IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITB12879

This paper appears in the book, *Emerging Trends and Challenges in Information Technology Management, Volume 1 and Volume 2*
edited by Mehdi Khosrow-Pour © 2006, Idea Group Inc.

Ontology-Based Security Specification Tools for SOA

Myong Kang, Anya Kim, Jim Lo, Bruce Montrose, & Amit Khashnobish
Center for High Assurance Computer Systems, Naval Research Laboratory, Washington DC 20375, T: (202) 767-3654,
{mkang, kim, luo, montrose, amith}@itd.nrl.navy.mil

ABSTRACT

Service Oriented Architecture (SOA) is an emerging paradigm for distributed computing. Web services and SOA aim to provide unobtrusive access to resources and data. However, risks and threats on the Internet compel us to protect and limit access to these same resources through security policies and mechanisms. Therefore, the success of SOA relies heavily on the ability to communicate the relevant security information required to access these resources in a machine- understandable manner.

In this paper, we introduce a set of (prototype) tools that enable the specification of ontology-based security information for each layer of the SOA. They build upon existing technology standards and enable dynamic discovery of Web services. Every well-thought out security requirement has some reasons/justifications behind it. However, oftentimes the justification or context is not well documented or disconnected from the requirement. Our tools will present an application-centric view of the SOA and capture the context information from which security requirements are derived.

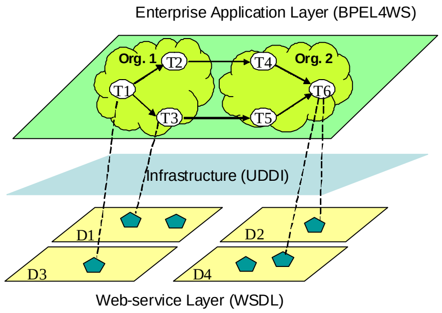
1. INTRODUCTION

Modern computer systems and networks are protected by a myriad of security measures necessary to defend against intruders, viruses and other threats in today's computing environment. However, these measures also act as barriers for access by legitimate users and potential customers. This is especially true in the context of Web services in a Service Oriented Architecture (SOA) where interoperability is commonly required across multiple domains. In order to match Web services and clients according to their security requirements and capabilities, security concepts need to be described precisely with sufficient detail to allow for interaction. To address this problem, we created the NRL Security Ontology that can precisely describe security requirements and capabilities, and support semantic matchmaking [1, 2].

SOA operations can be abstracted into three layers each with their own set of enabling technologies (Figure 1). At the bottom is the Web Service layer where individual Web services are described by their service providers. Accepted standards for Web service description include Web Service Description Language (WSDL) [3], and systems of categorization and identification developed by industry. At the top is the Enterprise Application layer where the business logic is captured and requirements for underlying Web services are derived. A current enabling technology for the Enterprise Application layer is Business Process Execution Language for Web Services (BPEL4WS)[1][4]. In between lies the Infrastructure layer where application level task requirements are matched to specific Web service descriptions[2]. Currently, Universal Description, Discovery and Integration (UDDI) [5] is the accepted registry standard for the Infrastructure layer. Note that the concept of layers is a relative one. For example, a Web service in Figure 1 may be a distributed process itself, thus becoming an application to its underlying Web services.

The problem with the existing SOA standards is that they do not support semantics. The enabling technologies in all three layers are syntax-based. However, semantic description and matchmaking are essential for security information as we explained in [1, 2]. Moreover, we feel that relatively little attention has been paid to the Enterprise Application layer. The focus has been on the Web services and Infrastructure layers with respect to describing and matchmaking of individual Web services. However, Web services rarely operate in isolation. They are usually parts of an overarching enterprise application and their requirements must be derived from the higher level business process. This context information provides the justification and rationale behind security decisions. For example, a process that spans competing organizations may require more stringent security requirements than one that spans organizations with an established trust relationship. Hence, context information must be captured and taken into account when developing the security requirements for the enterprise applications and underlying Web services. Although BPEL4WS is an industry standard to capture business logic, it cannot capture important context information for security descriptions. For example, since BPEL4WS has a centralized view, it neither captures application boundaries nor organization boundaries. Furthermore, BPEL4WS has no place to capture the security-relevant information such as organizations or hosts.

