Ontology-based Security Policies for Supporting the Management of Web Service Business Processes

Diego Zuquim Guimarães Garcia, Maria Beatriz Felgar de Toledo
Institute of Computing
University of Campinas, São Paulo, Brazil
{diego.garcia,beatriz}@ic.unicamp.br

Abstract

Typically, in areas such as e-business and e-government, among others, Web services are used as basic components for building business processes. Participants in a business process may have different computational platforms that should interoperate in order to achieve the process goals. This interoperability is supported by the Web service technology. Thus, the importance of the technology is growing and its use in these areas demands security concern. However, the current approach for building processes from services, based on the Web Services Business Process Execution Language (WS-BPEL), does not consider security. This paper proposes an approach for building processes according to provider capabilities and consumer security requirements. These characteristics are expressed using Web Services Policy Framework (WS-Policy) policies and a Web Ontology Language (OWL) ontology. The main contribution of this paper is the use of semantics-enriched security policies for enriching Web service business processes.

1. Introduction

Significant progress has been done towards making the Web service technology a suitable solution for areas such as e-business, e-government, among others. For instance, the Web service technology enables the building of business processes by composing Web services. However, there are still open issues hindering the wide scale deployment of the technology [30].

This paper focuses on the security of Web service business processes, specifically, building processes that have security characteristics compliant with consumer requirements. With the growing importance of Web service business processes in areas such as e-business, the inclusion of security management support becomes a critical factor for success.

Several solutions have been proposed to deal with service composition. For example, some languages for service orchestration and choreography have been developed. Among them, the Web Services Business Process Execution Language (WS-BPEL) [2] is a popular language for orchestration.

WS-BPEL has been developed by a group of software companies. Recently, it has been defined as an OASIS (Organization for the Advancement of Structured Information Standards) standard. However, the current approach for building business processes from Web services supported by WS-BPEL does not consider security issues.

The goal of this paper is to propose an approach that considers provider security capabilities and consumer security requirements during business process building. It extends WS-BPEL to allow the description of consumer security requirements and the selection of Web services with suitable security capabilities. Thus, it enables the building of Web service business processes that are secure according to consumer security requirements.

The approach supports the composition of services compliant with the Web Services Security - SOAP Message Security (WS-Security) [21] standard. This is achieved by enriching the description of a service with its security capabilities. Security capabilities and requirements are expressed using Web Services Policy Framework (WS-Policy) [4] policies based on a Web Ontology Language (OWL) [23] security ontology.

The use of an ontology is important because the WS-Policy syntactic approach may restrict the selection of suitable Web services. This limitation is overcome by policy annotations based on the ontology. Thus, policy specifications offer semantic information about security requirements and capabilities. This information can be used to verify policy compatibility and to guarantee that business processes are built in a secure manner.

The paper major contribution is the use of ontology-based policies for considering security characteristics during the building of Web service business processes.

978-0-7695-3279-0/08 $25.00 © 2008 IEEE
DOI 10.1109/ICSC.2008.35

The rest of the paper is organized as follows. Section 2 presents basic concepts. Section 3 provides an overview of the proposed approach. The use of ontology-based policies in business processes and an extension to WS-BPEL are described in Sections 4 and 5, respectively. Section 6 discusses related work. Finally, Section 7 closes the paper with conclusions.

2. Basic concepts

2.1. Web services and business processes

A Web service is an electronic service identified by a URI (Uniform Resource Identifier). The Web service technology comprises three basic standards [1]:

- Web Services Description Language (WSDL): format for describing Web service functionality.
- Universal Description Discovery & Integration (UDDI): registry for service publication and discovery.
- SOAP (formerly Simple Object Access Protocol): protocol for message exchange among services.

Consumers may require a service implemented by a single Web service or by means of a business process. A business process is a service composition. WS-BPEL [2] is a language for describing Web service business processes. Descriptions are based on interactions between the process and its participants. The interaction with each participant occurs through a Web service interface.

