A Semantic Approach for Access Control in Web Services

M. I. Yagüe, J. Mª Troya
Computer Science Department, University of Málaga, Málaga, Spain
{yague, troya}@lcc.uma.es

Abstract

One of the most important features of XML Web services is that they can be easily accessed over the Internet, but this makes them vulnerable to a series of security threats. What makes security for web services so challenging is their distributed and heterogeneous nature. In this sense, this paper presents an access control system for Web services. We introduce the *Semantic Policy Language* (SPL) for the description of access control criteria based on the use of attribute certificates. This language has been specifically designed to take advantage of semantic information about resources and the context to achieve full (syntactic and semantic) validation of policies. Furthermore, another objective in its design has been to facilitate the security management. In particular, SPL is modular, enables the abstraction and reuse of components, the composition of SPL policies in an unambiguous way, and the dynamic instantiation of parameters based on semantic properties about resources. Finally, the semantic integration of a Privilege Management Infrastructure (PMI) in access control systems of heterogeneous Web services built upon SPL enables their interoperability.

1. INTRODUCTION

XML Web Services are the basic building blocks in the move to distributed computing on the Internet. Applications are constructed using multiple XML Web Services from various sources, which cooperate regardless of their location and implementation. Although we can find many different definitions of XML Web Services, most of them agree in that:

- XML Web Services expose useful functionality to Web users through a standard Web protocol. In most cases, the protocol used is SOAP[1].
- XML Web services provide a way to describe their interfaces in enough detail to allow a user to build a client application to talk to them. This description is usually provided in an XML document called a Web Services Description Language (WSDL) document [2].
- XML Web services are registered so that potential users can find them easily. This is done with Universal Discovery Description and Integration (UDDI) [3].

One of the most important features of XML Web services is that they can be easily accessed over the Internet, using ubiquitous web protocols and data formats such as HTTP and XML. Nevertheless, this ease of access makes them vulnerable to a series of security threats. In exposing critical business functions to the Internet, WS can expose valuable corporate data, applications and systems to a variety of external threats. Much debate has been focused on the use of HTTP as the access protocol, while other issues have not received so much attention. Actually, access control for Web Services (WS), when present, is based on the same mechanisms used for web pages. However, each WS presents different security requirements. Therefore, the inherent heterogeneity of Web Services has to be taken into account in the design of the access control model.
This paper presents an access control system for Web services. We describe the *Semantic Policy Language* (SPL) for the specification of access control criteria based on the use of attribute certificates. Means to integrate *Privilege Management Infrastructures* (PMIs) and to facilitate administration based on semantic information are also presented. Semantic information is used in our approach for the integration of the external PMI, semantic validation of policies, and dynamic instantiation of parameters of the policies based on properties of the resources to be accessed. Summarising, our proposal is scalable, facilitates the

administration of the access control system and enables semantic integration and interoperability of heterogeneous Web Services.

2. BACKGROUND

XML Web services are supported by the combination of three standards: SOAP, UDDI and WSDL. SOAP is used to show the way to talk to XML Web Services, UDDI is used to publish them and WSDL is used to describe their functionality. However, some other questions such as the security requirements have not been considered in these standards. Web Services demand a security framework for access control that does not impede the exchange of data essential for their success. Languages such as WSDL represent a valuable tool for the description of functionality but not for security properties. This is reasonable, since WSDL is designed to be used by clients of the Web Service to learn what it has to offer. On the other hand, a mechanism to describe and enforce security requirements will not be used directly by clients, but by the access control system. Furthermore, functional descriptions are public while security properties are usually confidential. Therefore, secure interoperability requires additional mechanisms.
WS-Security [4] is a recent initiative that describes enhancements to SOAP to provide protection of messaging. WS-Security also provides a general-purpose mechanism for associating security tokens with messages and describes how to encode binary security tokens such as X.509 certificates [5]. Although it does not address other security issues, such as authorization or access control, WS-Security represents a useful initiative to support other security services.
Attribute certificates are the foundation for scalable and flexible access control schemes since access conditions are expressed in terms of sets of attributes instead of users or groups. This approach scales well in the number of users and also in the number of different factors (attributes) used by the access control system. Attribute certificates provide means to transport authorization information to decentralized applications. In this way authorization information becomes mobile and interoperable, which is highly convenient for scenarios such as Web Services.
Taking into account security, scalability and interoperability it is wise to separate the certification of attributes and access control management responsibilities. Therefore the access control system needs to be complemented by an external component, the PMI, that provides certification functions. In a PMI, several entities, known as *Source Of Authorizations* (SOA), issue attribute certificates. Usually, each SOA issues certificates for a small number of semantically related attributes. Because the certification function is external, a mechanism to establish the trust between the access control systems and the PMI is required. For this problem, our solution uses semantic information, expressed by means of XML metadata standards such as RDF [6] and XML Schema [7], about the certifications issued by each SOA.
In the case of access control systems for WS, the integration of an external PMI represents a step towards the solution of problems such as scalability, interoperability and separation of duties. Semantic integration is the best approach to interoperability in this case, as it allows valuable information to be described and exploited. This is a very interesting application scenario. Its relevance is derived from the security requirements of the environment and the necessity for access control systems to understand the semantics of the attribute certificates managed by the PMI.

