ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 6, No. 3, 2011, *pp*. 177-194

Ontology-Based Access Control Model for Semantic Web Services

A. Mohammad [1], G. Kanaan [2], T. Khdour [3], S. Bani-Ahmad [4]

[1]The Arab academy for Banking and financial sciences, Damascus, Syria.
[2]The Arab academy for Banking and financial sciences, Amman, Jordan
[3,4] Al-Balqa Applied University, Salt, Jordan

(*Received March 7, 2011, accepted March 20, 2011*)

Abstract. Studies show that reducing the gap between security services and semantic web is important. In this paper we present an ontology-based access-control (OBAC) to support semantic web service. For that, security ontologies are developed to specify concepts and terms involved in this model. Our proposed access control model is expressive and general with these important features: (i) The use of ontology provides reasoning ability for access control decision making, and allows access control information to be automatically searched, queried and discovered. (ii) Our proposed model has a higher degree of interoperability compared to other approaches of access control mechanism. This is because of the nature of ontologies in providing semantic interoperability. (iii) Our proposed model is context sensitive; the constraint ontology represents different types of context constraint. (iv) Our proposed model is designed based on the widely accepted semantic web languages, Web Ontology Language (OWL) and Web Ontology Language for Service (OWL-S), therefore its implementation can be easily achieved by using already existing tools designed for working with these languages.

1. Introduction.

"The Semantic Web is not a separate Web but an extension of the current one, in which information is given well defined meaning, better enabling computers and people to work in cooperation. " (Berners-Lee et al., 2001). The functioning of the Semantic Web will depend on a number of technologies. Some important ones include XML, RDF and ontologies figure 1. Semantic Web services is an essential part of the Semantic Web development, it's vision is to describe Web services' capabilities and content in an unambiguous, computer-interpretable language and improve the quality and robustness of existing tasks such as Web service discovery and invocation (Mcilraith et al., 2001), bringing semantic to security services especially to access control play an important role in the integration between semantic web and security service, this integration in his turn play a major role in facilitating automatic reasoning for access control of Semantic Web services, The boom of the Internet led to the creation of ontology languages for exploiting the characteristics of the Web, Such languages are usually called Web-based ontology languages or ontology markup languages, From all of them, the ones that are being actively supported are now RDF, RDF Schema, ontology web language (OWL), and ontology web language for services (OWL-S) which was developed in the context of the work on Semantic Web Services, OWL-S defines an upper ontology for describing the properties and capabilities of Web services in OWL. It is intended to enable users and software agents to automatically discover, invoke, compose, and monitor Web resources offering services, under specified constraints, hoverer the shift from current Web to semantic aware environments such as the Semantic Web poses new security challenges especially in the field of access control. Access to web services on the Semantic Web can not be controlled in a safe way unless the access decision takes into account all the factors such as context constraints, heterogeneous of subjects and resources and automation of role assignment, Traditional access control models like MAC, DAC and RBAC fail to address these issues since they need to be accommodate to dynamic, open and distributed web service environment and then to be compatible with

[1] *E-mail address*: abdulgahfour@yahoo. com
[2] *E-mail address*: ghkanaan@aabfs. org
[3] *E-mail address*: khdour_thaer@hotmail. com
[4] *E-mail address*: sulieman@case. edu

the semantic web. in this paper our proposed access control takes into account all of the issues and utilize the web ontology language for services (OWL-S) to be compatible with semantic web, semantic web service (e. g. defined by OWL-S) is represented as process which has input, output, preconditions and effects, in this process, we introduce the semantic access control model as a condition figure 3, So In this paper, we present the ontology-based Access Control model (OBAC) which is an extension, or in other word an ontology representation to Context Sensitive Attribute and Task- Role Based Access Control (CSAT-RBAC) for web service application which utilize the characteristics of Role Based Access Control Model (RBAC), Attribute Based Access Control Model(ABAC) and Task Based Access Control Model(TBAC), however CSAT-RBAC has several considerable features that make it a suitable access control model for Semantic Web services, for example CSAT-RBAC is capable of handling dynamic and anonymous users and reducing security management tasks. CSAT-RBAC also supports a wide range of access control policies and provides fine-grained access control for Web service applications such as controlling parameters of the user request. However, the aim of this paper is to develop a semantically compatible access control model for Semantic Web services by providing ontological representations for the concepts and relations involved in the CSAT-RBAC model, each component of CSAT-RBAC is represented in a separate ontology such as credential ontology(on behave of subject), web service ontology (the protected resource), session ontology, constraint ontology, permission-role assignment ontology, user-role assignment ontology, and then we could integrate these separate ontologies to get the complete access control ontology which plays the role of a condition in semantic web service process. This will allow security services to be integrated with Semantic Web services. Such integration will facilitate automatic reasoning for access control of Semantic Web services.

The reminder of this paper is as follows; in Section 2 we discuss the preliminaries relevant to the Semantic Web. Section 3describes the related works on this topic, and section 4 states the fundamentals of OBAC by applying the ontology techniques to describe the access control model components (Each component of ERBAC will be defined in a separate ontology), the complete ontology-based access control introduced in section 5, Our proposed architecture for implementing the ontology-based access control(OBAC) model is presented in section 6 Finally, section 7 underlines some conclusions and future research lines.

2. Background

2.1. Semantic Web and Ontology

The aim of the Semantic Web initiative is to advance the state of the current Web through the use of semantics. More specifically, it proposes to use semantic annotations to describe the meaning of certain parts of Web information and, increasingly, the meaning of message elements employed by Web Services. For example, the Web site of a hotel could be suitably annotated to distinguish between the hotel name, location, category, number of rooms, available services and so forth. Such meta-data could facilitate the automated processing of the information on the Web site, thus making it accessible to machines and not primarily to human users, as it is the case today. The current web standard for semantic annotations is RDF and RDF Schema, and its extension OWL. Suitable annotations are useful for improving the accuracy of Web searches. The search engines can look for pages in which precise concepts from ontology are marked instead of collecting all pages in which certain, generally ambiguous, keywords occur. But the vision of the Semantic Web cannot be achieved solely by disambiguating and relating individual concepts. At least equally important is the integration or transformation of data structure elements. Besides some platform specific parts, data structures reflect in a contracted and simplified way how the designer perceives the possible states of affairs of the respective application. Ontologies allow for the formal specification of an application domain that can be shared by different systems. For instance, one system may distinguish hotels from guest houses. Another only refers to accommodation in general. Location may be given in coordinates, in metric distances or in walking distances to relevant fix points. Ontologies allow intelligent systems for mediating between these different forms to organize information. This ability constitutes a major prerequisite for the global access to Web services. A particularly interesting application of ontologies is the seamless integration of services, information systems and databases containing general knowledge. For instance, an ontology combining kinds of geographic units, kinds of tourist services and their relationships could be used to determine that Crete is an island in Greece, and therefore a Greek island and Heraklion a city on Crete. It would further describe that accommodations are immobile, and that hotels are kinds of accommodation. Such information would be crucial to establish a connection between a requester looking for accommodation on a Greek island, and a hotel advertisement specifying Heraklion as the hotel location (Grigoris et al., 2007).

DOI https://doi.org/2024-JICS-22676 | Generated on 2025-04-20 05:55:20

2.2. Semantic web services and web ontology languages

Adding semantics to represent the requirements and capabilities of Web services is essential for achieving automation in service discovery and execution. This need for semantics in Web services has led to the convergence of concepts from Web services and the Semantic Web community. These efforts have resulted in "Semantic Web services". Semantic Web services are Web services whose "properties, capabilities, interfaces, and effects are encoded in an unambiguous, and machine-interpretable form" (McIlraith et al., 2001).

