I.J. Computer Network and Information Security, **2011, 1, 18-25**
Published Online February 2011 in MECS (http://www.mecs-press.org/)

A Semantic Context-Based Model for Mobile Web Services Access Control

Haibo Shen
Hubei University of Technology / School of Computer, Wuhan, China
Email: jkxshb@163.com

Yu Cheng
Hubei University of Technology / School of Computer, Wuhan, China
Email: chengyu125@126.com

***Abstract*—As mobile web services becomes more pervasive, applications based on mobile web services will need flexible access control mechanisms. Unlike traditional approaches based on the identity or role for access control, access decisions for these applications will depend on the combination of the required attributes of users and the contextual information. This paper proposes a semantic context-based access control model (called SCBAC) to be applied in mobile web services environment by combining semantic web technologies with context-based access control mechanism. The proposed model is a context-centric access control solutions, context is the first-class principle that explicitly guides both policy specification and enforcement process. In order to handle context information in the model, this paper proposes a context ontology to represent contextual information and employ it in the inference engine. As well as, this paper specifies access control policies as rules over ontologies representing the concepts introduced in the SCBAC model, and uses semantic web rule language (SWRL) to form policy rule and infer those rules by JESS inference engine. The proposed model can also be applied to context-aware applications.**

***Index Terms*—mobile web services, context-based access control, ontology technology, OWL, SWRL**

I. INTRODUCTION

With the recent developments in the cellular world, the high-end mobile phones and PDAs are becoming pervasive and are being used in different application domain. Integration of the web services and cellular domains lead to the new application domain, mobile web services [1]. Mobile web services are defined as Web Services that are deployed on mobile devices and are published over the Internet, wireless network or within the operators' network. The goal of mobile web services is to offer new personalized services to consumers on their mobile devices such as telephones, wireless-LAN-enabled PDAs and laptop computers.

In mobile web services environment, the context of a user (i.e. location, time, system resources, network state, user's activity, battery power level, etc.) is highly dynamic, and granting a user access without taking the user's current context into account can compromise security as the user's access privileges not only depend on "who the user is" but also on "where the user is" and "what is the user's state and the state of the user's environment". As a result, even an authorized user can damage the system as the system may have different security requirement within different contexts. As well as, it is crucial to have a policy system that understands and interprets semantics of the context correctly. This type of access control is called semantic-aware access control. While the basic message-level security can be provided, the end-point security comprising proper identity and access control mechanisms still poses a great challenge [2]. Traditional access control mechanisms based on the identity/role of user break down in such an environments and a semantic-aware context-based access control mechanism is required [3].

To address the above problems, this paper proposes a semantic context-based access control model (called SCBAC) for mobile web services by combining semantic web technologies with context-based access control mechanism. In order to handle context information in the model, we propose a context ontology and a context handling framework to represent and handle contextual information, and employ it in the inference engine. Furthermore, this paper specifies access control policies as rules over ontologies representing the concepts introduced in the SCBAC model, and uses semantic web rule language (SWRL) to form policy rule and infer those rules by JESS inference engine.

This paper is organized as follows: Section 2 briefly discusses main related technologies. Section 3 presents SCBAC model and its authorization framework. Section 4 discusses the context modeling and handling. Section 6 develops a context-aware access control policy ontology. Section 6 presents provides a summary of existing research works in context modeling and access controls for mobile web services. In the section 7 the conclusion is given.

II.. BACKGROUND

Manuscript received January 1, 2008; revised June 1, 2008; accepted July 1, 2008.
Copyright credit, project number, corresponding author, etc.

A. Semantic Web Technologies and Access Control

With the advent of the Semantic Web, Web services have gained even more importance. Semantic Web technologies, especially ontologies, can describe Web services with machine understandable semantics, thus enabling new features like automatic composition, simulation and discovery of Web services. In interoperable e-business architectures based on the semantic web vision, ontology-based domain models are used as controlled vocabularies for resources description, allowing users to obtain the right resources at the right time. The semantic access control [4] or semantic-based access control [5] is an access control mechanism of applying semantic web technologies to access control.

OWL (Web Ontology Language) [6] was developed as an ontology language for constructing ontologies that provide high-level descriptions of Web content. These ontologies are created by building hierarchies of classes describing concepts in a domain and relating the classes to each other using properties (attributes). OWL can also represent data as instances of OWL classes—referred to as individuals—and it provides mechanisms for reasoning with the data and manipulating it. OWL also provides a powerful axiom language for defining how to interpret concepts in an ontology. This paper uses OWL to represent semantically resources and user attributes, and the context and policy ontology.

