Semantic Descriptions of Web Services Security Constraints

Dong Huang
Siemens AG, Corporate Technology
Otto-Hahn-Ring 6, 81739 Munich, Germany
dong.huang.ext@siemens.com

Abstract

Non-functional descriptions of web services and business rules play an important role in specification and analysis of the security constraints of web services. As existing approaches do not provide logic and semantic model for the web services security constraints, sharing and reasoning over them are infeasible. The proposal builds upon the project AKT's[1] work in defining a Semantic Web Constraint Interchange Format (CIF), which itself builds on the proposed Semantic Web Rule Language (SWRL).

The main contributions of this paper are a new ontology for representing security constraints as policy and a semantic policy framework for the management of the policies; we also show the possibility to integrate the business rules and non-functional descriptions into policy specification by means of converting them into Constraint Satisfaction Problem (CSP) using CIF.

1 Introduction

The concept of Web Services is thought to be the next generation of e-business architectures for the web. Such a Service-Oriented Architecture (SOA) describes principles for creating dynamic, loosely coupled systems based on services.

Two problems arise from the application of web services in e-business:

1. *Change management.* Nowadays e-business scenarios always involve more than one organization. Changes to business rules should take effect in an efficient way without recoding or stopping the system.

2. *Gaps between Business and IT.* Managers, who work at the business level, tend to use abstract, high-level and human-readable policies to specify requirements or describe the services. At the IT level, policies are usually specified in a machine-readable way and enforced according to the specific environment. Knowledge exchange and sharing is necessary in this case to build an intelligent service.

To solve these problems, a security policy framework for business processes was proposed in [7], which gave a preliminary idea of policy management and implementation in SOA. The service security constraints can be interpreted as policies using script language or compiled structure programming language. We prefer using ontology and rules language, such as Web Ontology Language (OWL)[11]or Semantic Web Rule Language (SWRL)[5], to represent our declarative policy. This allows us flexibly and dynamically to represent all kinds of service constraints, such as security, privacy, Quality of Service (QoS), and Digital Rights Management(DRM), etc. It will be a challenge to define and translate above abstract web service constraints to SWRL-based executable policy[6].

The central idea in our approach is to gather pertinent data/knowledge from multiple stakeholders in the e-business scenario, along with constraints specified by non-functional requirements of web services and business rules. These data and constraints are then fused by mediator software into a dynamically composed Constraint Satisfaction Problem (CSP), which is then dispatched to a solver inside the semantic policy framework. The security constraints are expressed against a semantic data model/ontology because it may be necessary to transform them at run-time. Security constraints in our approach are represented using an expressive quantified constraint language, the Constraint Interchange Format (CIF)[12].

The remaining sections of this paper are structured as follows. Section 2 outlines the requirements of the security constraints specification language for web services and introduces the principle of Constraint Interchange Format. Section 3 gives an overview of our semantic policy framework for web services. Section 4 surveys related works. Section 5 includes the future research direction for the work and conclusion.

[1] http://www.csd.abdn.ac.uk/research/akt/

2 Representing Security Constraints

Security of web services is a multi-dimensional concept. The multiple dimensions of security include Authentication, Access control, Audit, Confidentiality, Integrity, Availability and Non-repudiation. Generally, when security is referred to, it essentially implies one or more of the above dimensions of security.

2.1 Security Constraints Specification Language

Various approaches have been done to achieve security constraints specifications, including logic-based languages, role-based access control, various access controls and trust specification techniques [3]. But, a specification language, which can meet the following requirements, is still missing.

- How to model non-functional properties into the policies and enable reasoning over them?
- How to integrate business rules with the knowledge base and specify policy using rules?

Various web services and semantic web services approaches have been investigated to describe the non-functional properties of a service. In [14] a set of the most relevant non-functional properties for web services and their modeling are described. An overview of all these approaches is given in [16].

Business Rules are used for categorizing facts important to a business. They also require or prohibit actions by a business. OMG[2] has been widening its scope to include business modeling. Several of its recent requests for proposals have been about or related to business rules. These proposals are Semantics of Business Vocabulary and Business Rules, Production Rule Representation, Business Rule Management. We define the constraints specification language with an upper ontology in Figure 1.