Figure 1. Service-oriented computing paradigm and associated standards



We have developed a set of tools that can be used in conjunction with the existing SOA standards on all three layers to add support for ontology-based security descriptions and semantic query processing. Our tools capture the context of the security specifications at the Enterprise Application level so that end users can make better decisions regarding security.

2. SECURITY ONTOLOGY AND UDDI

Security statements such as "this resource requires authentication via an X.509 version 3 certificate signed by Microsoft," or "this resource can use IPSec with a Type 3 certified TripleDES encryption algorithm, and the Diffie-Hellman key exchange algorithm to exchange this key" are complex descriptions that cannot be expressed using a taxonomy. Using OWL [6] we developed the NRL Security Ontology [1, 2] to express security concepts ranging from high-level security objectives to lowlevel protocols and mechanisms, credentials, security assurance levels and security algorithms in various levels of specificity.

The matchmaking aspect of security specifications is more complex than that of functional descriptions. While functional matchmaking is only concerned with functional capability, security matchmaking requires both the client and the (potential) service to possess a set of security requirements and capabilities. Security capabilities define the mechanisms that the entity possesses while the security requirements detail the mechanisms that the entity expects the other party to possess (and produce) to successfully interact. Therefore, a successful match requires two-way matching that one party's security requirements be compatible with the other's security capabilities and vice versa [1, 7, 8] as opposed to the one way match of functional capabilities [9].

UDDI is the de facto registry standard for SOA with widespread use in both government and industry. However, UDDI is currently not capable of handling ontology-based markups for Web services due to its incompatible data model and limited matchmaking capabilities [9-11]. We developed an approach to provide ontology support with registries that conform to the UDDI Version 3 specification by leveraging the versatility of tModels [5]. A special lossless transformation scheme was developed to store ontologies and semantic service descriptions in the UDDI data model [8]. This scheme captures the semantic relationships established by the ontology and fully supports composite concepts including anonymous instance extensions of the ontology defined by service descriptions. Semantic query processing can be performed for ontology-based service descriptions stored in the registry using a combination of the UDDI search engine and a special syntax-based client-side matchmaker that supports the two-way matching that we described above. Using our approach, UDDI registries will in effect behave as semantic registries capable of correctly storing and matching semantic service descriptions such as those using the NRL Security Ontology.

3. SECURITY SPECIFICATION FOR WEB SERVICES

We developed the Ontology-based Security Specification Tool (OSST) to aid in the composition of semantic Web service descriptions. Directly composing OWL-based security statements to annotate services can be tricky and requires the full understanding of OWL syntax as well as extensive knowledge of all the ontologies involved. Our tool simplifies the task by providing an Ontology Browser feature. Figure 2 shows the Ontology Browser providing a hierarchical view of ontological concepts along with associated properties and domain and range values. Having browsed the ontologies, users can use the Query Composer feature to describe complex security requirements and capabilities. Queries are created by selecting a high level concept, attaching a property from the provided list of properties that can be used to further refine that concept, then selecting a value from the list of concepts that can be attached to that property. The property concept in turn can be further refined by attaching additional property definitions until the security concept is fully expressed. Figure 3 shows the Query Composer used to create the statement from section 2, "IPSec with the Type 3 certified TripleDES encryption algorithm, and the Diffie-Hellman key exchange algorithm to exchange this key" as a security requirement for a Web service.

With this tool users can load the appropriate ontologies, select the class (concept) they want to specify, add property values to the class, create a query tree specifying whether this is a security requirement or capability, and search UDDI for services that match the query. These individual requirements and capabilities can also be given names and stored to be reused later. The tool can also be used to attach security requirements and properties to an UDDI Business Service.

Our prototype uses the Systinet UDDI registry [12] as a repository for ontologies and service descriptions. The difficulty of annotating Web services has been acknowledged and ways to facilitate annotation suggested previously [13, 14]. However, these tools either semantically annotate the WSDL description of the Web service [13] or annotate Web services using OWL-S [14]. While both have their merits, our approach is to attach these specifications directly to the UDDI Business Service. This tool was initially developed to annotate Web services with security-related information, but it can easily be used to annotate Web services with any type of information, as long as the proper ontology is loaded.