This paper uses OWL to represent the metadata about the resources and user attributes. The processing and analysis of ontology, i.e. drawing conclusions and gaining new information through combination, takes place in the logical layer. Implicit information in the data can be made explicit by using so-called reasoners or inference engines. Simple inferences are already possible with OWL, for instance through inheritance. But there is a limitation in OWL reasoning, more complex custom inference rules require the usage of a special rule language. A promising approach is the Semantic Web Rule Language (SWRL) [7]. SWRL allows users to write rules that can be expressed in terms of OWL concepts and that can reason about OWL individuals. The main advantage of using SWRL is its ability to provide support for complex relationships between properties, therefore extending the expressiveness of what we can define in OWL (OWL-DL). For example, it is impossible to assert that persons who study and live in the same city are "home students" in OWL, whereas this can be done easily using SWRL rules:

Studies(x,y)^lives(x,z)^location(y,u)^location(z,u)→homeStudent(x)

The most evident advantage of integrating access control architectures with semantic web technologies is the opportunity of applying the fine-grained categorization primitives of semantic web languages to provide a more detailed description of the entities involved. By doing this, rules applying to a given concept can be extended to related concepts according to well defined principles such as subsumption, union, intersection. Context information (for instance users, roles, resources, and credentials) can be expressively represented with concepts and instances of an OWL ontology whose consistency can be automatically checked with existing tools.

B. Contexts and Context-Based Access Control

A widely accepted definition of context is [8]: Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves. A single definition of context does not exist nor would it be sensible. Bazire and Brezillon [9] collected 150 definitions from various areas of research.

Location, time, user's activity, battery power level, etc are typical contextual data for mobile computing. A system is context-aware if it can extract, interpret and use context information and adapt its functionalities to the current context. The access control for the resources in the context-aware system is usually called the context-aware access control or context-based access control [10]. The challenge for such systems lies in the complexity of collecting, representing, processing and using contextual data [11].

III. SEMANTIC-AWARE CONTEXT-BASED ACCESS CONTROL MODEL

The proposed SCBAC model adopts a context-centric policy modeling approach that treats context as a first-design principle for policy specification, therefore, SCBAC model is centered on the context concept, which is composed by information characterizing the controlled resources, requestors, owners, and environment surrounding them. In this model, the context act as a mediator between the entities requiring access to resources and the set of permissions assigned to these resources. In this section, we will introduce SCBAC model. After explaining the concepts of contexts, we first introduce the SCBAC model components, and then describe the authorization architecture.

A. Context Defining

Unlike Identity Based Access Control and Role Based Access Control, the SCBAC model adopts context-centric access control solutions, which defines permissions based on just about any security relevant contexts. For access control purposes, we are concerned with four types of contexts:

1. Subject Contexts (SC). A subject is an entity (e.g., resource requester or owner) that takes action on an object or resource. Subject Contexts (SC) define the specific subject-related contexts that must be held or exercised by the subject in order to obtain rights to an object or resource. In our SCBAC model, subject contexts are used to determine access rights for an entity requesting access privileges. Entities requesting access privileges may or may not be required to possess unique subject contexts, in which case their privacy is protected at the time of access. Subject-related contexts include: the subject's role, identity, credentials, name, organization,

activity, location, task; services, devices, network, and platform provided to subject, social situation (e.g. who you are with, and people that are nearby), and so on.

2. Object Contexts (OC). An object is an entity (e.g., a web service, data structure, or system component) that is acted upon by a subject. As with subjects, objects have contexts that can be leveraged to make access control decisions. Object contexts (OC) are any object-related information that can be used for characterizing the situation in which the protected object was created and its current status, which is relevant for making access control decisions. As with subjects, objects may or may not be defined using unique object contexts.

3. Transaction Contexts (TC). In pervasive computing environment, transactions involve the user, the mobile platform, the specific resource or service being accessed, and the physical environment (such as location, time) of both the user and platform. A transaction specifies a particular action to be performed in the system. For example, a transaction may be initiated by a user from a specific location, to access a resource that is currently in a state s, at a particular time of day. Transaction contexts capture significant information about the transactions that are occurring or have occurred in the past.

4. Environment Contexts (EC). Environment Contexts describe the operational, technical, and even situational environment at the time a transaction takes place. Environment contexts, such as current date and time, the current virus / hacker activities, the network's security level, temperature, air quality, light or noise level, or other contextual information that is relevant to access control, are not associated with a particular subject or a resource, but may nonetheless be relevant in applying an access control policy. The state of the environmental conditions must be captured via sensors that are embedded in the environment.