- **Service Domain** is a collection of services. The service types can be composite service and atomic service. The services may be chained together to a composite service for special business goals or processes.
- **Policy Domain** categorizes the policies by different aspects like security, trust and management. Meta-Policy, so called policy of policy, can be defined in this domain.
- **Rule Domain** aims to represent the requirements of business activities, which may be application-specific terms, e.g. Legal Rules applied to online-shopping.

[2]http://www.omg.org

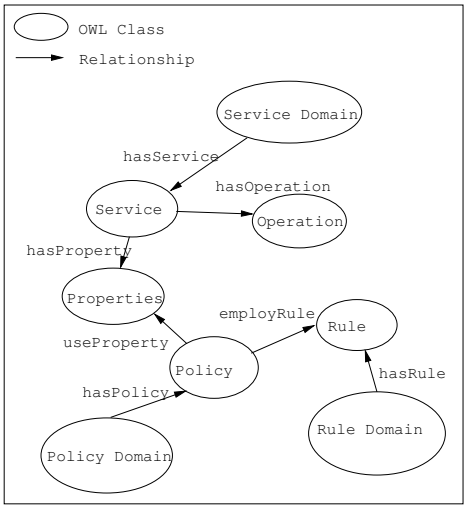

Figure 1. Upper ontology for policy

- **Properties** apply for all service descriptions: functional, behavioral and non-functional.
- **Rule** is a statement that can be represented as IF *Condition* THEN *Action*.

From the upper ontology, a domain-specific ontology describes the vocabularies, business rule terms, service descriptions used within the domain. By using the domain-specific specification template, a constraints specification can be generated and exchanged automatically with Ontology language OWL, Rules language SWRL and CIF, which is described in the next section.

2.2 Constraint Interchange Format

Constraint Interchange Format (CIF) is based on the Colan[1] constraint language, which is based on range restricted first order logic (FOL). Earlier versions of the language were aligned with Resource Description Framework (RDF) [9] and SWRL. CIF constraints are essentially defined as quantified implications, so we re-use the implication structure from SWRL, but allow for nested quantified implications within the consequent of an implication. An example CIF constraint is shown in human-readable SWRL-style syntax below:

$$(\forall ?x \in X, ?y \in Y) p(?x, ?y) \wedge Q(?x) \Rightarrow$$
$$(\exists ?z \in Z) q(?x, ?z) \wedge R(?z) \Rightarrow$$
$$(\forall ?v \in V) s(?y, ?v)$$

In[12], an RDF/XML syntax is provided as an extension to the one given for SWRL to support publishing and interchange of CIF constraints. A new **rdfs:Class** Constraint,

with properties *hasQuantifiers* and *hasImplication* is defined. For example, if we wanted to introduce a business requirement like "every delegation group must contain at least one participant from government", the following code shows RDF/XML for this constraint.

```
<cif:Constraint>
 <cif:hasQuantifiers
      rdf:parseType="Collection">
  <cif:Forall>
   <cif:var rdf:resource="#g"/>
   <cif:set rdf:resource="#Delegationgroup"/>
  </cif:Forall>
  <cif:Exists>
   <cif:var rdf:resource="#p"/>
   <cif:set rdf:resource="#Government"/>
  </cif:Exists>
 </cif:hasQuantifiers>
 <cif:hasImplication>
   <swrl:Imp>
    <swrl:body rdf:parseType="Collection"/>
    <swrl:head rdf:parseType="Collection">
     <swrl:IndividualPropertyAtom>
      <swrl:classPredicate
            rdf:resource="#has-member"/>
      <swrl:argument1 rdf:resource="#g"/>
      <swrl:argument2 rdf:resource="#p"/>
     </swrl:IndividualPropertyAtom>
    </swrl:head>
  </swrl:Imp>
 </cif:hasImplication>
</cif:Constraint>
```

RDF/XML for the constraints

Rule Interchange Format (RIF)[3] is another interchange formats for logic expressions on the Web. It is expected that CIF will evolve to use RIF in place of SWRL as the new format takes shape. As it is currently planed, Phase 1 RIF is essentially Horn Logic. If Phase 2 RIF includes full FOL then this format may wholly subsume CIF. At that point it is conceivable to simply define CIF as a subset of RIF: constraints would be interchanged in RIF itself[15].