3. RELATED WORK

XML Web Services can be implemented so that only authorized clients (end users, computers or businesses) can be authorized to access them. Several proposals have been introduced for access control to distributed heterogeneous resources from multiple sources based on the use of attribute certificates and PMIs. The Akenti Project [8] proposes an access control system to restrict access to distributed resources controlled by multiple stakeholders. The requirement for the stakeholders to trust the rest of the servers in the network and some security vulnerabilities related to the existence of positive and negative use-conditions are the main drawbacks of Akenti. The PERMIS Project [9] objective is to set up an integrated infrastructure to solve identification and authorization problems. Because the PERMIS system is based on the RBAC [10] model, it shares its limitations. The PERMIS researchers have examined various policy languages concluding that XML is the most appropriate candidate for a policy specification language.
In this line, XML-based languages have been proposed for access control, digital rights management, authentication and authorization. Although many similarities and interesting features can be found among them, some other such as policy parameterisation and composition are not supported. Moreover, some features provided by those languages are not necessary in Web Services scenarios [11].
Two relevant access control systems using XML are the Author-X system [12] and the FASTER project [13].

Because both systems have been specifically developed to control access to XML documents, they do not fit naturally in the WS environment. While Author-X policy language uses DTDs, our Semantic Policy Language and FASTER use XML-Schema. Additionally, our access control uses a second XML metadata technology, RDF. The access control scheme described in [13] has been applied to the XML structure of SOAP calls [14]. Since this access control is based on user groups, roles and physical locations, following the technique of defining a subject hierarchy, it is not adequate for scenarios where the structure of groups can not be anticipated. Moreover, in our case, new Web Services are incorporated dynamically and each one may need a different group structure and access control policy. Furthermore, the policy for a given Web Service may change frequently. Another characteristic of the work described in [13] is the use of hierarchies and the propagation of the authorizations of a group to all its members, of a role to all its specializations and of a location pattern to all the machines in its subnetwork. As consequence, for the general case of Web Services with non-trivial security requirements the number of different authorizations (positive and negative) that have to be defined grows up rapidly. Additionally, negative authorizations granted to roles are unreliable as stated in [13]. A different approach is followed in our work. Based on the separation of the access control function from the authorization (credential issuance or attribute certification in our case), we propose the integration of an external PMI supported by semantic information about the certification entities. An additional advantage of our solution is that semantic and contextual validation of policies is made possible.

4. DESCRIPTION OF THE PROPOSAL

4.1 Semantic Language Components

Although other XML-based languages have been developed for access control and authorization, their genericity results in a high complexity. Furthermore, many of their features are not useful in this environment. On the other side, some important features of SPL are not considered in these languages [11]. For this reason we have developed a specific XML-based language to specify the access control policies. This language is called *Semantic Policy Language* because it is based on the semantic properties of the resources to be accessed, the PMI and the context. These semantic properties are used during the processes of specification of access control criteria, dynamic policy allocation, parameter instantiation and policy validation.

Because the definition of access control policies is a complex activity that presents many similarities with computer programming we have included some of the mechanisms used in this field to reduce the complexity, such as modularity, parameterisation and abstraction. In order to provide the simplicity and flexibility required in complex systems, our solution is based on the modular definition of policies. Modularity in our solution implies: (a) the separation of specification in three parts; that is, access control criteria, allocation of policies to resources and semantic information (properties about resources and context); (b) the abstraction of access control components; (c) the ability to reuse these access control components; and (d) the reduction of the complexity of management due to previous properties. Moreover, the use of semantic information about the context allows the administrator to include contextual considerations in a transparent manner, also helping the (semantic) validation task.

