Access Control and Data Integrity for DAML+OIL and DAML-S*

Grit Denker
SRI International
Menlo Park, California, USA
denker@csl.sri.com

Abstract

DAML+OIL is a markup language that allows the annotation of web pages to indicate their meaning. Languages like DAML+OIL are at the core of the Semantic Web, the future Internet in which machine-processable metadata describes the meaning of web resources. Semantic annotations will enable software agents to extract web content more quickly than it is possible with current techniques. Given that security plays an increasingly important role in today's society, it will also be vital to the success of the Semantic Web. We propose a security ontology for DAML+OIL that handles access control restrictions and data integrity measures of web resources. We also provide hooks into the existing standard of XML Signatures. We show how security annotations can be applied to the specification of web services in DAML-S. Finally, we provide a wrapper for the Security Assertions Markup Language (SAML) Protocol. SAML is currently under development to become a standard for XML-based security frameworks.

Keywords: Semantic Web, DAML+OIL, DAML-S, access control, data integrity, security, services, SAML, XML Signature
Word count: approx. 4700

1 Introduction

Though today's Internet is a vast information resource, its lack of structure and metadata make it difficult to extract the desired information in a reasonable time. The Semantic Web is a vision of a future Internet in which web resources are enriched with machine-processable metadata, which describes their meaning. This will enable software agents or search engines to interpret and extract web content much more quickly and precisely than it is possible with current techniques, such as keyword search. In the Semantic Web, resources can be evaluated with respect to their appropriateness for the given query, which in turn will lead to greater efficiency of web resource allocation, despite the daily expansion of web space. The DARPA Agent Markup Language (DAML [6]) is a language that allows the annotation of web pages to indicate their meaning. Its latest version, DAML+OIL [2], is a combination of DAML with the European OIL (Ontology Inference Layer) [21], which in turn is built on top of the description logics [13].

In this paper, we are bridging the gap between the Semantic Web and security. Information security plays an increasingly critical role in society. Over the last decade there has been considerable progress in the field of security, ranging from theoretical foundations and models, over languages, methods, and tools, to applications and security system development. Given the increased importance of the World Wide Web for business, industry, finance, education, government and other sectors, security will play a vital role in the

*Supported by the Defense Advanced Research Projects Agency through the Air Force Research Laboratory under Contract F30602-00-C-0168.

success of the Semantic Web. It is essential that we have tools and techniques in place that will allow to store, maintain, and process information of the Semantic Web in ways that meet security requirements such as authentication, authorization, and data integrity among others.

We propose language constructs for expressing security aspects of web pages and show how such annotations can be exploited in web service applications. We emphasize techniques to describe access control restrictions and data integrity. Our effort is embedded in the DAML project for all proposed language elements are defined in DAML+OIL.

Section 2 introduces an ontology for access control and data integrity and the various cryptographic techniques to ensure them, such as authentication, checksums and digital signatures. We illustrate the use of our ontologies by defining well-known protection mechanisms for web pages such as login-password or certificates. Moreover, we link our ontology to the existing standard for XML signatures [26, 3] that, among others, provides schema definitions for key information of well-known key management frameworks such as X509 [11, 10], PGP [27] and others.

In Section 3 we address security aspects as part of web services. An important feature of the semantic web is that location, selection, and execution of services that are based on semantic description become amenable for automation. DAML-S [5, 7, 23] is a language for describing capabilities of services and procedures available on the Web. Security will become a central aspect of emerging web services. Quality, reliability, access control, and privacy issues of web applications need to be addressed. We apply our security ontology to authorization and authentication requests that are specified as processes in DAML-S.

Currently, OASIS, the Organization for the Advancement of Structured Information Standards, is leading an effort on the Security Assertions Markup Language (SAML) [18, 19], an XML-based security standard for exchanging authentication and authorization information. SAML defines a request process that handles the exchange of SAML assertions in order to establish trust relations as a basis for security related decisions. In section 4 we outline an embedding of SAML into DAML-S in order to take advantage of the rich SAML framework.