3 Semantic Policy Framework

In order to support conflict resolution and life-cycle management of policies, as well as enable reasoning over the knowledge base, an architecture of the semantic policy framework is illustrated in Figure 2. It includes two supporting services: a policy service and a knowledge service. Additional components include: reasoner, management tools and repository.

- **Policy Service.** The policy service acts as a Policy Decision Point (PDP) [4] for web services policies, including security and QoS requests. The policy service acts on service requests and renders a decision.
- **Knowledge Service.** The knowledge service manages the Virtual Knowledge Community (VKC)[8] and the repository of static knowledge base: business rules, policies and service descriptions. The concept of a *virtual knowledge community (VKC)* enables us to model *corporate knowledge* as the amount of knowledge provided by individual agents [10].

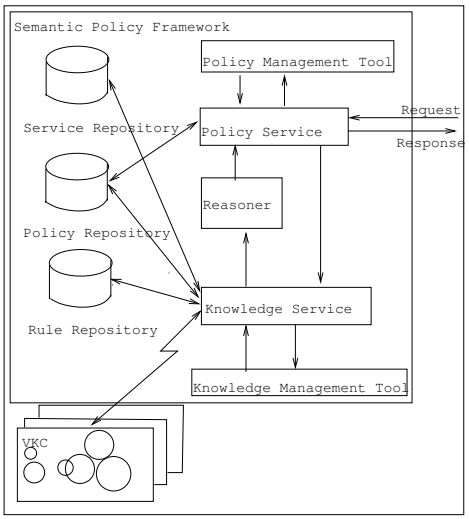

Figure 2. Framework architecture

- **Repositories.** Three repositories are used to store service descriptions, policies and business rules. This information can be managed through the policy and knowledge service.
- **Reasoner.** Reasoner is used to perform logical inference over corporate knowledge based on the knowledge repositories and VKCs. In our implementation, we use KAON2 [5] as reasoner. The major advantage of KAON2 is that it is a very efficient reasoner when it comes to reasoning with Description Logics ontologies containing very large ABoxes and small TBoxes[13]. The terms Abox and Tbox are used to describe two different types of statements in ontologies.
- **Management Tools.** The policy management tool acts as the interface to policy service; it manages the life-cycle of policies, creates and deploys new policies. The knowledge management tool provides an interface to manage the VKCs and knowledge repositories.

As the policy service acts as a PDP, it receives the service request from the Policy Enforcement Point (PEP) at the service end. There are two possible options: a) It finds out the suitable policies from the policy repository, renders the decision and sends the result back to PEP, b) It cannot find the proper policy or there are conflicts in policies, the

[3] http://www.w3.org/2005/rules/

[4] http://www.ietf.org/

[5] http://kaon2.semanticweb.org/

request will be converted to a request to the knowledge service. The knowledge service will analyze the request and prepare a knowledge base for the next reasoning step. Based on the result of reasoning on the knowledge base, the policy service can make its decision of the service request.

4 Related Work

WS-Policy[6] defines a framework and a model for the expression of the capabilities, requirements, and general characteristics of entities in an XML Web Services-based system as policies. In [4], a complete policy-based management framework is presented, which includes a policy specification language and architecture for deploying policies. KAoS Policy and Domain Services[2] use ontology concepts encoded in OWL to build policies. These policies constrain allowable actions performed by actors which might be clients or agents. All of these approaches did not address knowledge exchange and sharing among all the stakeholders in the e-business scenario.

Rein[7] is a decentralized framework for representing and reasoning over distributed policies in the Semantic Web. Rein (Rei and N3) uses high level Rei concepts for policies and N3 rules to connect these policies to each other and the Web. In our approach, the framework is centralized, which is designed to support the special service platform and act as a policy management component.