4. SECURITY SPECIFICATION FOR ENTERPRISE APPLICATIONS

Enterprise applications are not necessarily composed of a single business process. Several applications from different organizations may be strung together to create an inter-organizational enterprise application. To maintain the autonomy of these different organizations, the applications may form a peer-to-peer collaboration structure rather than having a master (top level) application that is composed of several slave applications (e.g., a simple Web service). For example, in Figure 1, we can view the enterprise application as being composed of two business processes (which themselves are enterprise applications for their own organizations). The asynchronous nature of the tasks is represented as a peer-to-peer structure. Each individual application may also be comprised of sub applications that string together several sub tasks to form it. For instance, task T5 from Figure 1 may be composed of several smaller subtasks (i.e., have corresponding Web services that are invoked in a logical sequence). In fact, each of these subtasks themselves may be composed of smaller tasks and so on and so forth.

These enterprise applications, like Web services also need security-related specifications, describing the security requirements and capabilities of the application so that they can communicate them to underlying Web services (and other business processes). However, because they have multiple contact points to outside agents, security specification of enterprise applications is more involved than that of simple Web services. First, the enterprise application is composed of several business processes that are in turn, composed of tasks that invoke lower-level services and transitions that pass necessary data between tasks. Each of these components can possess a different set of security requirements and capabilities. Second, there is other complex contextual information such as organization, location, security domain, etc. in an enterprise application that should be captured. This contextual information provides the framework for security policies and access control decisions, and also justifies the need for specific security requirements. Furthermore, peer-to-peer applications are, in general, characteristically asynchronous (as opposed to synchronous messaging between a high-level application and a Web service); an application could wait indefinitely to receive a response message from a peer. So survivability and QoS (quality of service) requirements also need to be factored.

Therefore, when adding security specifications at the Enterprise Application layer, we need a way to easily view application components and

Figure 2. OSST tool: Ontology browser feature

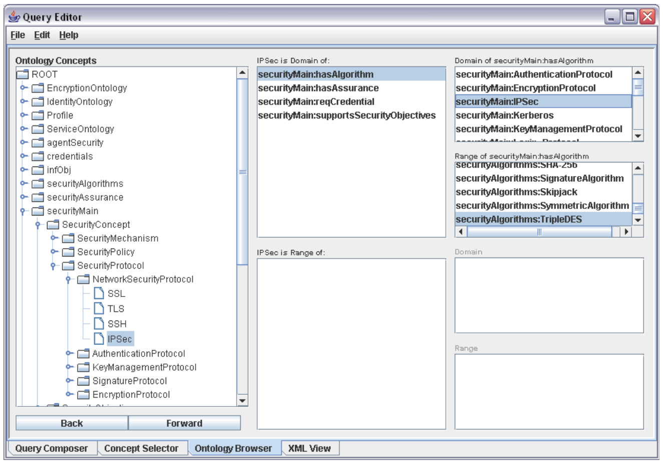
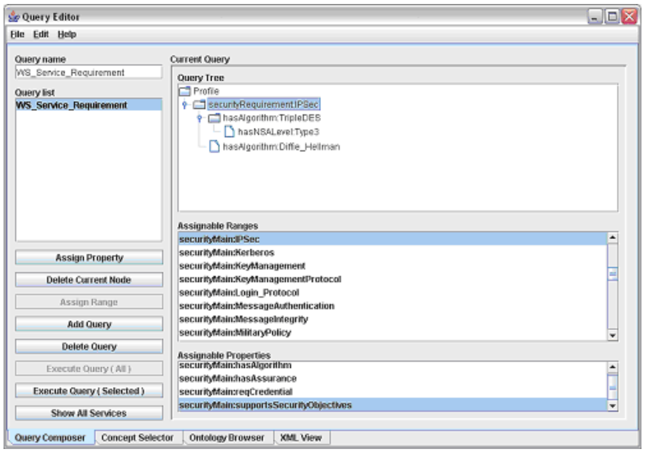