Section 5 concludes with a brief summary and future work.

2 A Security Ontolgy for DAML+OIL

In the Semantic Web of the future, web resources can be annotated with respect to their content and meaning. One particular application of semantic mark-ups is the specification of security information concerning access restrictions of web pages or reliability assertions about the content of web pages. Our ontology comprises notations to mark-up web resources with respect to access control and data integrity. A web resource may be access restricted. In order to be granted access to the resource, the requesting agent or user has to pass an authorization test. Traditionally, authentication techniques are used to get authorized access. A web resource may have assertions about the integrity of its content. Data may have been inadvertently or deliberately altered. The use of cryptographic techniques assures the integrity of stored or received data.

We are defining a security ontology in DAML+OIL [2] that allows to annotate web resources with respect to access control and data integrity. Our ontology taps into current standards for the XML signature syntax [3].

2.1 Access Control

The process of establishing and verifying the identity of a requesting party in a web application, so-called authentication, is often the basis of the decision whether access is granted or not. Authentication is based on some token, also called credential, that the requesting party would know or have. Credentials can employ different well-know authentication techniques such as name-passphrase login, public and private keys or certificates. Our goal is it to be able to specify access control restrictions of web pages or web services that use authentication as a requirement for authorized access. The following are examples how we envision access control annotations in DAML+OIL. The annotations are meant for web pages that (1) have no access restrictions, (2) use a user-password login, and (3) accept X509 certificates to authenticate a user.

```
<AccessControl rdf:ID="None">
  <requiredCredential>NoCredential</requiredCredential>
</AccessControl>

<AccessControl rdf:ID="AccessByLogin">
  <requiredCredential>Login</requiredCredential>
</AccessControl>

<AccessControl rdf:ID="AccessByX509v3Certificate">
  <requiredCredential>X509v3Certificate</requiredCredential>
</AccessControl>
```

The above annotations make use of a security ontology that defines at the top-level a class `AccessControl` with a property `requiredCredential`. In order to restrict the range of the property only to classes that describe credentials, we defined the metaclass `ClassOfCredentials`. [1]

```
<daml:Class rdf:ID="ClassOfCredentials">
  <rdfs:comment>
    This is a metaclass in the sense that it is a subclass of daml:class.
    It will contain as instances all classes that have to do with credentials.
    The metaclass allows us to define a property for the AccessControl class
    that has as range instances of this metaclass.
  </rdfs:comment>
  <rdfs:subClassOf
   rdf:resource="http://www.w3.org/2001/03/daml+oil#Class"/>
</daml:Class>

<daml:ObjectProperty rdf:ID="requiredCredential">
  <rdfs:comment>
    requiredCredential is an ObjectProperty whose range is ClassOfCredentials
  </rdfs:comment>
  <rdfs:range rdf:resource="#ClassOfCredentials"/>
</daml:ObjectProperty>

<daml:Class rdf:ID="AccessControl">
  <rdfs:comment>
    AccessControl is a class that has at least one "requiredCredential"
    specified. Since the class "Credentials" includes "NoCredential",
    this allows the annotation of web resources that do not have access
    restrictions. In the case that several credentials are defined,
    they present a choice of credentials that can be used to get access
```

[1] We will use the following XML namespace definitions throughout the document: `xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns"`, `xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema"`, `xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"`, `xmlns:daml="http://www.daml.org/2001/03/daml+oil"`, and `xmlns:dsig="http://www.w3.org/2000/09/xmldsig"`

```
    to a web resource.
  </rdfs:comment>
  <rdfs:subClassOf>
    <daml:Restriction daml:minCardinality="1">
      <daml:onProperty rdf:resource="#requiredCredential"/>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
```