B. SCBAC model

The basic SCBAC model has the following components:

1. S, O, E and T are subjects, objects, environments and transactions, respectively;

2. SC_i ($1 \le i \le K$), OC_j ($1 \le j \le M$), EC_k ($1 \le k \le N$) and TC_n ($1 \le n \le J$) are the contexts for subjects, objects, environments, and transactions, respectively;

3. CONT(s), CONT(o), CONT(e) and CONT(t) are context assignment relations for subject s, object o, environment e, and transaction t, respectively:

$$\mathrm{CONT}(s) \subseteq SC_1 \times SC_2 \times \ldots \times SC_K;$$
$$\mathrm{CONT}(r) \subseteq OC_1 \times OC_2 \times \ldots \times OC_M;$$
$$\mathrm{CONT}(e) \subseteq EC_1 \times EC_2 \times \ldots \times EC_N;$$
$$\mathrm{CONT}(t) \subseteq TC_1 \times TC_2 \times \ldots \times TC_L$$

4. Action (Act): an action is an event that a subject seeks to perform. An action can be given a list of parameters (e.g. contexts) defining how the action must be performed.

5. Permission Assignments (PA): a permission grants the right to a subject to perform an action on an object or resource. Permission assignments (PA) capture the privileged actions that a subject is authorized to hold or exercise on an object. The authorization is determined based on subject contexts, object contexts, transaction contexts, and environment contexts. One significant advantage of our SCBAC model is that rights can be assigned to contexts only; this allows policy to be specified on mere contexts alone.

The following function captures the rights that are assigned to a user when a given set of environment contexts are active and she is attempting to access an object with a particular set of object contexts:

(< Act, SC, OC, EC, TC >, Perm) $\in$ PA, where Perm = {Allow, Deny}

As indicated above, the permission assignment (PA) not only associates a permission with the user context(s), but makes it conditional on a set of active environment contexts. Clearly, rights may change for the same user accessing a resource if the object contexts, environment contexts, or even user contexts vary between requests. In our system, a request will be granted access rights if and only if:

(1) The policy rule assigning a specified action (Act) to an access request exists with the specified subject contexts (SC), object contexts (OC), environment contexts (EC), and transaction contexts (TC) that match those specified in the set of permission assignments (PA)

(2) The subject contexts (SC) are active for the user making the current request

(3) The object contexts (OC) are active for the object being accessed by the user

(4) The environment contexts that are made active by the current environmental conditions are contained in the set EC.

6. In the most general form, a Policy Rule that decides on whether a subject s can access an object o in a particular environment e and within a transaction t, is a Boolean function of s, o, e and t's contexts:

Rule: can_access (s, o, e, t) $\leftarrow f$ (CONT(s), CONT(o), CONT(e), CONT(t))

Given all the context assignments of s, o, e and t, if the function's evaluation is true, then the access to the resource is granted; otherwise the access is denied.

The SCBAC model is illustrated in Fig. 1.

C. SCBAC Authorization Framework

Authorization decision evaluates all contexts and the requested rights. SCBAC authorization architecture is illustrated in Fig.2 below.

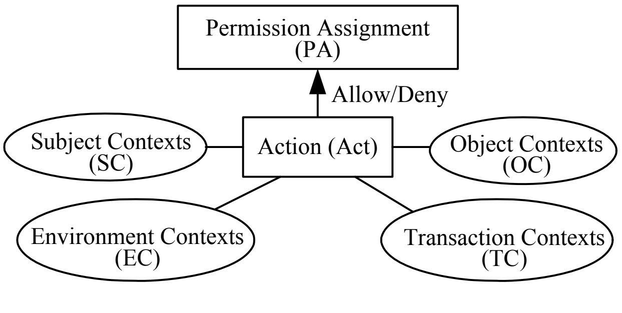
Permission Assignment (PA)
Allow/Deny
Subject Contexts (SC)
Action (Act)
Object Contexts (OC)
Environment Contexts (EC)
Transaction Contexts (TC)

Figure 1. SCBAC model components

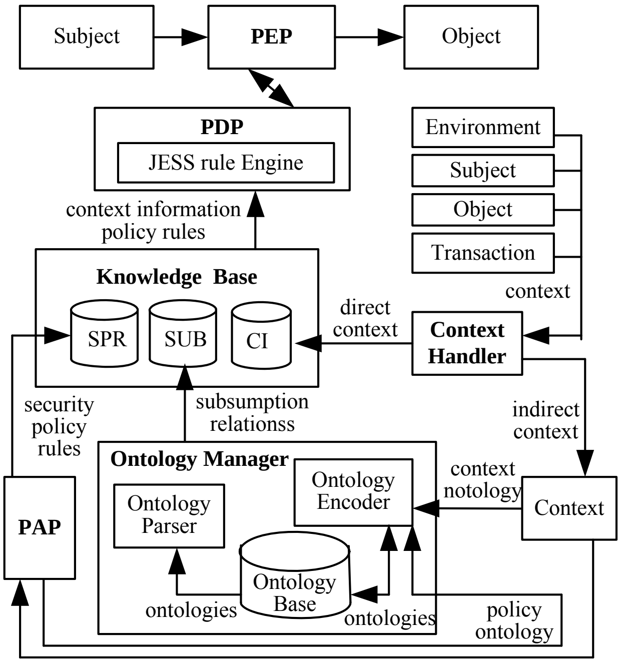

Figure 2. SCBAC authorization architecture

The diagram reflects the following logical actors involved in SCBAC model:

1. The Knowledge Base (KB) is a data repository of domain ontology. KB is composed of the set of security policy rules in SPR, subsumption relations between concepts (SUB) and direct context information (CI). Inference of implicit authorization rules is based the facts and rules in the KB.

2. The Ontology Manager is responsible for gathering and updating ontologies in domains of subjects, objects, actions, and policies and also reducing the semantic relations to the subsumption relation. We can use the Protégé-OWL ontology editor in the Protégé-OWL ontology development toolkit [12] to create all ontologies.

3. The Context Handler is responsible for getting the contextual information from external environment, and performing assertion to the KB database according to the model of the domain knowledge as well as maintaining the consistency of the KB since the KB has to accurately reflect the dynamic changes of the environment.

4. The Policy Enforcement Point (PEP) is responsible for requesting authorization decisions and enforcing them. In essence, it is the point of presence for access control and must be able to intercept service requests between service requester and providers. The most important security engineering consideration for the implementation of a PEP is that the system must be designed such that the PEP cannot be bypassed in order to invoke a protected resource.

5. The Policy Decision Point (PDP) is responsible for evaluating the applicable policies and making the authorization decision (allow or deny) by making use of an inference engine, based on facts, contexts and rules in the KB.

In general, an access request can be modeled as a triple (u, p, c, r), which means that a user u requests to execute the privilege p on the resource r in a given context c. To evaluate this request the framework has to verify whether there exists an access control authorization granting p to requester r in given context c. Thus, it is necessary for PDP to perform a query (for example, SPARQL query [13]) to KB. The PDP is in essence a policy execution engine. When a policy references a subject contexts, object contexts, or context that is not present in the request, Jess Rule Engine [14] in PDP contacts the KB to retrieve the contextual information.

6. The Policy Administration Point (PAP) is responsible for creating a policy or policy set.

Ⅳ. CONTEXT MODELING AND CONTEXT HANDLING

The SCBAC model is a context-centric access control solutions, context is the first-class principle that explicitly guides both policy specification and enforcement process. But context-centric access control solutions need to adopt ontology technologies as key building blocks for supporting expressive context modeling and reasoning. Therefore, this section will discuss the context modeling and handling.

A. Semantic Context Modeling

In order to use the context information in SCBAC model, it first needs to find out what the context consists of. In mobile web services environment, some elements, such as location, time, devices, network, resources, resource requestors, resource owner and requestor's activity, are most fundamental context for capturing the information about the executing situation. Context modeling is the specification of all entities and relations between these entities which are needed to describe the context as a whole. The object of context modeling is to model a set of upper-level entities and provide flexible extensibility to add specific concepts in different application domains. In this paper, we present a context ontology MWSContext (shown in Fig.3) for modeling context in mobile web services environment.

MWSContext categorizes the context into eight main upper categories: Location, Time, Activity, User, Platform, Resource, Policy and Environment. Each entity is associated with its properties and relations with other entities. MWSContext consists of eight subclasses. Time class is defined equivalent to time:TemporalEntity from W3C's standard ontology of Time [15]. The class Location represents the abstraction of a physical location, which has two subclasses including LogicalLocation and PhysicalLocation. The LogicalLocation class may include subclass URI, which represents the Universe Resource Identifier in Web. The class PhysicalLocation is used for representing physical location, such as a room, a city, and so on.

The resources are the entities located in mobile web services environment and accessed. The Resource class can have many subclasses, such as Service, Data, Agent, Application, Information, and so on. Resource may also have many properties, such as ResourceType and ResourceStatus.