Figure 3. OSST tool: Query composer feature



Query name
WS_Service_Requirement

Query list
WS_Service_Requirement

Current Query

Query Tree
Profile
securityRequirement:IPSec
hasAlgorithm:TripleDES
hasNSALevel:Type3
hasAlgorithm:Diffie_Hellman

Assignable Ranges
securityMain:IPSec
securityMain:Kerberos
securityMain:KeyManagement
securityMain:KeyManagementProtocol
securityMain:Login_Protocol
securityMain:MessageAuthentication
securityMain:MessageIntegrity
securityMain:MilitaryPolicy

Assignable Properties
securityMain:hasAlgorithm
securityMain:hasAssurance
securityMain:reqCredential
securityMain:supportsSecurityObjectives

any contextual information. BPEL4WS is the de facto industry standard to capture business logic to form an enterprise application. However, it was not designed for security specification as it does not capture information such as organization or security domain. Also, when inter-enterprise applications are structured in a peer-to-peer format, BPEL4WS cannot provide an overall view of the application that is necessary for security specifications.

We developed a tool that we call the Context-aware Security Specification Tool (CaSST) for specifying security at the enterprise application layer. This tool renders a graphical view of the enterprise application that may span multiple organizations with contextual information, enables specification of (security) requirements and capabilities, and querying for underlying web services. While the business process is written in BPEL4WS, our tools take information embedded in BPEL4WS files to enable specification of security-related information at the Enterprise Application layer by providing an overall picture of the application that contains the following components:

- **Task:** Represents a component of an enterprise application that acts as a client to an underlying Web service. It can contain contextual information such as the application that it belongs to (e.g., a BPEL process), detailed information about the underlying Web service(s), etc.
- **Flow:** Represents data flow from one task to another. This flow may go across organizations that may require stronger security mechanisms than a flow within an organization.
- **Data:** Represents the data component, distinct from the transition on which the data flows. A transition may be associated with several different pieces of data or the same data may be associated with different transitions. Separating data from transition enables the specification of different security requirements for the data and the transition.
- **Application:** Represents a business process application (e.g., BPEL) that may hold contextual information such as the organization that runs this application, its security domain, etc.
- **Underlying Web service:** Represents a Web service that may itself be a BPEL process. It holds information such as the organization that it belongs to, security requirements and capabilities, input, output, etc.

The above components provide the overall picture and contextual information required to make appropriate security specifications to an enterprise application.

The tool extracts pertinent information from various BPEL files along with relevant contextual information into a Data Flow Diagram (DFD) format. The first component of CaSST, the DFD Context Editor renders a graphical view (i.e., network diagram) of the enterprise application displaying the components described above. Figure 4 is a snapshot of the DFD Context Editor displaying the sample enterprise application introduced in Figure 1. The two peer applications are differentiated by shades. The arrows show the transition (data flow) between the tasks. When clicked on, tasks and transitions provide further contextual information on the right such as information regarding application, organization, and underlying services. Transitions also provide information about specific data that is carried on that transition.

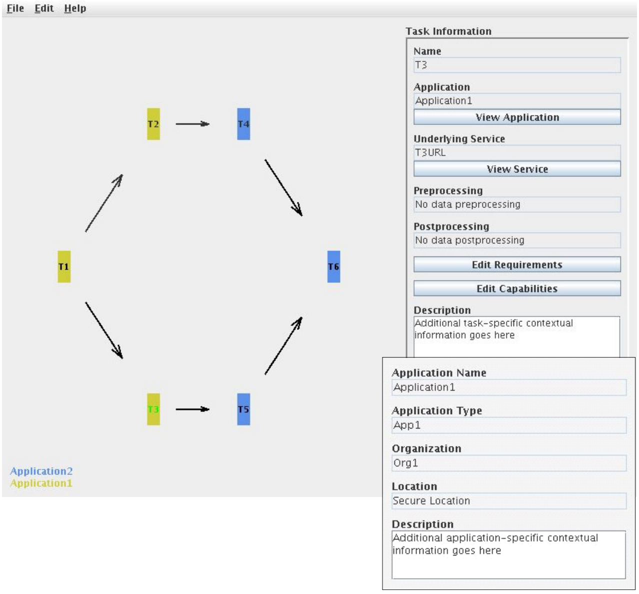
In particular, Figure 4 shows contextual information for task T3, such as the fact that it belongs to Application1, and that it has a predefined underlying Web service located at T3URL. The Description Field contains further contextual information that may be useful in making security-related decisions. Further information about the application or service can also be viewed by selecting the appropriate button. The pop-up window in Figure 4 shows additional information about the application that task T3 belongs to when the View Application button is clicked.