2.2. Web service policies and security

Additional standards for Web services are under development. One example is WS-Policy [4]. It provides a model for expressing service properties as policies. Policies can be associated with XML elements, as defined in the Web Services Policy Attachment (WS-PolicyAttachment) [3] specification. A policy is a collection of alternatives and each alternative is a collection of assertions. Assertions specify characteristics that are critical to service selection and use, for instance, security characteristics.

In Web services, mechanisms to protect SOAP messages are defined in the WS-Security [21] specification. These mechanisms include digital signature, to protect against inappropriate message alteration, and encryption, to deal with incorrect message disclosure.

To guarantee message integrity, the digital signature mechanism uses the XML Signature [5] standard along with security tokens. A token is a collection of claims. A claim is a statement made by an entity, for instance an identity or capability statement.

Encryption is based on the XML Encryption [14] standard and tokens to offer message confidentiality.

Other standards focus on different security aspects [11, 12]. For instance, the Web Services Security Policy Language (WS-SecurityPolicy) provides a WS-Policy assertion set to describe how services work in terms of security. Typically, security policies are complex. Therefore, the mechanism for expressing policies must allow precise specifications. However, information provided by WS-SecurityPolicy does not include explicit meaning.

2.3. The Semantic Web

The Semantic Web is described as a World Wide Web evolution in which information available on the Web includes machine-accessible semantics for increasing information processing automation and improving information system interoperability [25].

It includes several standards. OWL [23] is a standard ontology language. An ontology represents the meaning of terms in vocabularies and their relationships. OWL extends the RDF Schema (RDFS) and provides additional vocabulary along with formal semantics for increasing semantics expressiveness.

3. Secure Web service business processes

This section presents an overview of the approach for building secure Web service business processes.

In the approach, service capabilities and service consumer requirements regarding security are used as the base for building Web service business processes that satisfy consumer security requirements.

The current Web service architecture relies on WSDL to describe Web services. However, WSDL descriptions are based on service functionality.

WS-Policy may be used to improve Web service descriptions by considering aspects that are not directly related to service functionality, including Quality of Service (QoS) characteristics. It is a candidate to become a future standard for Web service policy specification due to its flexibility and extensibility.

Therefore, in the approach, WS-Policy is employed to complement WSDL descriptions with the inclusion of Web service QoS characteristics. Particularly, WS-Policy policies specify security characteristics. However, the same general approach may be applied to deal with other QoS characteristics, for instance, reliable messaging.

Both service capabilities and consumer requirements regarding security are specified using policies.

WS-Policy security policies are attached to WSDL service interfaces. Thus, Web services have security policies associated with them and consumers are able to select services considering their security requirements.

Security capabilities of services could be expressed directly on WSDL files. However, WSDL is designed to encode service functionality. Typically, functional aspects are more fixed than non-functional aspects. Flexibility is achieved by using separate files to encode QoS. QoS characteristics can change without changing WSDL files.

Consumers have to describe their required services by means of service compositions. In order to perform this task, they use WS-BPEL to describe business processes, according to the current approach for building Web service business processes.

To state their security requirements with respect to business processes, consumers use policies in WS-Policy and attach them to WS-BPEL process descriptions.

A consumer may specify policies with security requirements for the services that compose a business process. It is also possible to specify security policies with requirements for the whole business process.

Specifications of security requirements and capabilities are based on an OWL security ontology.

The use of semantics-enriched policies in the building of Web service business processes and the extension to WS-BPEL are described in the next sections.

4. Semantics-enriched security policies

In this section, a security description mechanism for the approach is presented. It is based on an ontology.

4.1. Security ontology

The ontology includes concepts for protecting Web service message exchanges. It supports a high abstraction level for dealing with security goals. In Figure 1, the main ontology classes and their relationships are presented. Properties and other restrictions are not shown. The classes are created to be equivalent to some XML elements of the WS-Security [21], XML Signature [5] and XML Encryption [14] standards.