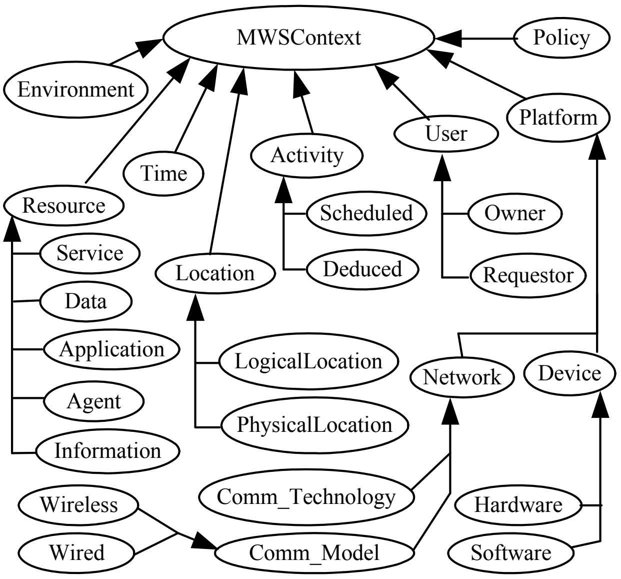

Figure 3. MWSContext ontology

The platform is an intermediate to let a requestor access a resource; therefore, it contains the subclasses Device and Network. Device class includes the two subclasses: Hardware, and Software. Hardware may include subclasses: Processor, Storage, and BatteryLife. Software may include subclasses: Application, Driver, and OperatingSystem. The Network class is defined by the some other sub networks, each network may itself contain some other connections. The Network class includes the two subclasses: Comm_Technology and Comm_Model. Comm_Model has two subclasses: Wireless and Wired. The Comm_Technology represents the technology (such as 2G,3G,4G etc.) by which the device can communicate with the other devices or services. Network has many important properties, such as bandwidth, speed rate, connection state, and so on. The quality of network is very important for mobile web services environment.

Fig.1 only shows the upper context ontologies. The built-in OWL property owl:subClassOf allows for hierarchically structuring sub-class entities, thus providing extensions to add new concepts that are required in a specific application domain.

B. Context Handling

In order to implement SCBAC model for mobile web services, we need to provide some mechanisms to obtain, interpret and store the context. The context handling includes: (1) gather raw, low-level contextual data; (2) interpret the raw contextual data into high-level interpreted context; (3) reason the interpreted context to derive implications; (4) adapt the application behaviour on the basis of the implications.

The context is sensed through the sensors that are locally built into the mobile devices as well as present in the environment and is gathered by the acquisition modules of system. The gathered data is represented in a standard format so that it assists the sharing process. The gathered and represented context data is stored on local storage. The storage process allows the system to maintain a history of context that is used to identity preferences of the entities. Interpreting raw, low-level context to meaning, high-level interpreted context is the first phase of the interpretation process. In the second phase implications are then reasoned from the interpreted context. The interpreted contexts are categorized as 'What', 'Who', 'Where' and 'When' contexts. These contexts are the subsets of the gathered contextual data and provide identification, activity, spatial and temporal information.

This paper proposes a context handler to handle the context. The context handler is shown in Fig.4, which is related to two other external entities: Knowledge Base (KB) and Environment. KB is a repository in which the semantic contextual information is stored. It asks for contextual information from Context Handler, and interprets it too. Context Handler contains the four main parts: Context Interface, Context Acquisition Module, Context Reasoning Engine and Jess Rule Engine. Context Interface is an interface from which KB may ask for contextual information. Context Acquisition Module is responsible to gathers raw context information from the outside world and performs instance assertions into Context Reasoning Engine. A direct assertion (gets direct context) is performed by Context Reasoning Engine (for example, Racerpro). Racerpro is a DL interpreter that can perform direct assertions over OWL entities. Then, the implicit information is passed on to Jess Rule Engine for further processing (Considering present reasoning system can't process OWL knowledge and SWRL rules directly, the system uses SWRLJessTab in SWRLTab to transform OWL knowledge and SWRL rules into Jess facts and Jess rules). The Jess Rule Engine can perform indirect assertions (get indirect context) from the information using SWRL rules written as part of the context ontology. Results (direct context and indirect context) are asserted into CI in KB by Context Interface.