Usual components of access policies include the target resource, the conditions under which access is granted/denied and, sometimes, access restrictions. Opposed to other languages, specifications in SPL do not include references to the target object. Instead, a separate specification called *Policy Applicability Specification* (PAS) is used to relate policies to objects dynamically when a request is received. Both *SPL Policies* and *PAS* use semantic information about resources included in *Secured Resource Representation* (SRRs) and other contextual information documents. *SPL Policies* and *PAS* can be parameterised allowing the definition of flexible and general policies and reducing the number of different policies to manage. Parameters, which can refer to complex XML elements, are instantiated dynamically from semantic and contextual information. Finally, policies can be composed importing components of other policies without ambiguity. This compositional approach allows us to define the abstract meaning of the elements of the policies, providing a mechanism to achieve abstraction, which also helps in reducing the complexity of management. Tools developed to graphically manage the relations among policies and with other components are also essential for a simple and flexible management.

The schema for SPL specifications is represented as a set of XML-Schema templates that facilitate the creation of these specifications, allowing their automatic syntactic validation. Figure 1 shows the structure of the SPL language.

SPL policies can include locally defined components as well as imported elements. The ability to import elements enables the modular composition of policies based on the XPath standard [15]. An SPL Policy is

composed of a set of *access_Rule* elements, each one defining a particular combination of attribute certificates required to gain access, associated with an optional set of actions (such as *Notify_To*, *Payment* and *Online_Permission*) to be performed before access is granted. In this way provisional authorization or *provision-based access control* (PBAC) [16] is enabled in SPL. Figure 2 shows an example of an *SPL policy* requiring an attribute certificate stating the client is an authorized broker.