The boom of the Internet led to the creation of ontology languages for exploiting the characteristics of the Web. Such languages are usually called Web-based ontology languages or ontology markup languages. Their syntax is based on existing markup languages such as HTML and XML whose purpose is not ontology development but data presentation and data exchange respectively. The most important examples of these markup languages that are being actively supported now are RDF, RDF Schema, OWL. Finally, in the context of the work on Semantic Web Services, a new ontology languages are being developed, named WSML (WSML, 2005) and OWL-S (McIlraith et al., 2001).

RDF was developed by the W3C (the World Wide Web Consortium) as a semantic-network based language to describe Web resources (Lassila et al., 1999). , the RDF Schema (Brickley et al., 2004) language was also built by the W3C as an extension to RDF with frame-based primitives. The combination of both RDF and RDF Schema is normally known as RDF(S). RDF(S) only allows the representation of concepts, taxonomies of concepts and binary relations. Some inference engines and query languages have been created, In RDF, every object (on the Web) is called a resource and comes along with a unique identifier, called URI. The most elementary building block of RDFS is a class, which defines a group of individuals that belong together because they share some characteristics.

Ontology Web Language (OWL) was proposed as a W3C recommendation in February 2004. OWL is built on top of RDF(S), extending its expressiveness with more primitives that allow representing complex expressions to describe concepts and relations. OWL is divided into three layers (OWL Lite, OWL DL and OWL Full), each of them providing different levels of expressiveness that can be used depending on the representation and inference needs of an ontology. OWL is based on the description logic language SHOIN(D+) and has several inference engines that can be used for constraint checking of concepts, properties and instances, and for automatic classification of concepts into hierarchies.

Web Service Modeling Language (WSML) (De Bruijn, 2006) is being developed in the context of the WSMO framework. (WSMO, 2005) This language is aimed to be used not only for representing ontologies, but also for representing Semantic Web Services; hence it contains many additional features that are not present in the languages aforementioned. Like OWL, it is divided in several layers. Each of these layers is based on different knowledge representation (KR) formalisms: description logic, logic programming and first order logic.

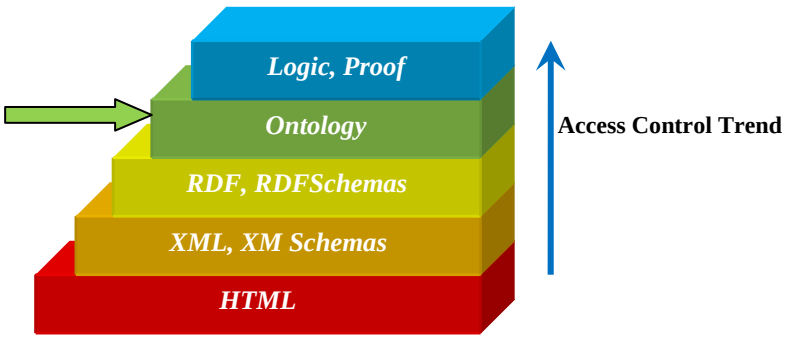


Fig 1. OBAC in the Semantic Web Stack

Finally, OWL-S defines an upper ontology for describing the properties and capabilities of Web services in OWL (OWL, 2004). It is intended to enable users and software agents to automatically discover, invoke, compose, and monitor Web resources offering services, under specified constraints. It defines high level constructs figure 1 such as a service profile: to represent the interfaces of services including inputs, outputs, preconditions and effects, a service (process) model to represent the details of inner working of a service, and

a service grounding to provide information about how to use a service. Whereas OWL-S profile model views a service as an atomic process, OWL-S service (process) model captures the state of a service as a complex interaction process. While OWL-S profile defines a model for describing the functional properties of a service via constructs such as inputs, outputs, preconditions and effects (sometimes referred to as IOPEs), OWL-S service model uses workflow constructs such as sequence, if-then-else, fork, repeat-until and so forth, to define a composite processes. OWL-S grounding model defines the necessary links to Web service industry standard WSDL to use its invocation model.

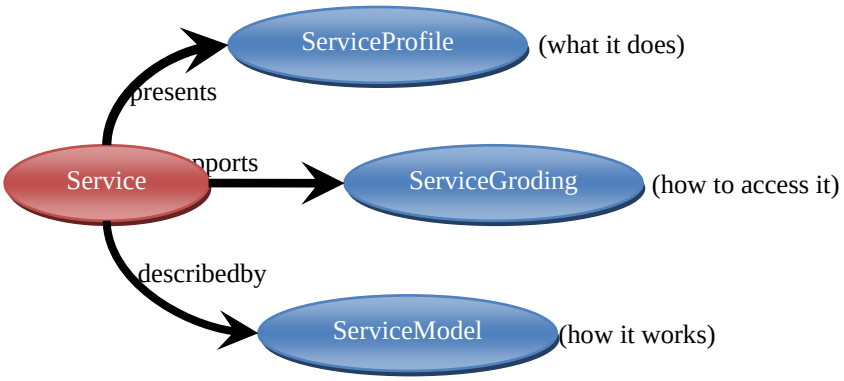


Figure 1: Top Level of the Service Ontology

3. The Role of Ontology-Based Access Control (OBAC) in Web Service Process

Before building our proposed ontology-based access control, it necessary to explain how the ontology-based access control integrate with the semantic web service, to understand the relationship between semantic web services and our proposed access control, we utilize the ontology web language for services (OWL-S) which describes web service by service(process) model (figure 1), The operation of a Web service is described in terms of a process model, which has Two main components, the *process ontology*, which describes a service in terms of its inputs, output, preconditions, effects and possible sub-processes; and the *process control ontology* which describes each process in terms of its state, including initial activation, execution and completion a Web service,

We introduce our proposed ontology-based access control as a condition that a user must fulfill in order to gain access to the Web service.

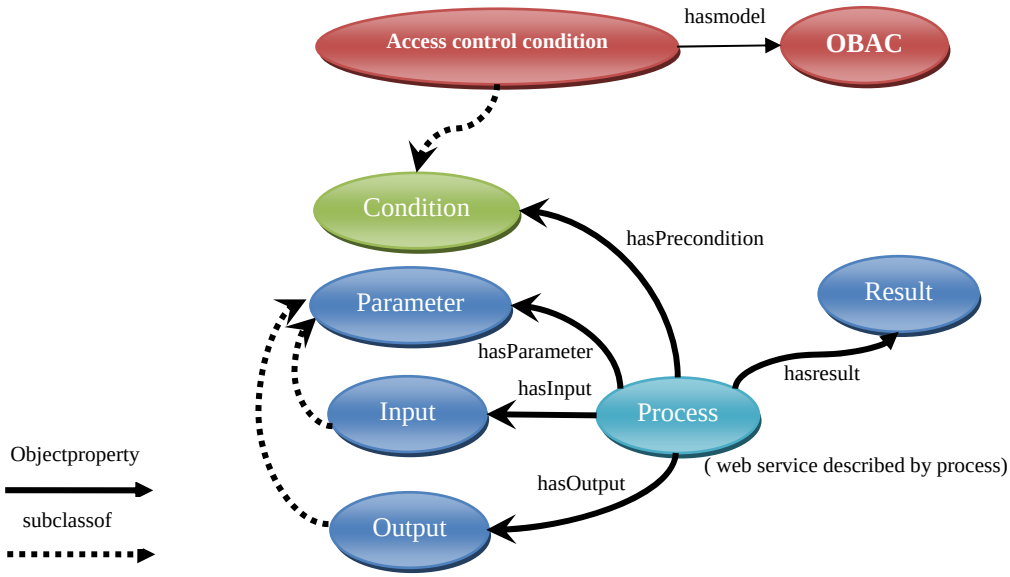


Figure 3. The Relation between Web Service Ontology and OBAC

As shown in figure 2, in OWL-S a web service may have many preconditions, each of which is a sub-

property of the property hasprecondition of a process. The property hasprecondition ranges over Condition class. To model ontology-based access control model as a condition, we introduce a class AccessControlCondition as subclass of the class Condition of OWL-S process model which ranges over our proposed ontology-based access control model.