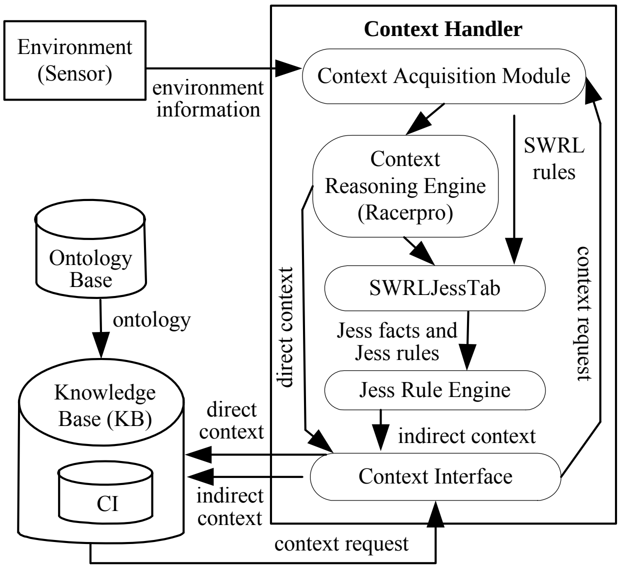

Figure 4. Context handler framework

V. Context-Based Access Control Policy

A. Context-Based Access Control Policy

Access control policies specify the actions that subjects are allowed to perform on resources depending on various types of conditions, e.g., resource state and context aspects. Policies are usually written in the form of restricted rules in that the action component of the rule returns a "Deny" or "Allow" decision. Each policy must be associated with the domain knowledge. Moreover, to make easier the task of policy evaluation, policies are enforced through a set of authorizations, stating for each subject the rights she has on the protected resources. Thus, we can encode each policy as a rule, that is, a rule whose antecedent represents the conditions stated in the policy subject and object specifications, and the consequent represents the entailed authorizations. In context-based access control, a policy must be able to use the contexts of the situation in order to perform access control. Therefore, it is necessary to reason over the domain knowledge to obtain semantics of the contexts. If the policy requires defined contexts, it will be written with a set of rules which identify which contexts are under consideration. In other words, these rules provide the parameters to reason over the KB in order to obtain the accurate information of the context. The context ontology provides contexts and their semantics to the policy in order to construct policy rules. The context-based access control policy can always look up the meanings of the contexts from the context ontology. Fig.5 depicts the relationships among an access control policy, policy rules, and context ontology. In SCBAC framework, context-aware access control policy is introduced. Context-aware access control policy is an access control policy but written in a form of rules to be able to capture the situational context.

SWRL has been introduced to extent the axioms provided by OWL to also support rules. In SWRL, the antecedent (called the body) and the consequent (called the head) are defined in terms of OWL classes, properties and individuals. This paper adopts SWRL to encode policy rules where the antecedent encodes the conditions specified in the policy, whereas the consequent encodes the implied authorizations or prohibitions. As consequence, the access control policies can be enforced by simply querying the authorizations, that is, the KB. The query can be easily directly implemented by the ontology reasoner by means of instance checking operations, or can be performed by a SPARQL query, if the ontology is serialized in RDF.

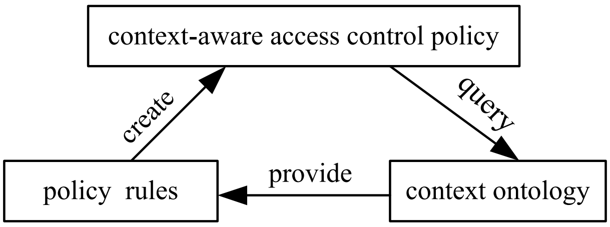

Figure 5. Construction of a context-aware access control policy

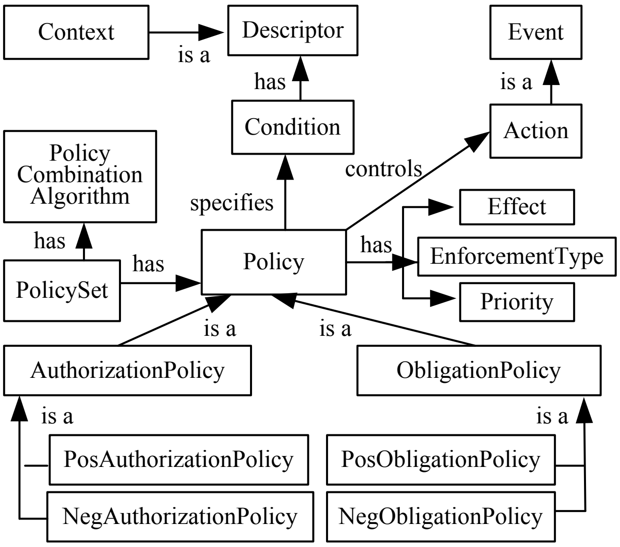

Figure 6. SCBACPolicy ontology

B. Policy Ontology

The SCBAC model is expressed in the form of ontologies, and the related ontologies define the SCBAC policy language. We can specify the ontology-based access control policies (policy ontology) according to the related ontologies in SCBAC model. Therefore, this paper develops a policy ontology (called SCBACPloicy) to integrate it easily with ontologies in SCBAC model. SCBACPolicy is shown in Fig.6.

The root class of SCBAC policy ontology is PolicySet, which includes member policy and has policy combination algorithm. Policies can be combined to form policy sets. Each policy set uses a policy combination algorithm that describes how the (possibly conflicting) policies should be combined. A policy combination algorithm's primary goal is to prevent or solve possible conflicts within one policy set, which can be achieved by specifying which of the different policies has priority.