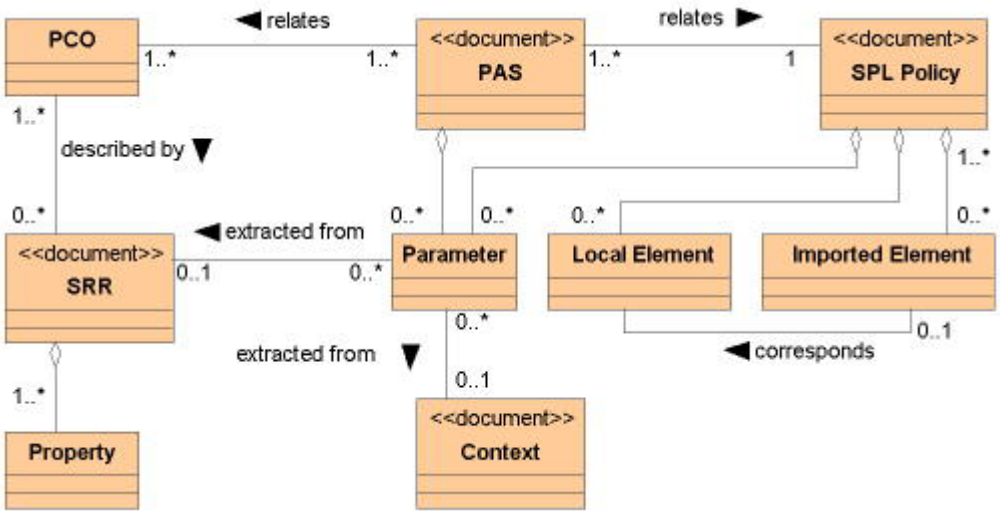

FIG. 1: Conceptual model of the SPL Language

The Policy Applicability Specification provides an expressive way to relate policies to resources, either explicitly or based on the metadata about the objects (e.g. type of content, owner, price, etc.). PAS documents include three main elements: policy, objects and instantiation. The policy element indicates which policy is applicable to the specified objects. Objects are defined by their location and conditions to be fulfilled by the semantics of these objects (SRRs). Optionally, operation elements can be used to define which operations of the target object are controlled by the declared policy, allowing a finer grained access control. In case no operation element is included, the policy is applicable to all of the object operations. The instantiation element describes the mechanism to instantiate parameters in the policies. Figure 3 shows an example of applicability rules for SPL policies to WS indicating that the Consulting_Access.xml is applicable to all WS of type 'Investment' in http://www.lcc.uma.es/Consultancy_WS. The Secured Resource Representation is a simple and powerful mechanism to describe properties about resources. Properties described in SRRs are used for the instantiation of policies and PAS, and to locate the applicable policies. An example of a SRR is included in figure 2. Dynamic allocation of policies to resources is a very flexible and useful mechanism that solves the problem of associating policies to newly created objects. The use of dynamic policy allocation needs a rich set of metadata about the resources. This semantic meta-model is used to locate the right policy for each resource, based on its relevant properties.

```
<?xml version="1.0" encoding="UTF-8"?>
<spl:policy xmlns:spl="http://www.lcc.uma.es/WS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.lcc.uma.es/WS
PolicyTemplate.xsd">
   <spl:access_Rules>
      <spl:access_Rule Name="Authorized_Brokers" Public="false">
         <spl:attribute_Set>
            <spl:attribute attributeID="Authorized_Broker">
            <spl:attribute_Name>Function</spl:attribute_Name>
            <spl:attribute_Value>Broker</spl:attribute_Value>
            <spl:SOA_ID>CBOT_ADMIN</spl:SOA_ID>
               <!-- Chicago Board Of Trade Administration -->
            </spl:attribute>
         </spl:attribute_Set>
      </spl:access_Rule>
   </spl:access_Rules>
</spl:policy>
```

a) Consulting_Access.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<spl:SRR xmlns:spl="http://www.lcc.uma.es/WS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.lcc.uma.es/WS SRR_WS.xsd"
resource="http://www.uma.es/Consultancy_WS[Profit_Classification]">
<!-- Access control properties -not public- about the Profitability
Classification Operation of the Consultancy Web Service -->
      <spl:property_Name>responsible</spl:property_Name>
      <spl:property_Value>admin@consulting.com</spl:property_Value>
   </spl:property>
   <!-- e-mail of the administrator -->
   <spl:property>
      <spl:property_Name>scope</spl:property_Name>
      <spl:property_Value>local</spl:property_Value>
   </spl:property>
   <!-- the operation does not access external resources -->
</spl:SRR>
```

b) SRR for the Profitability_Classification Operation

FIG. 2: Policy and SRR Examples