4. Related work.

The researches in bringing semantic to access control have two parallel directions. One has focused on efforts to develop new access control models to meet the policy needs of real world application domains in parallel, and almost separate thread, researchers have developed semantic policy languages for access control.

(Bonatti et al., 2006) discussed important requirements for access control policies for semantic web, (Denker et al., 2003) Developed security annotations to describe security requirements and capabilities of web services providers and requesting agents. In the first layer of figure 1, (Prud'hommeaux, 2001) studied File–level access control systems, for Protecting HTML resources, in the next layer, several XML based approaches are proposed such as XML Role-Based Access Control by (Joshi et al., 2004), in the RDF layer, (Kagal et al., 2003) proposed policy languages based on Semantic Web languages like RDF and DAML+OIL and have developed a framework based on that policy. In the ontology layer (Qin et al., 2003) proposed a concept level access control model which Considers some semantic relationships in the level of concepts in the objects domain, This model was based on the subject-action-object paradigm and employed a Semantic Access Control Language (SACL). The subject-operation-object paradigm has some drawbacks in terms of security management, and it cannot support complex security policies. (Camera et al., 2003) developed a high level OWL-DL ontology that expresses the elements of a role based access control system and they built a domain-specific ontology that captures the features of a sample scenario. Then, they joined these two artifacts to take into account attributes in the definition of the policies and in the access control decision. In this work the researchers focus on the integration between access control ontology and a domain–specific ontology, in our work the focus is on the integration between the access control ontology and web service ontology, in addition we introduce web service task as a basic unit in the proposed model. (Agarwal et al., 2004). provided a credential-based access control for Semantic Web services using DAML-S and Simple Public Key Infrastructure (SPKI)/Single Distributed Security Infrastructure (SDSI). this paper only provided a semantic description of credentials without providing a comprehensive ontology-based description for all security concepts involved in Web service access control such as sessions and constraints. Furthermore, the proposed framework relies on traditional ACLs for access control, which is unable to model a variety of security policies and security concepts in a heterogeneous Web environment.

(Torsten et al., 2006) presented an approach for simplifying the specification and maintenance of attribute-based access control policies by extending the attribute management with an ontology-based inference facility. A semantic mapping between different attributes can be performed in ontology. This approach is based on the established XACML (Moses, 2005) standard and features thorough use of open standards like RDF and OWL in the semantic extension of the architecture. In (Torsten et al., 2006), the authors tries only to support the attribute based access control by using ontology-based inference facility, they didn't provide ontology representation for the model elements to be compatible with semantic web services. However In attribute based access control the permission assigned to user directly, in our model we place the role as intermediate between user attribute and permission this give the model more management power. (Finin et al., 2008) studied the relationship between the RBAC security model and OWL and represented the RBAC model in OWL, they described two possible approaches to RBAC in OWL, representing roles as classes and sub-classes in one approach and as attributes in an alternate approach, in this paper they studied RBAC and attributes based access control separately, so the didn't provide a generic access control model that take into account different access control situations, in our paper we propose a generic access control model by combining the users management ability in RBAC and the dynamic features of attribute based access control then we applying the ontology techniques on this proposed model. In addition Finn et al. studied the NIST RBAC without any extension related to context constraint; in our work we proposed top level context ontology as precondition in user role assignment ontology.

5. Analysis of the CSTA-RBAC Model for Web Services Applications:

Our proposed ontology-based access control model (OBAC) is an extension, or in other word ontology representation to context sensitive attribute and task role based access control (CSAT-RBAC) for web

service application which utilize the characteristics of Role Based Access Control Model (RBAC), Attribute Based Access Control Model(ABAC) and Task Based Access Control Model(TBAC), however in this section we summarize the important features and components of the CSAT-RBAC for web service applications before applying the ontology techniques on this model in the rest of the paper :

1. Task based access control: Task is used to solve the problem of object heterogeneity, Task is the basic unit of the new model; it is also used to represent permissions in authorization specifications. In Task based approach, only one access will be defined which reduce the complexity result from resource heterogeneity.
2. Constraint CSAT-RBAC: The notion of constraint can be used to describe any types of security policies. In our model, a constraint is applied to many areas of access control for various reasons. For example
   - Context constraint :we classified the context constraint as entity and environment constraint ;the Entity context contains are dynamic data which are user attributes, Task attributes, object attributes, and statically-defined relationships among users, objects, tasks and roles. on the other hand the Environment context which is an abstraction for system measurable data (time, location, or system load….). in addition, constraints can be applied to a variety of concepts such as role, Tasks, attributes of the Tasks and role and in permission assignment process.
   - Content-based access control constraint: the contents of the input and output parameters of a Task play an important role in access control decision making. The contents of input parameters are provided upon user requests, and it determines what resources are accessed at the back-end. The content of the output parameters may reveal confidential information to the user.
   - Least Privilege Principle – many access control models do not enforce the least privileged principle. The principle of least privilege states that users should only be granted privileges when they have justifiable needs. The proposed CSAT-RBAC model will strictly enforce the least privilege principle in role assignment, so that only the least privileged role is assigned to the user request.
3. Fine-grained access control: in CSAT-RBAC a Web service can be described by a number of Tasks, and an a Task has a number of parameters. However, a user may not have access to all Tasks of a Web service or all parameters of a Task. Partial access can be provided by introducing Web service instances and Task instances. The granularity of access control will not be limited to complete Web services or Tasks, but instances of Web services and instances of Tasks. Together with content-based access control, Web service access control can be performed in a fine-grained manner.
4. New features such as credential-based access control and dynamic role assignment are added to our model in order to address user heterogeneity and dynamicity in the Web environment :
   - Credential based access control: a possible alternative to authentication based access control is to use credentials provided by a trusted third party. It can enforce access control without knowing the user's identity. In Web service environments, the user is represented by user credentials. In our proposed model, a user credential must contain user capabilities for the target Web service. In order to access system resources, user capability will be used to activate different instances of the Web service. As a result, different users will observe different instances of the same Web service in different sessions. the credential will be represented in a separate ontology called credential ontology which work on behave of the user.
   - Dynamic role assignment: dynamic role assignment is dependent on all the above-mentioned features. In Web environments, access control decision making ought to be automated according to a variety of dynamic conditions. These conditions include user credentials, the context constraint including the contents of the input and output parameters of the WS-Task and a variety of constraints (least privilege principle can be seen as a type of constraint). Dynamic role assignment will save manual administrative work of specifying authorizations for each security subject against each security object, hence making security management simpler and more efficient.
5. CSAT-RBAC complements the security services that are used for communication level access control.
6. Components of CSAT-RBAC: CSAT-RBAC model is defined in terms of four model components: Core CSAT-RBAC, Hierarchical CSAT-RBAC, Static Separation of Duty Relations, and Dynamic Separation of Duty Relations, Core CSAT- RBAC includes sets of six basic data elements called users (USERS) defined by its attributes and capabilities, roles (ROLES), objects (OBS), operations (OPS), permissions (PRMS) and (Tasks), Besides the relations defined in the Core RBAC, the CSAT-RBAC model as a whole is fundamentally defined in terms of individual users being assigned to roles and permission being assigned to roles, The relations among of them are many-to-many. In addition, the Core CSAT-RBAC model includes a set of sessions (SESSIONS) where each session is a mapping onto an activated subset of roles that are assigned to the user.