Different kinds of credentials are defined as instance of the top-level metaclass `ClassOfCredentials`.

```
<ClassOfCredentials rdf:ID="Credential">
</ClassOfCredentials>

<ClassOfCredentials rdf:ID="NoCredential">
  <rdfs:subClassOf rdf:resource="#Credential"/>
</ClassOfCredentials>

<ClassOfCredentials rdf:ID="Certificate">
  <rdfs:subClassOf rdf:resource="#Credential"/>
</ClassOfCredentials>

<ClassOfCredentials rdf:ID="X509v3Certificate">
  <rdfs:subClassOf rdf:resource="#Certificate">
</ClassOfCredentials>

<ClassOfCredentials rdf:ID="Key">
  <rdfs:subClassOf rdf:resource="#Credential"/>
</ClassOfCredentials>
```

`NoCredential`, `Certificate`, and `Key` are all subclasses of the credential class. Instances of `X509v3Certificate` are special kinds of certificates. The `subClassOf` relation allows to hierarchically structure classes of credentials. It also allows for future extensions if one would like to add new kind of credentials that are based on new technologies like voice or biometric (fingerprints, etc.).

Some credential classes come along with properties. For instance, a login credential always requires a user name and a passphrase.

```
<daml:DatatypeProperty rdf:ID="name">
  <rdfs:comment>
    name is a DataTypeProperty whose range is xsd:string.
  </rdfs:comment>
  <rdf:range
   rdf:resource="http://www.w3.org/2000/10/XMLSchema#string"/>
</daml:DatatypeProperty>

<daml:DatatypeProperty rdf:ID="passphrase">
  <rdfs:comment>
    passphrase is a DataTypeProperty whose range is xsd:string.
  </rdfs:comment>
  <rdf:range rdf:resource="http://www.w3.org/2000/10/XMLSchema#string"/>
</daml:DatatypeProperty>
```

```
<ClassOfCredentials rdf:ID="Login">
  <rdfs:comment>
    A login has exactly one user-name and one passphrase.
  </rdfs:comment>
  <rdfs:subClassOf rdf:resource="#Credential"/>
  <rdfs:subClassOf>
    <daml:Restriction daml:Cardinality="1">
      <daml:onProperty rdf:resource="#name"/>
    </daml:Restriction>
    <daml:Restriction daml:Cardinality="1">
      <daml:onProperty rdf:resource="#passphrase"/>
    </daml:Restriction>
  </rdfs:subClassOf>
</ClassOfCredentials>
```

In order to establish a relationship between a document or resource and its access control declaration, we define a property `usesAccessControl` on resources.

```
  <rdf:ObjectProperty rdf:ID="usesAccessControl">
    <rdfs:comment>
    A Resource uses an access control technique, such as Login or Certificates.
    <rdfs:domain rdf:resource="rdfs:Resource"/>
    <rdfs:range rdf:resource="#AccessControl"/>
  </rdf:ObjectProperty>
```

2.2 Tapping into XML Signature

So far we presented ontology elements that describe access control rights at a very abstract level. Some applications might want to express more details about the access restrictions of a web page. For instance, a web page that uses certificates, only accepts X509 version3 certificates [11], currently among the most wide-spread certificates used in web browsers. Or even more specifically, a web page only accepts X509 certificates that have been issued by a particular certification authority such as VeriSign or Thawte. Such information will be extremely helpful in directing software agents to web resources that are available to them. A software agent that searches for a particular web service could be equipped with a collection of certificates. Whenever the agent encounters a service that satisfies other functional aspects of its request, it may do some simple computation to conclude whether the service will be available.