5 Conclusion and Future Works

In this paper, we have proposed a representation for security constraints at the Semantic Web logic layer. We illustrated the use of the CIF/SWRL constraints and new upper ontology to integrate the business rules and non-functional descriptions of web services in the policy specification. By integrating the knowledge management service, the semantic policy framework is able to access the application-specific information about the transaction in e-business outside.

The development of the semantic policy frmework is ongoing; currently we are trying to employing different kinds of Reasoner to evaluate the complexity, scalability and performance. The aim is to enable a semantic policy framework and knowledge management methodology, which enable security, trust and QoS in service-oriented computing environments and provide a novel solution for fields like e-business, telecommunication and enterprise application integration.

[6]http://www.ibm.com/developerworks/library/ws-polfram

[7]http://groups.csail.mit.edu/dig/2005/05/rein/

References

[1] N. Bassiliades and P. G. CoLan. A functional constraint language and its implementation. *Data and Knowledge Engineering*, pages 203–249, 1994.

[2] J. Bradshaw and A. Uszok. Representation and reasoning for daml-based policy and domain services in kaos and nomads. In *AAMAS '03: Proc. of the second international joint conference on Autonomous agents and multiagent systems*, pages 835–842, New York, NY, USA, 2003. ACM Press.

[3] N. Damianou, A. Bandara, M. Sloman, and E. Lupu. A survey of policy specification approaches. Technical report, Imperial College, UK, 2002.

[4] N. Damianou, N. Dulay, E. Lupu, and M. Sloman. Ponder: A language for specifying security and management policies for distributed systems. Technical report, Imperial College, UK, October 2000.

[5] I. Horrocks and P. F. Patel-Schneider. SWRL: A semantic web rule language combining OWL and RuleML. Technical report, The Rule Markup Initiative, May 2004.

[6] Y.-J. Hu. Combining ontology and rules as service constraint policy for P2P systems. In *Proc. of WWW2005*, Chiba, Japan, May 2005.

[7] D. Huang. Semantic policy-based security framework for business processes. In *Proc. of the Semantic Web and Policy Workshop*, Galway, Ireland, November 2005.

[8] D. Huang, Y. Yang, and J. Calmet. Modeling web services policy with corporate knowledge. In *Proc. of 2006 IEEE International Conference on e-Business Engineering*, Shanghai, China, October 2006.

[9] K. Hui, S. Chalmers, P. Gray, and A. Preece. Experience in using RDF in agent-mediated knowledge architectures. *Agent-Mediated Knowledge Management (LNAI 2926). Springer-Verlag*, pages 177–192, 2004.

[10] P. Maret and J. Calmet. Modeling corporate knowledge within the agent oriented abstraction. In *Proc. of International Conference on Cyberworlds (CW'04)*, pages 224–231. IEEE Computer Society, 2004.

[11] D. L. McGuinness and F. van Harmelen. OWL: Web ontology language overview. Technical report, W3C Recommendation, February 2004.

[12] C. McKenzie, P. Gray, and A. Preece. Extending SWRL to express fully-quantified constraints. In *Proc. of RuleML 2004 Workshop at ISWC 2004*, Hiroshima, Japan, November 2004.

[13] B. Motik, U. Sattler, and R. Studer. Query answering for OWL-DL with rules. In *Proc. of International Semantic Web Conference 2004*, pages 549–563, Hiroshima, Japan, November 2004.

[14] J. O'Sullivan, D. Edmond, and A. H. M. ter Hofstede. Formal description of non-functional service properties, queensland university of technology. Technical report, http://www.service-description.com, 2005.

[15] A. Preece, S. Chalmers, C. McKenzie, J. Pan, and P. Gray. Handling soft constraints in the semantic web architecture. In *Proc. of RoW2006 Reasoning on the Web at WWW2006*, Edinburgh, UK, 2006.

[16] I. Toma and D. Foxvog. Non-functional properties in web services. Technical report, DERI, 2006.