In the reminder of this paper, each of these concepts can be represented by a separate ontology, and the aggregation of all these ontologies form the complete ontology-based CSAT-RBAC model. Ontologies can

be classified into different types such process ontology, role-based ontology and service ontology, each serving different purposes. the user is represented by the credential ontology which work on behave of the user, The heart of the CSAT-RBAC model is user-role assignment that ultimately determines access control outcomes. The user-role assignment can be represented by a process ontology which is described in terms of input, output, precondition and result. Another important relationship, permission-role assignment, defines which roles have what access rights. It differs to user-role assignment, as the latter is dynamic and are depend on a variety of conditions, whereas permission-role assignment is predefined statically by system administrators, and used to provide access control policies for the user-role assignment process. Thus, permission-role assignment can be represented by policy ontology.

6. Ontology-based Access Control Model Components:

Before we start to represent components of OBAC, the vocabulary and symbols used to describe the proposed access control model should be presented clearly, some of these vocabularies are used in OWL and OWLS, each of the components of the CSAT-RBAC model, such as user and role, can be represented by the term Class which is an OWL term. These security components (classes) are linked together by using the term ObjectProperty (OWL term). The attributes of the security components are described as DatatypePropery (OWL term). All properties will have domain (OWL term) and range (OWL term): the domain specifies the owner of the property, whereas the range specifies the target of the property. The term subClassOf (OWL term) is used to specify that a security component, a class, is a subclass of another security component (class), hence it inherits all the properties of the latter. We define a set of symbols to represent the above mentioned terms and relations as depicted in Figure 4, a predefined class: means this class has been defined by other ontologies in the model. By using OWL and OWL-S vocabulary, our ontology-based CSAT-RBAC model has been made Compatible with the Semantic Web.

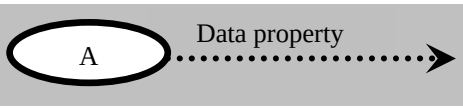
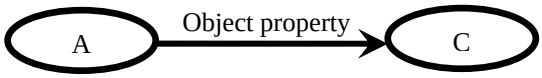
Symbol	Meaning
	Represents **a Class**
	Represents an **object property**
	Indicates Class A has **Data Property** f
	Indicates Class A has object Property b with **range** C
	Indicates Class D is **a subclass** of Class E
	Indicates F is a **predefined** class in the model

Figure 4: Ontology-based Access Control Symbols

In the following subsections we provide an ontology representation to the different components of the CSAT-RBAC.

6.1. Web service and Web Service Task Ontology

The description of Web service ontology will be based on the definition provided in the CSAT-RBAC model based on the process and profile model for describing and presenting semantic web service in OWL-S specifications (OWL-S, 2004) and. In the CSAT-RBAC model, a Web service is considered as a security object assigned to roles and accessed by users. Therefore, the proposed Web service ontology will only

involve security-related aspects. A Web service contains a set of properties that distinguishes one Web service from others such as service name, service type, service security level and composed of composed of property which ranges over the Task class. each Web service consists of a set of Tasks to carry out designated tasks. Each task represented as a process with input, output parameters, and a set of data properties as shown in figure 5, In the following we describe the main parts of the web service ontology.

service name property: refers to the name of the service that is being offered. It can be used as an identifier of the service.

Service security level : each task is assigned a security level property indicating its security importance.

serviceClassification : is used to specify the type of service provided, The value of the property is instance of classes specified in OWL ontologies of services, it defines a mapping from a Profile to an OWL ontology of services, such as an OWL specification of NAICS.

Composedof: A Web service application consists of a set of Web services**;** the composedof has the web service a domain and ranges over the task class.

Task : is the protected resource of this model, web service task is a function exposed by a web service to the outside world for the purpose of executing a certain user request on back-end resources, It is the only channel through which a web service interacts with users or other web services. , a Task has the following properties: tname, ttype, toperation, tresource, tsecurity level, hasinput and hasoutput as shown in Figure 5.

ttype :a task type may be determined by the purpose of a task.

resource and operation : each task performs one or more operations on back-end resources, therefore can be represented by operation and resource.

tsecuritylevel: each task is assigned a weight indicating its security importance. “security level” can be derived in different ways. The conventional way is based on the importance of the task. In general, a Task updating or

modifying the content of databases or files will be given the highest security protection, and hence will have more security level than a Task reading from or writing to databases or files. However, there is no single way of defining “securitylevel”. It always depends on the actual application.

Parameter: a class that has the following attributes: parameter name, parameter type and parameter value. Input and output classes inherit all three attributes. Every parameter has a type, specified using a URI. This is not the OWL class the parameter belongs to, but a specification of the class (or datatype) that values of the parameter belong to. It's convenient to identify parameters with what are called variables in SWRL, the language for expressing OWL Rules.

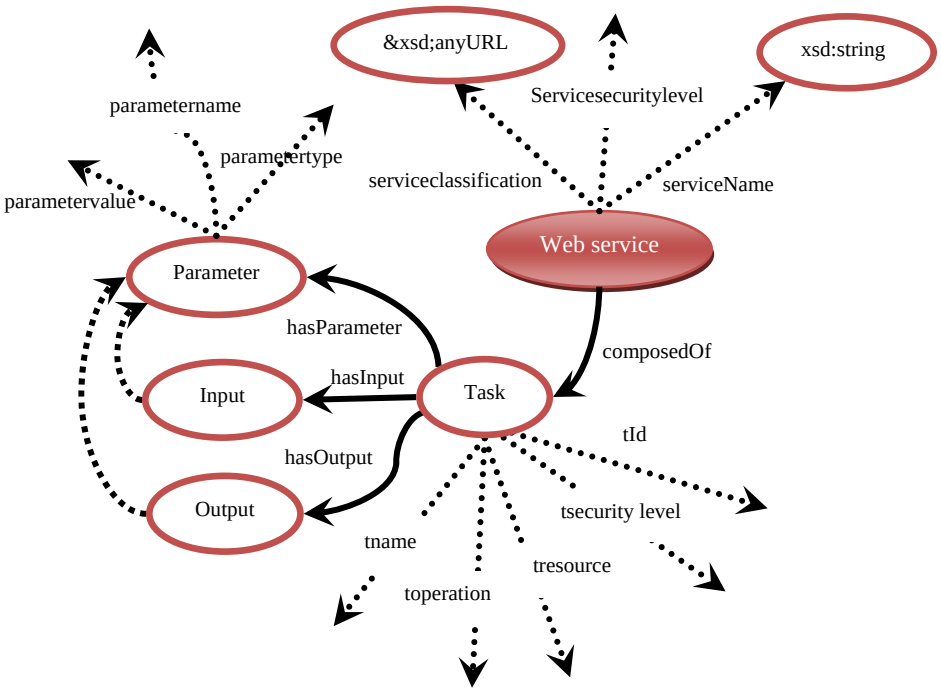


Figure 5. Web Service Ontology

```
<owl:Class rdf:about="Parameter">
 <rdfs:subClassOf rdf:resource="&swrl;#Variable"/>
</owl:Class>

<owl:DatatypeProperty rdf:ID="parameterType">
 <rdfs:domain rdf:resource="Parameter"/>
 <rdfs:range rdf:resource="&xsd;anyURI"/>
</owl:DatatypeProperty>
```

Input and Output: specify the data transformation produced by the task. Inputs specify the information that the task requires for its execution. . the outputs produced by the invocation of an atomic process flow back to the client as a single message, the format of which is specified by the grounding. Inputs and outputs are subclasses of parameter.

```
<owl:Class rdf:ID="Input">
 <rdfs:subClassOf rdf:resource="Parameter"/>
</owl:Class>

<owl:Class rdf:ID="Output">
 <rdfs:subClassOf rdf:resource="Parameter"/>
</owl:Class>
```

hasInput: ranges over input class. this property is subproperty of hasparameter

hasoutput : ranges over output class. this property is subproperty of hasparameter

hasParameter:The following example shows the definition of hasParameter, and its subproperties hasInput, hasOutput:

```
<owl:ObjectProperty rdf:ID="hasParameter">
 <rdfs:domain rdf:resource="Task"/>
 <rdfs:range rdf:resource="Parameter"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hasInput">
```

```
 <rdfs:subPropertyOf rdf:resource="hasParameter"/>
 <rdfs:range rdf:resource="Input"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hasOutput">
 <rdfs:subPropertyOf rdf:resource="hasParameter"/>
 <rdfs:range rdf:resource="Output"/>
</owl:ObjectProperty>
```