In order to allow such detail in access control descriptions, we decided to support a link to the XML Signature Syntax and Processing Rules [3]. This standard that was jointly proposed by IETF and W3C defines how signatures on any digital content can be created and represented in XML. Digital signatures are often used to ensure that data has not been subject to alterations, inadvertently caused by faulty transmission channels or deliberately caused by intrusive behaviors. In particular, the XML Signature framework includes references to well-known cryptographic algorithms and key management systems. A `KeyInfo` element in an XML `Signature` element indicates the keys that need to be used in order to validate the signature. XML Signature has syntax elements to define key data related to X509 certificates [3], to PGP public key pairs [27], and SPKI public key pairs [22]. We can exploit those structures by defining specific instance of the class `ClassOfCredentials` in our security ontology. For instance, a certificate that complies with the syntax of the XML Signature X509 key data element can be defined as follows:

```
<daml:ObjectProperty rdf:ID="associatedData">
  <rdfs:comment>
```

```
    associatedData is an Object property without range definition.
  </rdfs:comment>
</daml:ObjectProperty>

<ClassOfCredentials rdf:ID="XMLSignatureX509v3Certificate">
  <rdfs:subClassOf rdf:resource="#Certificate">
  <daml:Restriction daml:cardinality="1">
    <daml:onProperty rdf:resource="#associatedData"/>
    <daml:hasClass rdf:resource="http://www.w3.org/2000/09/xmldsig#X509Data"/>
  </daml:Restriction>
</ClassOfCredentials>

<!-- similar for PGPData or SPKIData elements -->
```

Now we can annotate a web page that only grants access to users that can present an X509v3 certificate that was signed by VeriSign. We use the `X509Data` element of XML Signature and its pointer `IssuerSerial` that defines the name of the certificate authority that issued the certificate.

```
<AccessControl
rdf:ID="AccessByXMLSignatureX509v3CertificateIssuedByVerisign">
  <requiredCredential>
     <ClassOfCredentials rdf:resource="#XMLSignatureX509v3Certificate">
       <associatedData>
         <dsig:X509Data>
           <dsig:X509IssuerSerial>
             <dsig:X509IssuerName>VeriSign</dsig:X509IssuerName>
           </dsig:X509IssuerSerial>
         </dsig:X509Data>
       </associatedData>
     </ClassOfCredentials>
  </requiredCredential>
</AccessControl>
```

2.3 Data Integrity

In many applications it is crucial that information that is retrieved can be trusted with respect to its integrity. That means, a requesting client would like to get an assurance that the content of the web page that is being retrieved is in fact what the purported author of the web page intended to state. There exists several techniques that assure data integrity. Checksums or cyclic redundancy checks are meant to detect errors that were caused by random faults. In order to reveal errors that are caused by malicious behavior, one needs to used cryptographically secure techniques such as message authentication codes (MACs) or digital signatures on hash values.

Our security ontology contains top-level classes for most of the well-known techniques. The approach is very similar to annotating web pages with access control requirements. We provide a class `DataIntegrity` that has a property `usesDataIntegrityTechnique` which ranges over all classes of data integrity techniques.

```
<daml:Class rdf:ID="ClassOfDataIntegrityTechniques">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2001/03/daml+oil#Class"/>
</daml:Class>
```

```
<daml:ObjectProperty rdf:ID="usesDataIntegrityTechnique">
  <rdfs:range rdf:resource="#ClassOfDataIntegrityTechniques"/>
</daml:ObjectProperty>

<daml:Class rdf:ID="DataIntegrity">
  <rdfs:subClassOf>
    <daml:Restriction daml:minCardinality="1">
      <daml:onProperty rdf:resource="#usesDataIntegrityTechnique"/>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
```

The commonly known techniques are defined as instances of the metaclass `ClassOfDataIntegrityTechniques`. We show only a part of the whole specification.

```
<ClassOfDataIntegrityTechniques rdf:ID="None">
</ClassOfDataIntegrityTechniques>

<ClassOfDataIntegrityTechniques rdf:ID="Checksum">
  <rdfs:comment>
    There may be different ways of computing a checksum, cryptographically
    secure techniques such as MACs or hash functions and digital signature,
    and insecure techniques such as XOR or CRC.
  </rdfs:comment>
</ClassOfDataIntegrityTechniques>

<ClassOfDataIntegrityTechniques rdf:ID="SecureChecksum">
  <rdfs:subClassOf rdf:resource="#Checksum"/>
</ClassOfDataIntegrityTechniques>

<ClassOfDataIntegrityTechniques rdf:ID="XMLSignature">
  <rdfs:subClassOf rdf:resource="#SecureChecksum"/>
  <daml:sameClassAs
   rdf:resource="http://www.w3.org/2000/09/xmldsig#Signature"/>
</ClassOfDataIntegrityTechniques>
```

Note that we provided a link into the class `dsig:Signature` of the XML Signature standard. An XML `Signature` element may contain comprehensive information concerning a wide variety of cryptographic techniques, including signature methods (e.g., hashing, public key algorithms, MACs, padding, etc.), a sequence of references to digital objects that have been signed and digest methods that have been used. This opens the door to the body of work of the W3C working group on digital signatures.