SCBAC policies are divided into three kinds of policies: AuthorizationPolicy and ObligationPolicy. They are subclasses of class policy. Authorization policies specify which actions a subject is allowed (positive authorizations policies, PosAuthorizationPolicy) or not allowed (negative authorizations policies, NegAuthorizationPolicy) to perform in a given context. Obligation policies specify actions that a subject is required to perform (positive obligations) or for which such a requirement is waived (negative obligations). Positive obligations policy and negative obligations policy are represented by PosObligationPolicy and NegObligationPolicy respectively.

A policy defines a set of conditions, which are evaluated to determine whether a set of actions may be performed on a resource. So access control policy can be described as condition-action rules. A condition determines whether or not an action should be performed. In other words, conditions specify the environment (i.e. the descriptor) for an action to be executed. Descriptors, such as time, location or other contexts, express the condition of a policy. Context is a subclass of Descriptor. Thus, every policy is context-aware. An action is an event that an agent seeks to perform. Note that an action can either be a simple operation, or a bundle of complex operations provided as an integrated set.

Each policy has properties that state meta-information associated with the policy such as the entity that issued the policy (issuedBy), entities to which it applies (appliesTo) and the location (appliesWhere) and time (appliesWhen) of enforcing the policy, priority determine how policies should be ranked, and so on. Every policy is associated with EnforcementType value that could, for instance, have the value of 'Negative' to prohibit doing the rule in the policy, or it could be 'Negotiable' to allow agents to negotiate how to enforce this policy in case of conflict or inconsistency.

Ⅵ. The related work

Using context information and semantic web technologies in access control mechanisms has been studied by different researchers. The related works is reviewed in three dimensions: semantic-based access control, context-aware access control and context ontology modeling, and semantic policy language framework.

In terms of context-aware access control and context ontology modeling, Chen et al. [16] concentrate on representing contexts in a formal way. This work serves as a very first approach in using semantic technologies for context representation. However, the system does not address the issues of context-based access control and how contexts can be integrated into a policy. One research work in the area of context-based access control is Ubiquitous Context-based Security Middleware (UbiCOSM) [17] that adopts a context as a principle for security policy specification and enforcement process. UbiCOSM adopts an RDF-based format but OWL-based format for context representation to cover heterogeneity of data representation. Toninelli et al. [18] suggest a semantic context-aware access control framework for secure collaboration in pervasive computing environments. They propose a simple OWL-based context model and based on this model, they propose a context-aware policy model and express policy statements using description logic but XACML rule. Filho and Martin [19] proposed a generalized context-based access control model for making access control decisions completely based on context information.

In terms of semantic-based access control, Naumenko et al. [20] propose to use semantic-based access control (SBAC) model for mobile web services. SBAC model is a result of introducing vocabularies and interpretations of specific security-related concepts inheriting all features of OWL and SWRL due to the compatibility with their direct model-theoretic semantics. Moussa et al. [21] present a semantic-based context-aware access control framework for semantic web. They propose a context ontology to represent contextual information and employ it in the inference engine. But don't handle the access control policy issue. Dersingh et al. [3] proposed a context-aware access control using semantic policies for autonomic computing; their object is to demonstrate how contexts can be captured and represented semantically, and integrated into an access control policy by extending the XACML.

In terms of semantic policy language, Kagal [22] proposed the Rei policy language which allows policies to be written using any semantic web language. Rei has been implemented in the N3 language and is called the Rein policy framework [23]. Rei and Rein serve as a foundation for semantic-based access control by allowing policy writers to interpret the meanings of the contexts within policies.

Another semantic policy language is KAoS [24] which is a framework for the specification, management, conflict resolution and enforcement of policies. KAoS policies are based on OWL. KAoS uses Description Logic (DL) mechanisms to reason over policies in order to check for applicable policy as well as to allow for the classification of policy statements to enable conflicts to be discovered. But DL-based reasoning may not always be sufficient.

Ⅶ. Conclusions

Open, dynamic, and heterogeneous mobile web services environments require new access control solutions, changing the focus of access control models from identity or role-based to the contextual-based approaches. By combining semantic web technologies (especially ontology technologies) with context-based access control mechanism, this paper presents the semantic-aware context-based access control model (SCBAC) for mobile web services. This paper develops a context ontology to represent the contextual information and a policy ontology to express access control policy in mobile web services environment. In addition, this paper also proposes an implementation framework for SCBAC. The major strength of SCBAC model is its ability to make access control decisions based on the context information, which can also be applied to context-aware applications. In the future work, policy adaptation will be a key issue to be discussed.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China under grant No. 60873024.