6.2. Permission-Role Assignment Ontology

An access control policy may be described in general as a security subject being permitted or denied access to a particular security object with a particular operation, . in our proposed model, a permission specifies whether the security subject has access on the security objects (task or web service). Using the OWL-S vocabulary , an permission role assignment ontology is defined as illustrated in figure 6, Permission-role assignment specifies which role has the permission to access what resources. In fact, it provides predefined access control policies. Therefore, permission-role assignment can be represented by an access control policy ontology. It defines access control policies using a tuple of four elements (security subject, security object, operation, sign) as shown in Figure 6. The security subject will be a role. in the CSAT-RBAC model, operation is always “hasAccessTo”. a sign can be either positive or negative depending on what security policy is used. Security object can be any type of protected resource, such as a Web service or a Task.

Permissionroleassignment: this class is domain for tow dataproperty (operation and sign) and tow object property (securityobject, securitysubject).

```
<owl:Class rdf:ID="Role">
 <rdfs:subClassOf rdf:resource="permissionroleassignment"/>
</owl:Class>

<owl:Class rdf:ID="#Webservice">
 <rdfs:subClassOf rdf:resource=" permissionroleassignment "/>
</owl:Class>
```

operation: is a data property ranges over value "hasaccessto"**.**

sign: a sign can be either positive or negative depending on what security policy is used.

securitobject: ranges over instances of Web service as defined in the web service Ontology**.**

securitysubject: is object property ranges over Role class.

Role: Roles represent the privileges of users within an organization. They bring together users with permissions in the CSAT-RBAC model. a role is assigned with property “rolename”, “juniorrole”, “seniorrole” “roletype”, “roleweight” as shown in Figure 6.

roletype: can be either capability role or request role, depending on whether the role is mapped to a user capability or a user request.

rolewight :weight of the role is the total weight of permissions assigned to this role. A role is associated with users and permissions via user-role assignment and permission-role assignment respectively.

seniorrole Role inheritance is an important relation in the CSAT-RBAC model, it is specified by using the property.

juniorrole. If one role is specified as the senior role of another then it will inherit all properties of the junior role, whereas if one role is specified as the junior role of another role, then all its properties will be inherited by the senior role.

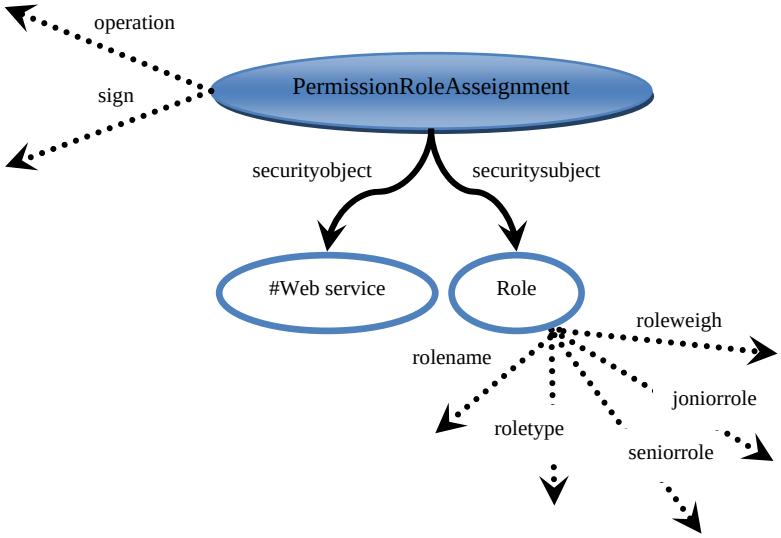


Figure 6. Permission Role Assignment Ontology

rolename: refers to the name of the role that is being offered by the administrator in the organization. It can be used as an identifier of the role.

6.3. Credential Ontology

In CSAT-RBAC model, user-role assignment is no longer performed between user and role. Instead, roles are assigned to user dynamically based on Credentials which are the mean to establish trust between a client and the service provider. They are assertions about a given user, referred to as the owner, issued by trusted third parties called Certification Authorities (CAs). They are digitally signed using the private key of the issuer CA and can be verified using the issuer's public key. A credential contains typically a set of arbitrary properties characterizing the owner and are specified via (name, value) pairs. each credential has a type based on the set of attribute names in the credential. In our proposed model, credentials contains also the user capability which indicates what security objects can be accessed by a particular user. It can be described in the form of subject-operation-object. We will describe user capability as an access control policy ontology, which describes an access control policy using a tuple of four elements (Security subject, Operation, Security object, Sign) Figure 7 User Capability is provided by a user credential, so it may be described as a property of user credential.

Credential : a credential is a class has tow object property(hasattribut, hascapaility) and four datatype properties (credenialid, credentialtype, issuer, owner).

issuer: is the name of the Certification Authorities (CA) that issues the credential.

owner is the name of the credential owner.

type: identifies the type of the credential.

hasattributes: ranges over Attribute class.

Attribute : is a pair (attributename ; attributevalue) dataproperties

attributename: is the name of the attribute

attributevalue :is a value in the attribute domain.

hascapapility: ranges over capability class.

UesrCapapility: indicates what security objects can be accessed by a particular user. It can be described in the form of subject-operation-object. We will describe user capability as an access control policy ontology, which describes an access control policy using a tuple of four elements (Security subject, Operation, Security object, Sign) Figure 7. User capability is provided by a user credential, so it may be described as a property of user credential.

usercapapilitysecuritysubject: ranges over an instance of web service as defined in the web service Ontology.

usercapapilitysecurityobject: ranges over the userid of a particular user.

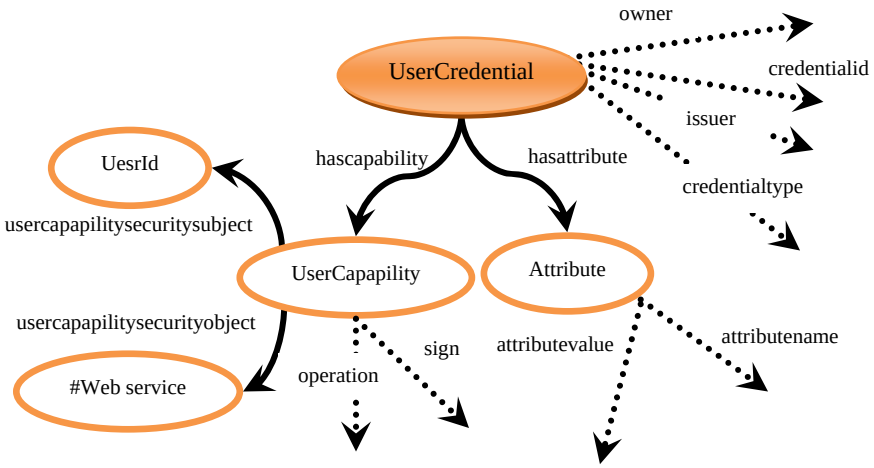


Figure 7. User credential ontology

6.4. Constraint Ontology:

Constraints are conditions that must be satisfied when authorizations are granted. They are used to describe different security policies.

constraint: has a number of properties including constraint name, constraint type, constraint target and constraint purpose as shown in Figure 8.

constraintType:can be entity's attribute constraint(user attribute, web service -task attribute), and environmental constraint(time, location, system load, …).

constraintTarget: indicates what the constraint is imposed on. For example, constraints may be imposed on user-role assignment.

constraintPurpose: tells what the constraint is used for, such as precondition or obligation. Preconditions must be satisfied before any action is taken, such as granting an access right to a user. Obligations are rules that must be followed in an action, they constrain the way a request is executed. Constraints can be imposed on any concept in the model.