Information on a web page that expresses that no assertions are made with respect to its content may lead a semantic web-enabled agent to disregard its content, independent from whether pieces of its description match other parts of the query. Such a web page is annotated as follows:

```
<DataIntegrity rdf:ID="None">
  <usesDataIntegrityTechnique>None</usesDataIntegrityTechnique>
</DataIntegrity>
```

Note that data integrity mark-ups themselves are not reliable if they are not signed. Thus, if one wants to achieve that also the assertion about the used integrity techniques can be trusted, it has to be signed for instance as part of an XML signature element.

3 Security for DAML-S services

DAML-S [5] is a language that supports a set of basic classes and properties (defined in DAML+OIL) for declaring and describing services. This work has its roots in previous work about web services [25, 8, 15, 16, 9]. Its top-level class is `Service`. A service presents a `ServiceProfile` (*what it does*), is described by a `ServiceModel` (*how it works*), and supports `ServiceGroundings` (*how it is used*).

A particular kind of service model has been proposed by the DAML service coalition [7], a so-called process model. A process may have parameters, preconditions, and effects that describe the constraints under which the service can be executed and what results it will produce. A process can be an atomic process or a structured composite process. Subsequently, references to the process specification of DAML-S refer to the following namespace definition: `xmlns:process="http://www.daml.org/services/daml-s/2001/05/Process.daml"`.

In the security setting, the process of authenticating a user in order to decide whether access is granted or not can be understood as a subprocess of a composite process. Access control descriptions become parameters of a DAML-S authorization process. In the following we illustrate the application of access control mark-ups in DAML services with the help of an example.

For the WWW2002 conference a web page has been set-up to handle activities such as authors submitting papers or program committee members refereeing papers. For instance, an administration service for authors allows to create a user account, update a user's record and submit papers. We show the top-level specification of such a process, called WW2002AuthorAdministration, that would be associated with a WWW2002 service.

```
<daml:Class rdf:ID="WWW2002AuthorAdministration">
  <rdfs:comment>
    Top-level specification of an expanded Administration Service for
    authors of the WWW2002 conference.
  </rdfs:comment>
  <rdfs:subClassOf rdf:resource="process:Process"/>
  <rdfs:subClassOf>
    <process:Sequence>
      <daml:item>
        <daml:Class rdfs:about="#TestAuthorization"/>
      </daml:item>
      <daml:item>
        <process:Alternative>
          <daml:item>
            <daml:Class rdfs:about="#ShowUserRecord"/>
          </daml:item>
          <daml:item>
            <daml:Class rdfs:about="#EditUserRecord"/>
          </daml:item>
          <daml:item>
            <daml:Class rdfs:about="#CreatePaperSubmission"/>
          </daml:item>
        </process:Alternative>
      </daml:item>
    </process:Sequence>
  </rdfs:subClassOf>
```

The above specification assumes appropriate definitions of processes `TestAuthorization`, `ShowUserRecord`, `EditUserRecord`, and `CreatePaperSubmission`. After authorization has been tested for a particular user, she has the choice of manipulating her data records or submitting a paper. We will detail the specification of the TestAuthorization subprocess, using the classes for describing access control restrictions as defined in section 2. Testing authorization is a subprocess that takes as a parameter a description of the access control defined for the webpage. Moreover, the credentials presented by the service-requesting agent are input to the authorization process. The effect of this process is a decision as to whether the user is authorized or not.

```
<!-- The Basic TestAuthorization Process -->
<daml:Class rdf:ID="TestAuthorization">
  <rdfs:subClassOf rdf:resource="process:Process"/>
</daml:Class>

<daml:ObjectProperty rdf:ID="withAccessControlRestriction">
  <daml:subPropertyOf rdf:resource="process:parameter"/>
  <daml:range rdf:resource="#AccessControl"/>
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="usingCredential">
  <daml:subPropertyOf rdf:resource="process:input"/>
  <daml:range rdf:resource="#Credential"/>
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="resultingDecision">
  <daml:subPropertyOf rdf:resource="process:output"/>
  <daml:range rdf:resource="xsd:boolean/>
</daml:ObjectProperty>
```