```
<?xml version="1.0" encoding="UTF-8"?>
<spl:PAS xmlns:spl="http://www.lcc.uma.es/WS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.lcc.uma.es/WS pas.xsd">
        <!-- this PAS states that 'Consulting_Access' is applicable to the
        'Profitability_Classification' operation of all the Web Services of type
        'Investment' in 'http://www.lcc.uma.es/Consultancy_WS/' -->
   <spl:policy>http://www.uma.es/WS/Consulting_Access.xml</spl:policy>
   <spl:object>
      <spl:object_Location>http://www.lcc.uma.es/Consultancy_WS</spl:object_Location>
      <spl:operations>
         <spl:operation>Profitability_Classification</spl:operation>
      </spl:operations>
      <spl:conditions>
      <spl:condition predicate="equals">
         <spl:property_Name>WS_Type</spl:property_Name>
         <spl:property_Value>Investment</spl:property_Value>
      </spl:condition>
   </spl:conditions>
   </spl:object>
</spl:PAS>
```

FIG. 3: PAS for the Consulting_Access.xml

5. SYSTEM OVERVIEW

A general overview of the main components of the system and their relation is depicted in figure 4. We can distinguish three different systems: the *WS client*, the *WS server* and the *PMI*. Several *PMI Nodes*, some of which are *SOAs*, compose the *PMI*. Each *SOA* produces and digitally signs *Source Of Authorization Descriptions* (SOADs) to express the semantics of the attribute certificates it issues. These RDF documents describe the different attributes certified by each *SOA*, including their names, descriptions and relations. *SOADs* are used to establish the trust between the *PMI* and the access control systems. They convey the information needed by the access control system to understand the semantics of the attribute certificates, which is essential in order to make sound access decisions. The information contained in *SOADs* is also essential for the semantic validation of the policies, enabling the detection of semantically incomplete or incorrect policies. The set of *SOADs* represents the semantic description of the *PMI*. Full integration of the *PMI* can be achieved transparently for the rest of the system thanks to this description.
The *WS Server* includes several components related to the access control. An access control proxy, *AC Proxy*, for the Web Services is included in order to provide a transparent access control service for both clients and WS developers. However, application-level access control is sometimes an important requirement. For security-aware Web Services the proxy might not be required. The *AC Proxy* intercepts the calls to the *WS*. After receiving an access request the *AC Proxy* requests the attribute certificates from the *PMI Client* and forwards the authorization request to the Access Control. Using the *SOADs* to determine which *PMI Node* must be contacted, the *PMI Client* requests the attribute certificates from that *PMI Node*.
The *Access Control* component is responsible for producing access decisions by performing dynamic allocation and policy evaluation. When a request is received, it analyses the semantic metadata available for the target resource contained in the SRR, finds the appropriate *Policy Applicability Specifications* (PAS) and retrieves the necessary *SOADs*. Using this information, the *Access Control* is able to find the applicable policies. These policies are then analysed and instantiated using the metadata about the resource (*SRR*) and the context. Finally, all policies are combined and evaluated producing an access decision that is returned to the *AC Proxy*. This process is called *dynamic policy allocation*.