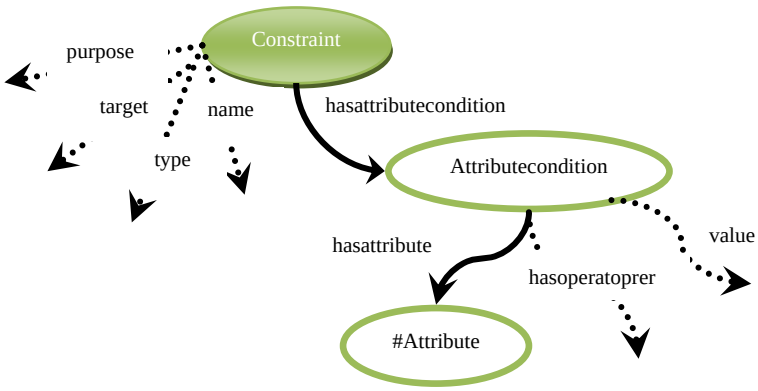


Figure 8. Constraint Ontology

each constraint is represented by conditions of the type <Attribute> <Operator> <VALUE>, where Attribute is an instance of the attribute in the credential ontology ; OP is a logical operator in the set {>, ≥, <,

≤, ≠, =}; VALUE is a specific value of attribute.

hasAttributeCondition: ranges over AtributeCondition

attributeCondition: has tow dataproperty (operator, value) and one objectproperty hasattribute

hasAttribute: ranges over an instance of attribute as defined in the credential Ontology.

hasOperator: The operation within an attribute condition may be relational operations (such as "=", "≠"), or containment operations, conjunction, disjunction, negation, and quantification

value: is the value of the attribute.

6.5. Dynamic Role Assignment Process ontology

User-role assignment matches user credentials with appropriate roles, and decides whether or not an access is granted. In this model, user-role assignment will be represented by an access control process ontology. OWL-S provides a vocabulary for describing service profiles and service processes. We will use some of the vocabulary provided in OWL-S to describe the user-role assignment process Figure 9.

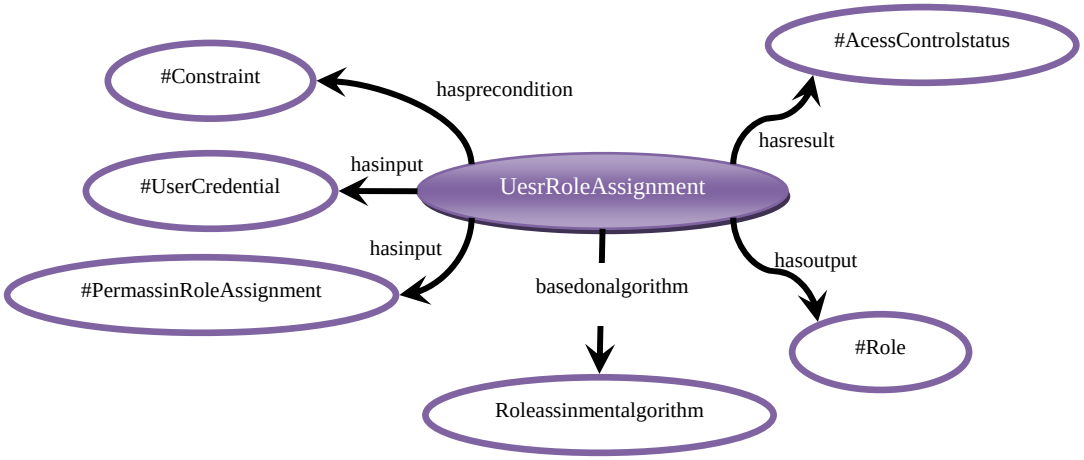


Figure 9. User Role Assignment Ontology

user credentials and permission-role assignment are described as the input to the process, and constraints imposed on the user-role assignment are described as the preconditions of the process. The process of deriving an access control decision relies on the role assignment algorithms. We describe it by using a new property called "basedOnAlgorithm" that has range "role assignment algorithm". The algorithms will now take user credentials and constraints as inputs, to produce a matching role, Since OWL-S provides vocabulary for describing logic expressions, we will describe role assignment with an OWL vocabulary. The output of the process is a matching role for each credential. The result will be an access control decision, either access granted or access denied.

6.6. Session Ontology

A Session refers to the set of events that occurs from when a user connects to an application system to when that user disconnects from it. A session ontology represents the profile of a particular session, Session profiles record all events that happened in each session, and they can be described by a number of parameters. These parameters are often dynamic, as they change with time. The session ontology represents all these parameters as properties. Session properties can vary from one system to another. The common properties are start time, end time, location, activity record and system status (Figure 10).

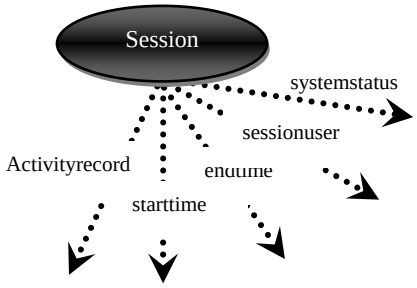


Figure 10. Session Ontology

A session ontology is linked with user ontology, role ontology, Web service ontology, A-function ontology and constraint ontology. It provides a variety of security conditions for supporting dynamic role assignment. . a session ontology is linked with credential ontology, role ontology, Web service ontology, Task ontology and constraint ontology. It provides a variety of security conditions for supporting dynamic role assignment

7. Complete Ontology-based Access Control Model

The complete CSAT-RBAC ontology shown in Figure 11 is formed by aggregating all component ontologies. These separate security ontologies are linked together by defining one ontology as the property of another. Once the relationship between different components of the model is formalized with ontologies, security reasoning becomes possible. We will be able to query and discover the required security information. Having all required security conditions available, the complete ERBAC ontology is able to generate access control decisions automatically.