An authorization processs has at least one kind of access control restriction as a parameter. The access control restriction parameter determines the kind of access control that is being tested for (e.g., test for login, certificate or the like). Moreover, the authorization process gets as input at least one credential that will be used in determining whether access is granted. The resulting decision is uniquely determined, access is either granted or not.

```
<daml:Class rdf:about="TestAuthorization">
  <rdfs:subClassOf>
    <daml:Restriction daml:minCardinality="1">
      <daml:onProperty rdf:resource="#withAccessControlRestriction"/>
    </daml:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <daml:Restriction daml:minCardinality="1">
      <daml:onProperty rdf:resource="#usingCredential"/>
    </daml:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <daml:Restriction daml:cardinality="1">
      <daml:onProperty rdf:resource="#resultingDecision"/>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
```

Following up with the WWW2002 web administration page which uses logins to determine a user's eligibility to access certain data, we can define an authorization test for a particular user as follows (assuming `MyUserName` and `MyPassword` have been properly defined elsewhere):

```
<TestAuthorization rdf:ID="WWW2002AuthorizationTestForMe">
  <withAccessControlRestriction>AccessByLogin</withAccessControlRestriction>
  </usingCredential>
    <Login><name>MyUserName</name>
           <passphrase>MyPassword</passphrase>
    </Login>
  </usingCredential>
  <resultingDecision>true</resultingDecision>
</TestAuthorization>
```

4 A Wrapper for SAML

Security Assertions Markup Language (SAML) [18, 19] is a standardization effort for defining an XML framework for exchanging authentication and authorization information. Previous work towards XML-based security specifications such as S2ML and AuthXML was taking into consideration from the early days of SAML. SAML defines a syntax for making security assertions and a basic request/response protocol to exchange such assertions. Bindings to several widely used protocols such as HTTP and SOAP have been proposed [20].

In the following we propose a link between DAML-S services and the SAML protocol. Additionally, the DAML-S TestAuthorization class as defined in the previous section can be extended to include authorization queries defined for SAML. All SAML notations referred to in the following use the following namespace definition: `xmlns:saml=wc"http://www.oasis-open.org/committees/security/docs/draft-sstc-schema-assertion-19.xsd"`.

The SAML protocol assumes SAML-aware agents that use a `Request` element as input of the request process and a `Response` element as its output. The following wraps DAML-S around SAML.

```
<daml:ObjectProperty rdf:ID="inputSAMLRequest">
  <daml:subPropertyOf rdf:resource="process:input"/>
  <daml:range rdf:resource="saml:Request"/>
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID="outputSAMLResponse">
  <daml:subPropertyOf rdf:resource="process:output"/>
  <daml:range rdf:resource="saml:Response"/>
</daml:ObjectProperty>

<daml:Class rdf:about="SAMLProtocol">
  <rdfs:subClassOf rdf:resource="process:Process"/>
  <rdfs:subClassOf>
    <daml:Restriction daml:cardinality="1">
      <daml:onProperty rdf:resource="#inputSAMLRequest"/>
    </daml:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <daml:Restriction daml:cardinality="1">
      <daml:onProperty rdf:resource="#outputSAMLResponse"/>
```

```
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
```