Each component of DFD Context Editor Tool has a button to edit security requirements or capabilities for that specific component. These buttons invoke the OSST to enable specification of (security) requirements and capabilities for the originating task, transition, data, application or Web service component. While similar to the OSST in section 3, it is not used independently but rather invoked by the DFD Context Editor Tool whenever security requirements or capabilities need to be added to an application component. This enables security specifications to be composed within the context of the enterprise application. In terms of how to use the tool, the features are similar to the independent OSST in Section 3, except that the security statement is not attached to an UDDI Business Service.

After adding security requirements for a specific component, users can search for potential services that match the given security requirement. Task requirements, as well as requirements from associated data and transitions for a given task are joined together to compose the query to match an underlying Web service. The backend query processing that extracts matching business services from UDDI consists of a UDDI query that returns candidate business services and the follow-on refined matching. Figure 5 shows the results of querying for Web services. It displays the possible services that match the query, and the actual business description of each service retrieved from UDDI. A detailed description of the search mechanism can be found in [8].

Besides providing users with a contextual view of the enterprise

Figure 4. CaSST: DFD context editor feature



Task Information

Name
T3

Application
Application1

Underlying Service
T3URL

Preprocessing
No data preprocessing

Postprocessing
No data postprocessing

Description
Additional task-specific contextual information goes here

Application Name
Application1

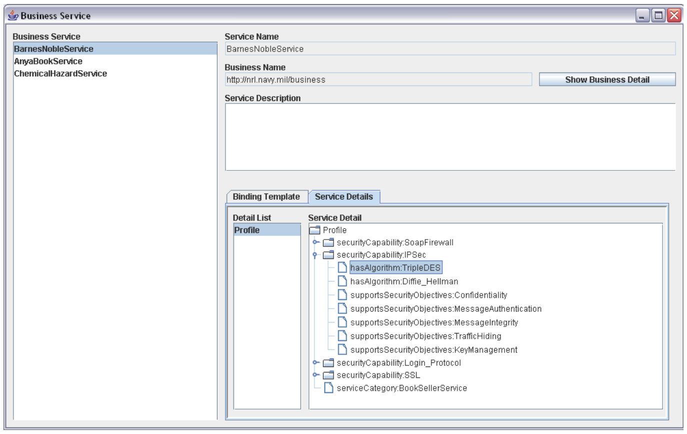
Application Type
App1

Organization
Org1

Location
Secure Location

Description
Additional application-specific contextual information goes here

Figure 5. CaSST: Business service search results



Business Service
- BarnesNobleService
- AnyaBookService
- ChemicalHazardService

Service Name: BarnesNobleService

Business Name: http://nrl.navy.mil/business

Service Description:

Detail List: Profile

Service Detail:

- Profile
  - securityCapability:SoapFirewall
  - securityCapability:IPSec
    - hasAlgorithm:TripleDES
    - hasAlgorithm:Diffie_Hellman
    - supportsSecurityObjectives:Confidentiality
    - supportsSecurityObjectives:MessageAuthentication
    - supportsSecurityObjectives:MessageIntegrity
    - supportsSecurityObjectives:TrafficHiding
    - supportsSecurityObjectives:KeyManagement
  - securityCapability:Login_Protocol
  - securityCapability:SSL
  - serviceCategory:BookSellerService

application to facilitate security specifications, the tools can be used to bridge the gap between the operational people that understand the business logic and the security people that understand the security requirements. Both parties may want to add security requirements into the enterprise application but from different perspectives. The operational people can use the tool to specify security requirements on the application from a high-level perspective and also to better communicate application context to the security people. The security people, with the aid of the visual tool can better understand the application logic add more specific security requirements in a context-appropriate manner.