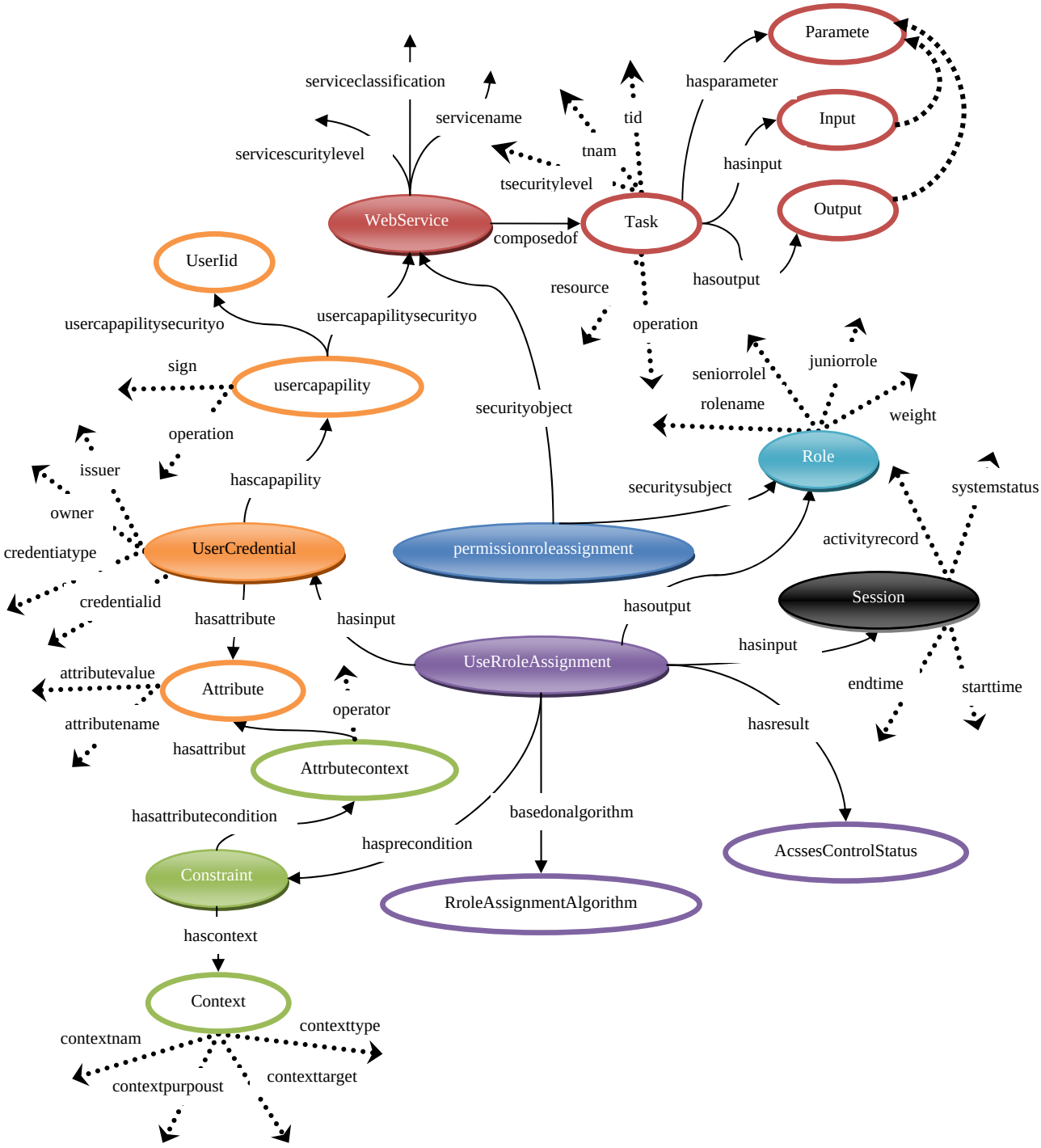


Figure 11. Ontology-based Access Control (OBAC)

8. A Proposed Architecture for Implementing the CSAT-RBAC Model

Figure 12 shows our proposed architecture for implementing the OBAC model. This architecture shows the details of the authorization process which is used during the decision making process in OBAC. This architecture contains a number of external components and a number of authorization components which are described in the following:

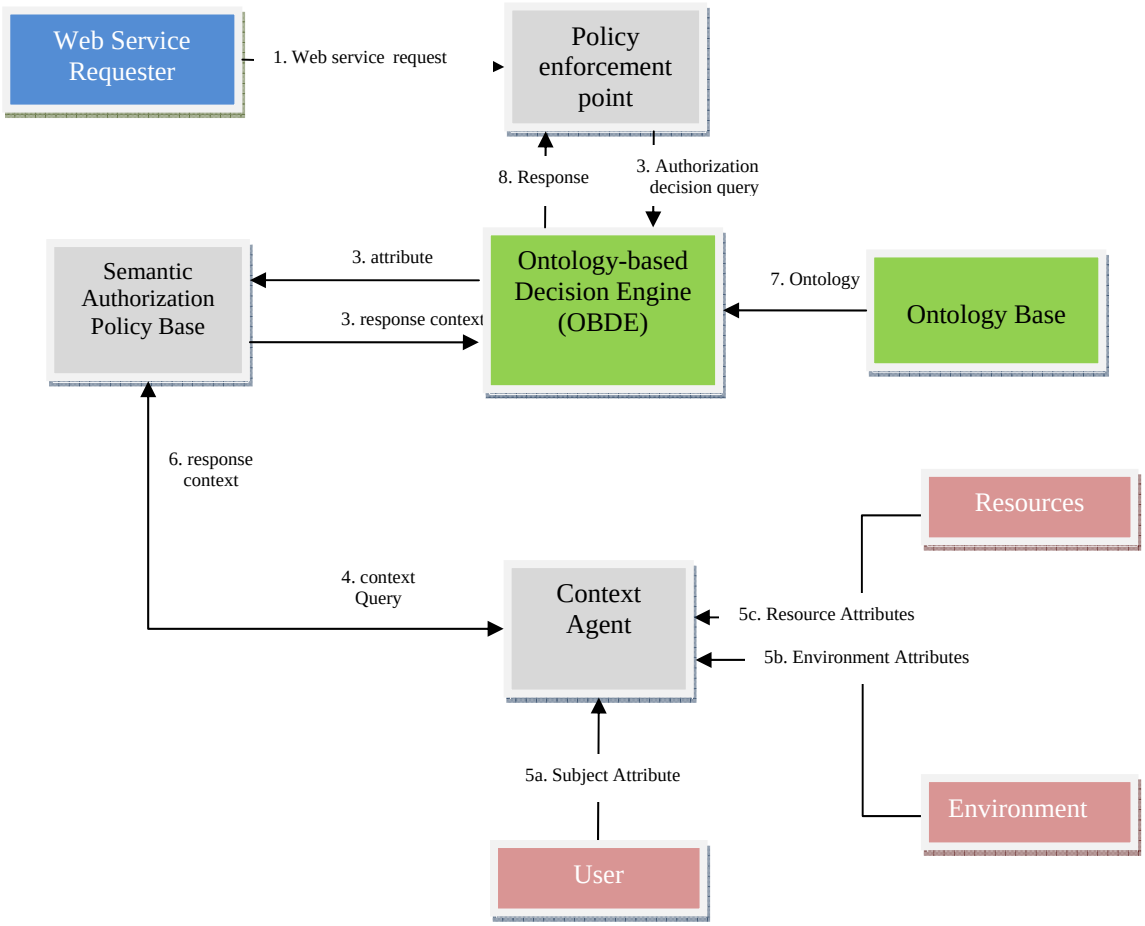


Fig 12. OB-CSAT-RBAC architecture

Web Service Requester: the agent (user) who makes the request to access a particular Web service or an Task of a Web service. Web service requesters are represented by their credentials.

Policy Enforcement Point: this module executes the result produced by the access control decision engine. If a request is granted, the access enforcement engine will fetch the user requested data from the back-end resource repository

Ontology-based Decision Engine (OBDE): which, after receiving a request from the policy enforcement point, uses its inference engine to determine whether this subject should be authorized to access the requested object?

Ontology Base: This includes ontologies that describe different domains of access control.

Semantic Authorization Policy Base: This includes the explicit authorization rules that are defined by security administrators of system.

Context Agent: collects context information from the resources, users and environment.

9. Conclusion and future work:

In this paper we presented an ontology-based access control (OBAC) to support semantic web service, Security ontologies are developed to specify concepts and terms involved in this model. Our research motivation comes from the need to reduce the gap between security services and semantic web, Our proposed access control model is Expressive and general with these important features:

- The use of ontology provides reasoning ability for access control decision making, and allows access control information to be searched, queried and discovered automatically.
- Modeling different domains of access control has added a considerable generality to the model, for example the credential ontology are going to be universally used for user authentication and authorization.

- Our proposed model has a higher degree of interoperability compared with other approaches to access control. This is because of the nature of ontologies in providing semantic interoperability.
- Context sensitive access control model, the constraint ontology represents different types of context constraint.
- Our proposed model is designed based on the widely accepted semantic web languages, Web Ontology Language (OWL) and Web Ontology Language for Service (OWL-S), therefore its implementation can be easily achieved by existing tools designed for working with these languages.