Queries are particular kinds of requests in SAML. A SAML AuthorizationQuery element specifies all parameters of an authorization query. Such an element could be used as the input parameter of a process that test authorization using SAML data elements.

```
<daml:Class rdf:ID="SAMLAuthorizationQuery">
  <rdfs:subClassOf rdf:resource="saml:AuthorizationQuery"/>
</daml:Class>

<daml:ObjectProperty rdf:ID="inputQuery">
  <daml:subPropertyOf rdf:resource="process:input"/>
  <rdfs:range rdf:resource="#SAMLAuthorizationQuery"/>
<daml:ObjectProperty>

<daml:Class rdf:about="SAMLTestAuthorization">
  <rdfs:subClassOf rdf:resource="#TestAuthorization"/>
  <rdfs:subClassOf>
    <daml:Restriction daml:minCardinality="1">
      <daml:onProperty rdf:resource="#inputQuery"/>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
```

5 Concluding Remarks

We have presented a security ontology for DAML+OIL that is extensible in that it allows the definition of further security aspects as well as new links to existing standards and frameworks. In particular, we defined an extension point to the work of the IETF and W3C XML Signature working group. With the given security framework for DAML+OIL most common access control restrictions and data integrity measures of existing web pages could be described. We also applied access control description in specification of DAML services.

The implementation of DAML+OIL-aware software agents that exploit security and service annotations is an interesting topic for future work. We are currently implementing a DAML+OIL-aware OAA agent. OAA [17] is a framework for integrating a community of heterogeneous software agents in a distributed environment. In the future we will enrich our DAML+OIL OAA agent with specifics about DAML security and services so that an agent equipped with a query and credentials, traversing the Web, can readily perform tasks such as deciding whether a web resource meets the data integrity requirements of the requesting party, or whether access control restrictions are likely to be met by the credentials.

In the current framework, we focus on access control restrictions that are used for authorization decisions. Those decisions are usually embedded into the broader context of policies that define restrictions a party has to comply to. It would be worthwhile to investigate the appropriateness of DAML+OIL for specifying policies. A formal definition of policies in DAML+OIL would have several advantages. The uniform representation becomes amenable to formal verification. Forward reasoning in a theorem prover such as SNARK [24] could be used to detect inconsistencies in a (set of) policies. This approach is similar to the SNARK verification of the DAML+OIL semantics that revealed some problems. Uniform representation of policies would also bring the automation of policy negotiation within the realms of possibility.

Logics for trust management is another topic of future investigations. There exists already a large body of work on trust management, e.g., [1, 4, 14] to name a few. Recent work on frameworks for distributed trust management in a multi-agent system [12] and existing authentication and delegation logics have to be taken into consideration for future logical extensions of DAML+OIL. A theory for reasoning about trust relations will be at the core of decision procedures for web agents that search for reliable information.

Acknowledgments

The author would like to thank David Martin and Richard Waldinger whose comments on an earlier version helped to improve the paper. It is a highly rewarding experience to cooperate with the members of the DAML team at SRI, Jerry Hobbs, Pat Lincoln, David Martin, Srini Narayanan, and Richard Waldinger, on the diverse aspects of web services and security.

References

[1] M. Abadi, M. Burrows, B. Lampson, and G. Plotkin. A Calculus for Access Control in Distributed Systems. *ACM Transactions on Programming Languages and Systems*, 15:706–734, 1993.

[2] Agent Markup Language Committee. DAML+OIL. `http://www.daml.org/2001/03/daml+oil.daml`, March 2001. See `http://www.daml.org/committee/` for committee members.

[3] M. Bartel, J. Boyer, B. Fox, B. LaMacchia, and E. Simon. XML-Signature Syntax and Processing Rules. `http://www.w3.org/TR/2001/PR-xmldsig-core-20010820/`, August 2001.

[4] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *Proc. 1996 IEEE Symposium on Security and Privacy*, pages 164–173. IEEE Computer Society, 1996.

[5] M. Burstein, J. Hobbs, O. Lassila, D. Martin, S. McIlraith, S. Narayanan, M. Paolucci, T. Payne, K. Sycara, and H. Zeng. DAML-S 0.5 draft release. `http://www.daml.org/services/daml-s/2001/05`.

[6] The DARPA Agent Markup Language (DAML). `http://www.daml.org`.