The last component, called *Policy Assistant*, is responsible for the administrative tasks. It uses the *SOADs*, *SRRs* and the *Context* metadata to produce and validate *SPL Policies* and *PAS*.

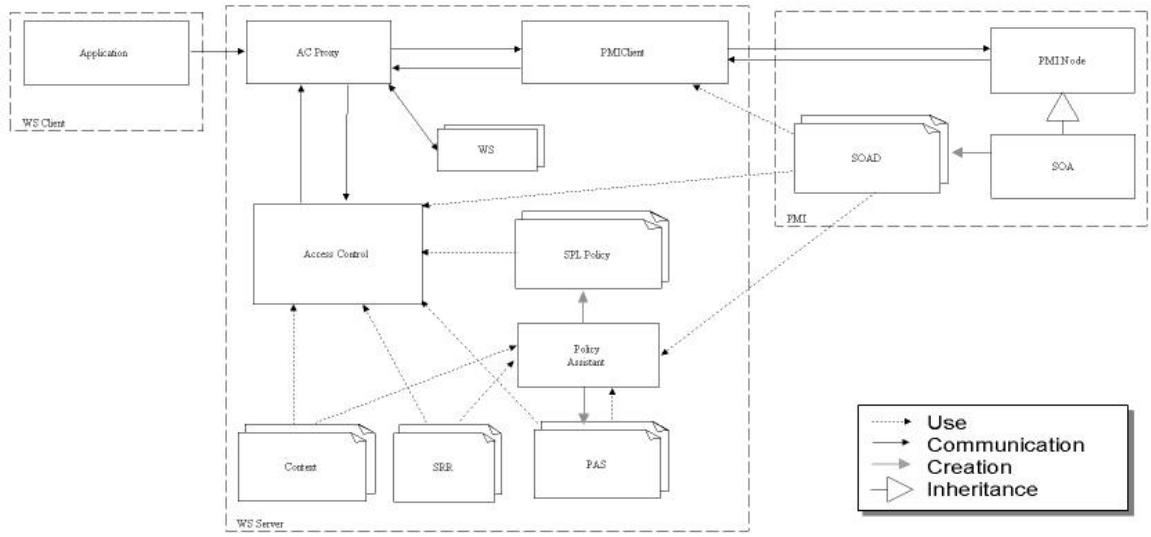

FIG. 4: Overview of the system

6. SEMANTIC AND CONTEXTUAL POLICY VALIDATION

The creation and maintenance of access control policies is a difficult and error prone activity. The *Policy Assistant* component is designed to help administrators to specify those policies and validate them to find syntactic and semantic errors. It includes components for the automated validation of policies at different levels. SPL policies are validated syntactically using XML Schema. Semantic validation is made possible by the use of our *Semantic Policy Validator*, included in the *Policy Assistant* component. This component has been developed using the DOM API [17]. Its function is to parse the *SPL policy* validating it with respect to the semantic information available through the metadata defined. An interesting feature of the *Policy Assistant* is that allows policies to be validated in the context where they will be applied. Policy context validation is based on the *SOADs* and the *Context* metadata. The higher expressiveness of SPL specifications along with the additional semantic information in the form of W3C standard metadata allows an easy semantic integration of our access control system with a PMI enabling interoperability among access control mechanisms of different Web Services.

To illustrate the context and semantic validation of policies, lets consider an investment adviser application that accesses different Web Services offered by consulting firms. Those Web Services classify a series of products according to their estimated profitability. Clients use the application to decide which products to invest to, based on the classifications received from the different consulting firms. Each web service wants to grant access to authorized brokers but at the same time, they need to prohibit the access to competing firms. This is a case where the use of negative authorizations seems reasonable. As we have yet mentioned, in access control systems based on credentials or attribute certificates, negative authorizations represent a problem because a user can avoid being subjected to a negative authorization simply by not presenting the corresponding certificate. Solutions proposed so far are unable to solve this problem because no information about the context is considered.

Our approach uses semantic information about the context to solve this situation without necessity of negative authorizations. For example, the context metadata can state that each authorized brokers is either member of a partner firm or member of a competing firm. This information is not always obvious for the administrator. Therefore, based on context metadata, the access control administrator can realize that the policy must require membership in a partner company for granting access. Figure 5 shows the resulting policy. This policy checks that the user is an authorized broker importing the corresponding attribute from the policy defined in figure 2. It declares a parameter called *Company*. Notice that, when instantiated, the parameter references will be resolved to attributes of the actual parameter. The *PAS* for this policy is shown in figure 6. Instantiation criteria for the *Company* parameter are declared in this document. The actual parameter is an XPath reference to a list of partners stored as semantic information about the context.