The strength of this tool is that it uses the DFD format that is independent of other standards that focus on functional aspects of the enterprise applications such as BPEL4WS or WS-CDL [15]. DFD is abstract and minimal enough that we expect the tool can be used in conjunction with a variety of business process standards. For instance, a skeleton of a DFD specification can be derived from enterprise applications written in WS-CDL if a WS-CDL specification already exists. On the other hand, if users start with DFD and the GUI tools that we provide, they can easily generate skeletons of BPEL4WS or WS-CDL specifications.

5. CONCLUSION

Annotation of SOA components with security information is an important milestone to achieving truly secure and survivable enterprise applications. Our approach is to provide semantic markup with security ontologies to dynamically integrate and compose services in the SOA framework. This work affects every layer in the SOA architecture from security markup of an application composed of multiple services to the security specification of individual services themselves, and the development of infrastructures to support semantic querying and matchmaking.

State-of-the-art technology standards for SOA are not readily usable to specify security. However, they are backed and widely used by big players in the industry. Therefore, our work is based on these standards, but extends and expands them in a way that enables specification of security information with a set of tools. The tools we developed are still in the prototype stage. We expect that once they've been tested extensively, we will discover additional security-related features to include and limitations to overcome.

6. REFERENCES

1. Kim, A., Luo, J., and Kang, M. (2005). Security Ontology for Annotating Resources. In *4th International Conference on Ontologies, Databases, and Applications of Semantics (ODBASE'05)*, Springer-Verlag: Agia Napa, Cyprus.
2. Kim, A., Luo, J., and Kang, M. (2005). Security Ontology for Annotating Resources. pp. 51, Naval Research Lab, NRL Memorandum Report, NRL/MR/5540-05-641: Washington, D.C.
3. Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S. (March 2001). Web Services Description Language (WSDL) 1.1, http://www.w3.org/TR/wsdl.
4. Andrews, T., Curbera, F., Goland, Y., Klein, J., Leymann, F., Roller, D., Thatte, S., and Weerawarana, S. (2003). Business Process Execution Language for Web Services, version 1.1, http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/.
5. OASIS UDDI Spec Technical Committee (2004). UDDI Version 3.0.2, http://uddi.org/pubs/uddi_v3.htm.
6. W3C Recommendation (2004). OWL Web Ontology Language Guide, vol. 2005, W3C.
7. Denker, G., Kagal, L., Finin, T., Paolucci, M., and Sycara, K. (2003). Security for DAML Web Services: Annotation and Matchmaking. In *Proc. of the 2nd International Semantic Web Conference (ISWC2003)*: Sanibel Island, Florida.
8. Luo, J., Montrose, B., and Kang, M. (2005). An Approach for Semantic Query Processing with UDDI. In *1st International Workshop on Agents, Web Services, and Ontologies Merging (AWeSOMe'05)*, Springer-Verlag: Agia Napa, Cyprus.
9. Paolucci, M., Kawamura, T., Payne, T.R., and Sycara, K. (2002). Semantic Matching of Web Services Capabilities. In *1st Intl. Semantic Web Conference*: Sardinia, Italia.
10. Paolucci, M., Kawamura, T., Payne, T.R., and Sycara, K. (2002). Importing the Semantic Web in UDDI. In *Web Services, E-business and Semantic Web Workshop (ESSW02)*.
11. Srinivasan, N., Paolucci, M., and Sycara, K. (2004). Adding OWL-S to UDDI, Implementation and Throughput. In *First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)*: San Diego, California.
12. Systinet (2005). Systinet Registry, 6.0 ed: Burlington, MA.
13. Patil, A., Oundhakar, S., Sheth, A., and Verma, K. (2004). METEOR-S Web service Annotation Framework. In *International WWW Conference*: New York, NY.
14. Scicluna, J., Abela, C., and Montebello, M. (2004). Visual Modelling of OWL-S Services. In *IADIS International Conference WWW/Internet*: Madrid, Spain.
15. W3C Recommendation (2005). Web Services Choreography Description Language Version 1.0, W3C.

ENDNOTES

[1] The Web Services Choreography Description Language (WS-CDL) is the corresponding W3 standard.

[2] In general, the tasks in the enterprise application layer are representations of underlying Web services.