5 (java.sun.com/j2se/1.5.0/) is a programming language and platform that was chosen for the prototyping of research ideas. Jena (jena.sourceforge.net/) is a semantic web framework for java developed within the HP Labs Semantic Web Programme (www.hpl.hp.com/semweb/). ARQ (jena.sourceforge.net/ARQ/) is a SPARQL processor for Jena. SPARQL is a query language for the RDF developed by W3C (Prud'hommeaux and Seaborne, 2006). Eclipse (www.eclipse.org) is an open source community that produces extensible with huge amount of plugins integrated development environment (IDE). The last version of the IDE is 3.2. The Eclipse Test & Performance Tools Platform (TPTP, www.eclipse.org/tptp/) project consists of four subproject one of which provides tools for tracing and profiling java applications for further analysis of performance. The protégé (www.protege.stanford.edu) is the most appropriate tool to create defined SBAC ontologies in the RDF/XML exchange syntax of OWL. The protégé is an open source and free ontology editor with the number of plugins for editing (Protege-OWL) and visualizing (Ontoviz, OWL Viz) OWL ontologies. Web server is a container for developed in the protégé SBAC, domain and policy ontologies that are accessible by the prototype through HTTP.

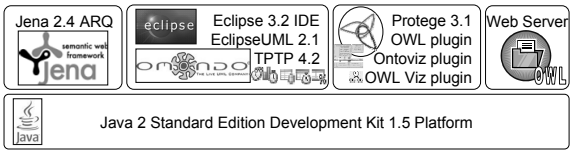

Figure 2: The development environment.

4.2 Testing the Prototype

The test application firstly creates a subject, a protected object and a guard. Then, it initiates a request from the subject to the guard, which evaluates the request. Basically, there are two processes with distinct characteristics and impacts on the overall performance of the guard. The first process starts up the guard. It creates and initializes all internal components. The performance of this process is crucial for the fast restarting. The time, at which the guard starts, is not as important and critical as the response time of run-time evaluating of requests. This is the second process. The overall performance accumulates both performances of the start-up and evaluating processes.

– The performance of the start-up process is determined by the time of start-ups of guard's components. For example, a decision maker can faster initialize a knowledge base with only SBAC ontologies, or can initialize the knowledge base with all ontologies and semantic annotations.

– The performance of the evaluating process is determined by manipulations with semantic annotations and a decision making procedure. The decision making procedure is broken down to three activities. The first activity combines retrieved semantic annotations, applicable SBAC, domain and policy ontologies into the knowledge base. The second activity prepares a query according to the request and SBAC authorization rules. The third activity queries the prepared knowledge base.

Several major factors influence the performance of the SBAC enforcement mechanism. Preparation of the knowledge base for the decision making process can be allocated to both the start-up and evaluating processes based on the availability of semantic annotations and ontologies in different environments. This allocation shapes the balance between performances of the both processes. The complexity of querying the knowledge base differs because of different complexity of the authorization rules for policies that commit to different features of SBAC. The performance of the query execution is the most crucial for the evaluating process. The performance of other operations with the knowledge base impacts performances of the both processes.

The UML component diagram (Figure 3) depicts the architecture of the prototype for the SBAC enforcement mechanism. The internal structure of the guard consists of the decision maker and the query engine (query processor) provided by the ARQ processor of SPARQL queries. The decision maker has in-memory knowledge base (decision set) in the form of ontology model provided by the Jena framework. All ontologies are accessible via HTTP.

The fastest response time of the evaluating process corresponds to the simplest policy ontology. The policy ontology consists of one class of active resources with one individual, one class of passive resources with one individual and one class of operations with one operation. The policy has the only one privilege statement defined using the above described classes. All these data are loaded into the decision set during the start-up process.

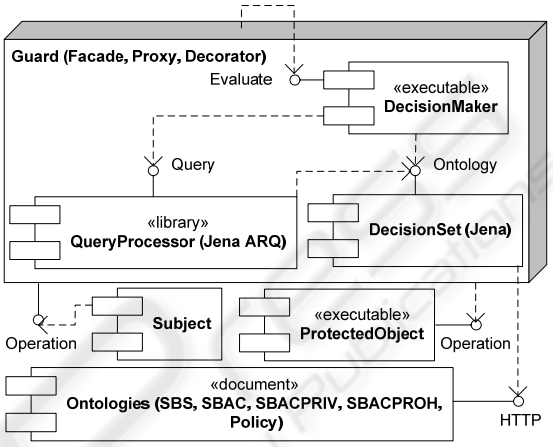

Figure 3: The architecture of prototype.

The cumulative CPU time of the guard start-up process is 12,256 seconds which are caused mainly by initializing the in-memory decision set (12,141 seconds). The average cumulative CPU time of the evaluating process is 0,813 seconds which are fully caused by the query execution over the decision set.

This cumulative CPU time is smaller than the real invocation time of both processes (14,559 and 2,05 seconds) but fairer for the comparison with fixed type of CPU because the overall cumulative time depends from number of characteristics of the hardware. The personal computer was used for testing. It was IBM PC with the CPU AMD Athlon XP 3000+, 1 GB of RAM, and OS Microsoft Windows XP Professional version 2002 with Service Pack 2.