REFERENCES

[1] P. Farley, and M. Capp, “Mobile web Services,” BT Technology Journal, vol. 23, no. 2, pp. 202-213, April 2005

[2] N.S. Satish, J. Matthias, and P. Wolfgang, “Security analysis of mobile web service provisioning,” International Journal of Internet Technology and Secured Transactions, vol. 1, no. 1, pp. 151-171, 2007

[3] A. Dersingh, R. Liscano, and A. Jost, “Context-aware access control using semantic policies,” In Ubiquitous Computing And Communication Journal-Special Issue of Autonomic Computing Systems and Applications, 2008, pp. 1-14

[4] M. Liu, D. Xie, and P. Li, “Semantic access control for web services,” In proceedings of 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing, 2009, pp. 55-58

[5] S. Javanmardi, M. Amini, R.Jalili, and Y. Ganjisaffari, “SBAC: semantic based access control,” In proceedings of the 11th Nordic Workshop on Secure IT-Systems, 2006. pp. 157-168

[6] D.L. McGuinness, and F. van Harmelen, “OWL web ontology language semantics and abstract syntax,” 2004. http://www.w3.org/TR/owl-semantics/

[7] I. Horrocks, P.F. Patel-Schneider, and H. Boley, “SWRL: A Semantic Web Rule Language Combining OWL and RuleML (2004)”. http://www.w3.org/Submission/SWRL/

[8] A.K. Dey, “Understanding and using context,” Personal and Ubiquitous Computing, vol. 5, no. 1, pp. 4–7, Feb 2001

[9] M. Bazire and P. Br´ezillon, “Understanding context before using it,” In proceedings of 5th International and Interdisciplinary Conference on modeling and using Context, 2005, pp. 29–41.

[10] B.F. Eduardo, M. Maria, and E.E. Alvaro, “Contexts and context-based access control,” In proceedings of the Third International Conference on Wireless and Mobile Communications, IEEE Computer Society, 2007

[11] N. Malik, U. Mahmud and Y. Javed, “Future challenges in context-aware computing,” In proceedings of the IADIS International Conference WWW/Internet 2007, pp. 306–310.

[12] Protégé Editor and API. http://protege.stanford.edu/plugins /owl

[13] SPARQL Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query/, 2008.

[14] JESS: The Rule Engine for Java Platform. http://herzberg.ca.sandia.gov/jess

[15] J.R. Hobbs and F. Pan, “Time ontology in OWL,” September 2006. http://www.w3.org/TR/owl-time/

[16] H. Chen, T. Finin, and A.Joshi, “An ontology for context-aware pervasive computing environments,” Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review, vol. 18, no. 3, 2004, pp. 197-207

[17] A. Corradi, R. Montanari, and D. Tibaldi, “Context-based access control for ubiquitous service provisioning,” In proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC'04), 2004. pp. 444-451

[18] A. Toninelli, R. Montanari, L. Kagal, and O. Lassila, “A semantic context-aware access control framework for secure collaborations in pervasive computing environments,” In proceedings of the 2006 International Semantic Web Conference, 2006, pp. 473–486

[19] J. B. Filho and H. Martin, “A generalized context-based access control model for pervasive environments,” In proceedings of SIGSPATIAL ACM GIS 2009 International Workshop on Security and Privacy in GIS and LBS, 2009, pp.12-21

[20] A. Naumenko, S. Srirama, and V. Terziyan, “Semantic authorization of mobile web services,” Journal of Theoretical and Applied Electronic Commerce Research, vol. 1, no. 1, 2006, pp. 1-15

[21] A.E. Moussa, A. Morteza, and J. Rasool, “Handling context in a semantic-based access control framework,” In proceedings of the 2009 International Conference on Advanced Information Networking and Applications Workshops, IEEE Computer Society, 2009, pp. 103-108.

[22] L. Kagal, “A policy-based approach to governing autonomous behavior in distributed environments,” PhD thesis, University of Maryland Baltimore County, 2004

[23] L. Kagal, and T. Berners-Lee, “Rein: Where policies meet rules in the semantic web,” Technical report, MIT, 2005

[24] J. Uszok, R. Bradshaw, and N. Jeffers, “KAoS Policy and Domain Services: Toward a Description-Logic Approach to Policy Representation, Deconfliction, and Enforcement,” In proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'03), pp.93-96, 2003

Haibo Shen received his PhD degree in Information Security from Huazhong University of Science and Technology, Wuhan, China, in 2007.

His research interests are computer and network security. He is currently a professor in Computer Department of Hubei University of Technology.

Yu Cheng received the M.S. degree in Hubei University of Technology.

Her main research interests include network security and database technology.