```
<?xml version="1.0" encoding="UTF-8"?>
<spl:policy xmlns:spl="http://www.lcc.uma.es/WS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.lcc.uma.es/WS
http://www.uma.es/WS/PolicyTemplate_WS.xsd">
   <spl:parameter>Company</spl:parameter>
   <spl:access_Rules>
      <spl:access_Rule Name="Partner_Member" Public="false">
         <spl:attribute_Set>
            <spl:attribute predicate="equals">
               <spl:attribute_Name>Member</spl:attribute_Name>
               <spl:attribute_Value>*Company[@name]</spl:attribute_Value>
               <spl:SOA_ID>*Company[@SOA_ID]</spl:SOA_ID>
            </spl:attribute>
            <spl:import Url="Consulting_Access.xml"
               XPath="//attribute[@attributeID='Authorized_Broker']"/>
         </spl:attribute_Set>
      </spl:access_Rule>
   </spl:access_Rules>
</spl:policy>
```

FIG. 5: Partner Membership Policy

```
<?xml version="1.0" encoding="UTF-8"?>
<spl:PAS xmlns:spl="http://www.lcc.uma.es/WS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.lcc.uma.es/WS pas.xsd">
      <!-- this PAS states that 'MemberShip_Partner' is applicable to the
      Profitability_Classification' operation of all the Web Services of type
      'Investment' in 'http://www.lcc.uma.es/Consultancy_WS/' -->
   <spl:policy>http://www.uma.es/WS/Consulting_Access.xml</spl:policy>
   <spl:object>
      <spl:object_Location>http://www.lcc.uma.es/Consultancy_WS</spl:object_Location>
      <spl:operations>
         <spl:operation>Profitability_Classification</spl:operation>
      </spl:operations>
      <spl:conditions>
         <spl:condition predicate="equals">
            <spl:property_Name>WS_Type</spl:property_Name>
            <spl:property_Value>Investment</spl:property_Value>
         </spl:condition>
      </spl:conditions>
   </spl:object>
   <spl:instantation>
      <spl:formal_Parameter>Company</spl:formal_Parameter>
      <spl:actual_Parameter path="Partners_List//Company">
                http://www.uma.es/WS/SOAD/MemberShip_Partner_Context.xml
      </spl:actual_Parameter>
      <!-- Partners_List contains names and SOADs of our partner firms -->
   </spl:instantation>
</spl:PAS>
```

FIG. 6: PAS for Partner Membership Policy

7. CONCLUSIONS AND FUTURE WORK

We have presented a system that provides distributed access control and enforcement for Web Services. We have addressed the integration of a separate *Privilege Management Infrastructure* by defining mechanisms for the semantic description of the components of the PMI. These mechanisms allows us to seamlessly integrate the authorization entities in our infrastructure. We have introduced the *Semantic Policy Language*, an XML-based policy definition language designed to specify policies in a simple way and to facilitate

semantic policy validation. SPL specifications are modular and can be composed without ambiguity. Finally, we also have addressed the problem of the association of policies to Web Services in a dynamic, flexible and automated way.
An important feature is the extensive use of XML metadata technologies, which facilitates the security administration and enable interesting functionalities of the system such as the contextual semantic validation of policies. Metadata are applied at different levels in our proposal. On one hand, access control policies benefit from metadata for its creation and semantic and contextual validation. Likewise, Web services have metadata associated that are used for the dynamic policy assignment and parameter instantiation. Additionally, metadata are used for the specification and acquisition of certification rules. On the other hand, metadata is an essential tool for the integration of the external PMI [18] in the access control system.
Achieving use control for some kinds of Web services requires the use of software protection mechanisms. In this case, ongoing work is focused on the extension of solutions proposed in [19-21] to provide full secure use control for such Web services.

REFERENCES

[1] W3C Note, "SOAP: Simple Object Access Protocol 1.1," 08 May 2000.
[2] Organization for the Advancement of Structured Information Standards. Universal Description, Discovery and Integration (UDDI). http://www.uddi.org/specification.html
[3] World Wide Web Consortium. Web Services Description Language (WSDL) 1.1 W3C Note 15 March 2001. http://www.w3c.org/TR/wsdl
[4] WS-Security. Web Services Security. 2002. http://msdn.microsoft.com/
[5] ITU-T Recommendation X.509: Information Technology - Open systems interconnection - The Directory: Public-key and attribute certificate frameworks, March 2000.
[6] World Wide Web Consortium, Resource Description Framework (RDF Model Theory). 2002. http://www.w3.org/TR/rdf-mt/
[7] World Wide Web Consortium, XML?Schema, http://www.w3.org/XML/Schema
[8] Thompson, M., et al., Certificate-based Access Control for Widely Distributed Resources. Proc. of the Eighth USENIX Security Symposium. pp. 215-227. 1999.
[9] Chadwick, D. W. An X.509 Role-based Privilege Management Infrastructure. Business Briefing. Global Infosecurity 2002. http://www.permis.org/
[10] Sandhu, R.S., E.J. Coyne, H.L. Feinstein and Youman, C.E. Role-Based Access Control Models. IEEE Computer, 1996. 29(2): pp. 38-47.
[11] Yagüe, M.I. On the suitability of existing access control and DRM languages for mobile policies. University of Málaga. Department of Computer Science. Technical Report nb. LCC-ITI-2002-10. 2002.
[12] Bertino, E., Castano, S., Ferrari, E. Securing XML documents with Author-X. IEEE Internet Computing, 5(3):21-31, May/June 2001.
[13] Damiani, E., De Capitani di Vimercati, S., Paraboschi, S., Samarati, P. A Fine-Grained Access Control System for XML Documents. In ACM Transactions on Information and System Security (TISSEC), vol. 5, n. 2, May 2002, pp. 169-202.
[14] Damiani, E., De Capitani di Vimercati, S., Paraboschi, S., Samarati, P. Securing SOAP E-Services. In Int. Journal of Information Security, vol. 1, no. 2, pp. 100-115. 2002.
[15] World Wide Web Consortium, XML Path Language (XPath) Version 1.0. 1999. http://www.w3.org/TR/xpath/
[16] Kudo, M., Hada, S. XML Document Security based on Provisional Authorisation. In Proc. of the 7th ACM Conference on Computer and Communications Security. 2000.
[17] World Wide Web Consortium. Document Object Model (DOM) Level 1 Specification. http://www.w3.org/TR/REC-DOM-Level-1/
[18] López, J., Maña, A., Ortega, J., Pimentel, E., Troya, J.M., Yagüe, M.I. Integrating PMI services in CORBA Applications. To appear in Computer Standards and Interfaces Journal. Elsevier Science. 2003.
[19] López, J., Maña, A. and Yagüe, M.I. XML-based Distributed Access Control System. Proc. 3rd Int. Conference on Electronic Commerce and Web Technologies. Springer-Verlag. LNCS 2455. 2002.
[20] Yagüe, M.I., Maña, A., López, J., Pimentel, E., Troya, J.M. Secure Content Distribution for Digital Libraries. Proc. Int. Conference On Asian Digital Libraries 2002. Springer-Verlag. LNCS 2002.
[21] López, J., Maña, A., Pimentel, E., Troya, J.M., Yagüe, M.I. Access Control Infrastructure for Digital Objects. Proc. Int. Conference On Information and Communications Security 2002. Springer-Verlag. LNCS 2513. 2002.