[7] DAML services. `http://www.daml.org/services`.

[8] G. Denker, J. Hobbs, D. Martin, S. Narayanan, and R. Waldinger. Accessing information and services on the DAML-enabled web. In S. Decker, D. Fensel, A. Seth, and S. Staab, editors, *2nd. Intern. Workshop on the Semantic Web SemWeb'2001, Workshop at WWW10, Hongkong, China*, May 2001. `http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-40/`.

[9] J. Hendler. Agents on the web. *IEEE Intelligent Systems, Special Issue on the Semantic Web*, 16(2):30–37, March/April 2001.

[10] R. Housley, W. Ford, W. Polk, and D. Solo. Internet X.509 Public Key Infrastructure. Certificate and CRL Profile. RFC 2459, IETF, January 1999. `http://www.ietf.org/rfc/rfc2459.txt`. Basis for [11].

[11] R. Housley, W. Ford, W. Polk, and D. Solo. Internet X.509 Public Key Infrastructure. Certificate and CRL Profile. Internet Draft , IETF, March 2000. `http://www.ietf.org/internet-drafts/draft-ietf-pkix-new-part1-11.txt`. Update to [10] defining X.509 Version 3 certificates and X.509 Version 2 CRLs.

[12] L. Kagal, S. Cost, T. Finin, and Y. Peng. A Framework for Distributed Trust Management. In *Proc. 2nd Intern. Workshops on Norms and Institutions in MAS, Autonomous Agents*, May 2001.

[13] P. Lambrix. Description logics references. http://www.ida.liu.se/labs/iislab/people/patla/DL/references.html.

[14] N. Li, B. Grosof, and J. Feigenbaum. A practically implementable and tractable delegation logic. In *Proc. 2000 IEEE Symposium on Security and Privacy (S&P'00)*, pages 27–42. IEEE Computer Society, 2000.

[15] S. McIlraith, T. Song, and H. Zeng. Mobilizing the semantic web with DAML-enabled web services. In S. Decker, D. Fensel, A. Seth, and S. Staab, editors, *2nd. Intern. Workshop on the Semantic Web SemWeb'2001, Workshop at WWW10, Hongkong, China*, May 2001. http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-40/.

[16] S. McIlraith, T. Song, and H. Zeng. Semantic web services. *IEEE Intelligent Systems, Special Issue on the Semantic Web*, 16(2):46–53, March/April 2001.

[17] Open Agent Architecture (OAA). http://www.ai.sri.com/oaa.

[18] OASIS Security Service TC (SSTC). Security assertion markup language (SAML). http://www.oasis-open.org/committees/security/.

[19] OASIS Security Service TC (SSTC). Security assertion markup language (SAML) - core assertion architecture. http://www.oasis-open.org/committees/security/docs/draft-sstc-core-19.p%df, 2001.

[20] OASIS Security Service TC (SSTC). Security assertion markup language (SAML)- bindings model. http://www.oasis-open.org/committees/security/docs/draft-sstc-bindings-%model-05.pdf, 2001.

[21] Ontology inference layer (oil). http://www.ontoknowledge.org/oil/.

[22] Simple Public Key Infrastructure (SPKI). http://www.ietf.org/html.charters/spki-charter.html.

[23] SRI DAML Service Team. Services on the semantic web. http://www.ai.sri.com/daml/services/.

[24] M. Stickel, R. Waldinger, and V. Chaudhri. *A Guide to SNARK*. SRI International, Artificial Intelligence Center, Menlo Park, CA, 2000. http://ai.sri.com/snark/tutorial.html.

[25] K. Sycara, J. Ju, M. Klusch, and S. Widoff. Dynamic service matchmaking among agents in open informatin environments. *ACM SIGMOD Record, Special Issue on the Semantic Interoperability in Global Information Systems*, 1999.

[26] IETF and W3C XMLSignature Working Group. http://www.w3.org/Signature/.

[27] P. Zimmermann. *The Official PGP User's Guide*. MIT Press, 1995.