5 CONCLUSIONS

There are several critical sections in the research on the SBAC ontologies. The suggested structure of access control statements reminds RDF statements. It is disputable whether this structure is universal enough to accommodate privilege, trust, trace, etc statements. The concepts defined in the SBAC ontologies require OWL Full profile that may cause problems in the stage of practical implementation, as long as existing reasoners do not fully support the

whole semantics of OWL. Thus, the ontologies could experience refinement based on practical needs. SBAC relies on the Semantic Web layers. The standards for some layers are still in active discussion and research. The provided feasibility study illustrates benefits of orientation to Semantic Web in reusability and expressivity. In general, the results are quite promising. The automated inferring makes the enforcement mechanism and the whole SBAC intelligent and flexible.

Presented in the paper ideas have clear practical and research implications. SBAC is an ambitious target. It further demands prototyping of ideas, reference implementations, and industrial deployments and evaluations. This should aim at rigorous and convincing specification of advantages.

The application of SBAC seems to be promising in areas where Semantic Web emerges and resources have their semantic annotations according to ontologies, for example multi-agent systems, semantic web services, semantic web portals, social networks, collaborative tools, etc. Semantic web services and agent technologies are the most promising because these environments already have means for ontologies and semantic annotations of resources (agents and services) and of operations (service processes and agent speech acts).

ACKNOWLEDGEMENTS

We are grateful for the financial support to the Rector and to the Department of Mathematical Information Technology, University of Jyväskylä.

REFERENCES

Berners-Lee, T., Hendler, J., and Lassila, O., 2001. The Semantic Web. *Scientific American*, Vol. 284, No. 5, pp. 34-43.

Gruber, T., 1993. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2): 199-220.

McGuinness, D., and Harmelen, F., (eds.). 2004. OWL Web Ontology Language Overview. *W3C Recommendation*, http://www.w3.org/TR/owl-features/

Moses, T., (ed.). 2005. eXtensible Access Control Markup Language (XACML) Version 2.0. *OASIS Standard*.

Nadalin, A., Kaler, C., Monzillo, R., Hallam-Baker, P., (eds.). 2006. Web Services Security: SOAP Message Security 1.1 (WS-Security 2004). *OASIS Standard*.

Naumenko A., Nikitin S., Terziyan V., Zharko A., 2005. Strategic Industrial Alliances in Paper Industry: XML- vs. Ontology-Based Integration Platforms, *The Learning Organization*, *Special Issue on: Semantic and Social Aspects of Learning in Organizations*, Emerald Publishers, Vol. 12, No. 5, pp. 492-514.

Naumenko A., Katasonov A., Terziyan V., 2007. A Security Framework for Smart Ubiquitous Industrial Resources, J.P. Müller and K. Mertins (Eds.), In *Proc. of the 3rd Int. Conf.. on Interoperability for Enterprise Software and Applications*, 13 pp. (In press).

Naumenko, A. and Luostarinen, K., 2006. Access Control Policies in (Semantic) Service-Oriented Architecture, Schaffert S. and Sure Y. (Eds.), In *Semantic Systems From Visions to Applications, Proc. of the SEMANTICS 2006*, Austrian Computer Society, Vienna, Austria, pages 49-62.

Naumenko, A., 2006. Contextual rules-based access control model with trust, Shoniregan C. and Logvynovskiy A. (Eds.), In *Proc. of the Int. Conference for Internet Technology and Secured Transactions*, e-Centre for Infonomics, London, UK, ISBN 0-9546628-2-2, pages 68-75.

O'Reilly T., 2005. What Is Web 2.0 Design Patterns and Business Models for the Next Generation of Software, http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html.

Patel-Schneider, P., Hayes, P., and Horrocks, I., (eds.). 2004. OWL Web Ontology Language Semantics and Abstract Syntax. *W3C Recommendation*, http://www.w3.org/TR/owl-absyn/

Prud'hommeaux, E., and Seaborne, A. (eds.). 2006. SPARQL Query Language for RDF. *W3C Candidate Recommendation*, http://www.w3.org/TR/rdf-sparql-query/

Qin, L. and Atluri, V., 2003. Concept-level access control for the Semantic Web. In *Proc. of the 2003 ACM Workshop on XML Security XMLSEC '03*. ACM Press, New York, NY, 94-103.

Tonti, G., Bradshaw, J., Jeffers, R., Montanari, R., Suri, R., and Uszok, A., 2003. Semantic web languages for policy representation and reasoning: A comparison of KAoS, Rei, and Ponder. In *Proc. of the Int. Semantic Web Conference*, pp. 419--437.

Wang, X., Lao, G., DeMartini, T., Reddy, H., Nguyen, M., and Valenzuela, E., 2002. XrML -- eXtensible rights Markup Language. In *Proc. of the ACM Workshop on XML Security. XMLSEC '02*. ACM Press, New York, NY, pp. 71-79.

Yagüe, M., Gallardo, M., and Maña, A., 2005. Semantic Access Control Model: A Formal Specification, In *Lecture Notes in Computer Science*, Springer, Volume 3679, pp. 24-43,

Yagüe, M., Maña, A., López, J., and Troya, J., 2003. Applying the Semantic Web Layers to Access Control. In *Proc. of the Int. Workshop on Web Semantics*, IEEE Computer Society Press, pages 47–63.

Yergeau, F., Bray, T., Paoli, J., Sperberg-McQueen, C., and Maler, E., 2004. Extensible Markup Language (XML) 1.0 (Third Edition). *W3C Recommendation*, http://www.w3.org/TR/2004/REC-xml-20040204/