10.References

[1] Agarwal S., B. Sprick. Access Control for Semantic Web Services. Web Services, IEEE International Conference on, p. 770, IEEE International Conference on Web Services (ICWS'04), 2004.

[2] Antoniou, G., & Doerr, M. Web Ontology Languages. Semantic Web Services: Theory, Tools and Applications. IDEA Group. 2006, pp.96-109.

[3] Berners-Lee, T., J. Hendler and O. Lassila. The Semantic Web. Scientific American Magazine. http://www.sciam.com/article.cfm?id=the-semantic-web&print=true. Retrieved March 26, 2008.

[4] Bonatti, Piero A., Claudiu Duma, Norbert Fuchs, Wolfgang Nejdl, Daniel Olmedilla, Joachim Peer, and Nahid Shahmehri. Semantic web policies - a discussion of requirements and research issues. In 3rd European Semantic Web Conference (ESWC), volume 4011 of Lecture Notes in Computer Science, Budva, Montenegro, June 2006. Springer.

[5] Brickley, D. , and Guha, R. V. RDF vocabulary description language 1. 0: RDF schema. W3C Recommendation. Retrieved November 4, 2009, from http://www. w3. org/TR/PR-rdf-schema

[6] L. Cirio, I.F. Cruz, and R. Tamassia, A Role and Attribute Based Access Control System Using Semantic Web Technologies. *in Proc. OTM Workshops (2).* 2007, pp.1256-1266.

[7] De Bruijn, J. 2006. The Web service modeling language WSML (Deliverable D16. 1v0. 21). Retrieved November4, 2009, from http://www.wsmo.org/wsml/wsml-resources/wsml-eswc2006-handouts.pdf

[8] Denker, G. , Kagal, L. , Finin, T. , Paolucci, M. , and Sycara, K. Security for DAML web services: Annotation and matchmaking. *In: Proceedings of the 2nd International Semantic Web Conference*. Sanibel Island, Florida, USA, 2003.

[9] Joshi, J. Access-control language for multi domain environments. *IEEE Internet Computing.* 2004, **8**(6): 40–50

[10] Kagal, L. , Finin, T. , Joshi, A. A policy language for a pervasive computing environment. *In Proceeding of 4th IEEE International Workshop on Policies for Distributed Systems and Networks*. 2003, pp. 63–74.

[11] Kagal, L. , Finin, T. , Joshi, A. , Niu, J. , Sandhu, R. , and Winsborough, W. ROWLBAC - Representing Role Based Access Control. *in OWL' Proceedings of SACMAT'08*. Estes Park, Colorado, USA, 2008.

[12] Lassila, O., Swick, R. Resource Description Framework (RDF) Model and Syntax Specification. Retrieved from http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/, 1999.

[13] Mcliraith, S. , Son, T. C., and Zeng, H. Semantic Web services. *IEEE Intelligent Systems, Special Issue on the Semantic Web*. 2001, **16**(2): 46-53.

[14] Moses, T. eXtensible Access Control Markup Language (XACML) version 2. 0. 2005. Retrieved October 4, 2009 from http://docs. oasis-open. org/xacml/2. 0/access control-xacml-2. 0-core-spec-os. Pdf, 2005.

[15] OWL Technical Committee. OWL: Web ontology language. A W3C specification. Retrieved November 19, 2009, from http://www.w3.org/2004/OWL/, 2004.

[16] OWL-S Technical Committee. OWL-S: Semantic markup for Web services. W3C member submission. Retrieved November 19, 2009, from http://www. w3.org/Submission/2004/SUBMOWL-S-20041122/ , 2004.

[17] Priebe, T., Dobmeier, W., Kamprath, N. Supporting attribute-based access control with ontologies. Availability, Reliability and Security, ARES 2006. The First International Conference on , vol., no., pp. 8 pp., 20-22 April 2006

[18] Prud'hommeaux, E. W3C ACL System. Retrieved from http://www.w3.org/2001/04/20-ACLs.html, 2001.

[19] Qin, L. , Atluri, V. Concept-level access control for the semantic web. In ACM Workshop on XML Security, Fairfax, VA, USA. 2003, pp. 94–103.

[20] WSMO Technical Committee. Web service modeling ontology (WSMO). A W3C Member Submission. Retrieved November 19, 2009, from http://www.w3.org/Submission/WSMO/ , 2005.

illustrate the effectiveness of the method.

7. Acknowledgements

This work was supported by the National Natural Science Foundation of China (50976072), leading academic discipline project of Shanghai municipal education commission (J50501) and the science foundation for the excellent youth scholar of higher education of Shanghai (slg09003)

8. Reference

[1] W. H. Reed, T. R. Hill. Triangular Mesh Methods for the Neutron Transport Equation. *Scientific Laboratory Report*. Los Alamos, LA-UR-73-479, 1973.

[2] P. Lesaint, P. A. Raviart. On a Finite Element Method for Solving the Neutron Transport Equation. *Mathematical Aspects of Finite Elements in Partial Differential Equations*. Academic Press, San Diego, 1974.

[3] C. Johnson and J. Pitkaranta. An analysis of the Discontinuous Galerkin Method for a Scalar Hyperbolic Equation. *Math. Comput*. 1986, **46**(1).

[4] G. R. Richter. An Optimal-order Error Estimate for the Discontinuous Galerkin Method. *Math. Comput*. 1988, **50**: 75.

[5] T. Peterson. A Note on the Convergence of the Discontinuous Galerkin Method for a Scalar Hyperbolic Equation. *SIAM J. Numer. Anal*. 1991, **28**: 133.

[6] B. Cockburn, C. W. Shu, Runge-Kutta. Discontinuous Galerkin Methods for Convection-dominated Problems. *J. Sci. Comput*. 2001, **16**: 173-261.

[7] N. Chevaugeon, J. Xin and P. Hu. Discontinuous Galerkin Methods Applied to Shock and Blast Problems. *J. Sci. Comput*. 2005, **22**: 227-243.

[8] J. X. Qiu, B. C. Khoo and C. W. Shu. A Numerical Study for the Performance of the Runge Kutta Discontinuous Galerkin Method Based on Different Numerical Fluxes. *J. Comput. Phys*. 2006, **212**: 540-565.

[9] J. X. Qiu, T. G. Liu and B. C. Khoo. Simulations of Compressible Two-Medium Flow by Runge-Kutta Discontinuous Galerkin Methods with the Ghost Fluid Method. *Commun. Comput. Phys*. 2008, **3**: 479-504.

[10] J. Naber. A Numerical Solver for Compressure two-fluid Flow, Report MAS-E0505, CWI, http://oai.cwi.nl/oai/asset/10961/10961D.pdf, 2005.

[11] N. Robert, D. Nam, T. Theo. Direct Numerical Simulation of Compressible Multiphase Flows: Interaction of Shock Waves with Dispersed Multimaterial Media, ICMF'04, Yokohama, Japan, May 30-June 4 , 2004.

[12] R.P.Fedkiw, T.Aslam, B.Merriman, S.Osher. A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows(theGhost Fluid Method) [J]. *J. Comput. Phys*. 1999, **152**: 457-492.

[13] S. Osher, J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton- Jacobi formulations. *J. comp. Phys.* 1988, **79**:12-49.

[14] S. Osher and R. P. Fedkiw. Level Set Methods: An Overview and Some Recent Results. *J. Comput. Phys*. 2001, **169**: 463-502.

[15] Jiang, G.S., and Peng, D.P. Weighted ENO schemes for Hamilton-Jacobi equations. *SIAM J. Sci. Comput*. 2000, **21**(6): 2126-2143.

[16] J. F. Haas and B. Sturtevant. Interactions of weak shock waves with cylindrical and spherical gas inhomogeneities. *J. Fluid Mech*. 1987, **181**: 41.