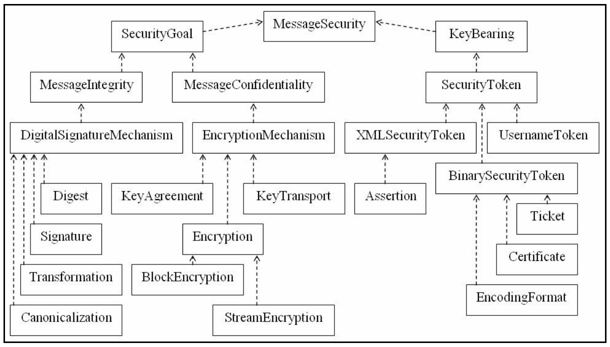

Figure 1. Security ontology.

In the ontology, the top-level class is called *MessageSecurity*. This class has some properties, including *keyBearing* of the *KeyBearing* type and *securityGoal* of the *SecurityGoal* type.

The *KeyBearing* class represents mechanisms for bearing security keys. An example of a key bearing mechanism is defined by the *SecurityToken* class, which is a subclass of the *KeyBearing* class.

The *SecurityGoal* class represents message security goals, including message integrity and confidentiality. These goals are captured in the ontology by defining two *SecurityGoal* subclasses: *MessageIntegrity* and *MessageConfidentiality*.

The digital signature mechanism is associated with the message integrity goal as a technique for achieving it. This mechanism is represented by the *DigitalSignatureMechanism* class, which includes properties of the following types:

- *Signature*: signature algorithms are represented by instances of this class, including DSA-SHA1 (Digital Signature Algorithm - Secure Hash Algorithm) and RSA-SHA1 (Rivest Shamir Adleman - SHA).
- *Digest*: digest algorithms are captured by this class, which includes instances, such as SHA1, SHA256 and SHA512.
- *Canonicalization*: this class includes the XML Canonicalization and Exclusive XML Canonicalization instances.
- *Transformation*: this class specifies transformation algorithms and includes instances, such as XSLT (eXtensible Stylesheet Language Transformation), XPath (XML Path Language), Enveloped Signature, SOAP Message Normalization and SecurityTokenReference Dereference Transform.

The encryption mechanism is associated with the message confidentiality goal. The *EncryptionMechanism* class represents this mechanism and includes properties of the following types:

- *Encryption*: this class specifies encryption algorithms. Two specializations are defined to represent block and stream encryption algorithms. The *BlockEncryption* class includes some instances, such as 3DES (Triple Data Encryption Standard), AES-128 (Advanced Encryption Standard), AES-192 and AES-256.
- *KeyTransport*: key transport algorithms are represented by instances of this class, including RSA-v1.5 and RSA-OAEP (RSA - Optimal Asymmetric Encryption Padding).
- *KeyAgreement*: this class defines key agreement algorithms. It includes the Diffie-Hellman instance.

Signature and encryption use security keys. Tokens are used to hold keys within or outside messages. There are different types of tokens with different manners of attaching them to messages. The *SecurityToken* class includes three token type subclasses:

- *UsernameToken*: username tokens offer a means of providing usernames to use Web services.
- *BinarySecurityToken*: this token type includes binary-formatted security tokens.
- *XMLSecurityToken*: this token type includes XML-based security tokens.

The *BinarySecurityToken* class has an *encodingFormat* property that indicates the token encoding format. For instance, the base64 encoding format is represented by the *Base64* instance. Two classes are specified for binary tokens: *Certificate* and *Ticket*, which define the certificate and ticket concepts, respectively. Moreover, specializations are defined, including *X.509Certificate* for the *Certificate* class and *KerberosTicket* for the *Ticket* class.

The *X.509Certificate* class includes several instances, which represent X.509 versions: X.509 Version 3, X.509 PKCS7 (Public-Key Cryptography Standards), X.509 PKI (Public-Key Infrastructure) Path Version 1 and X.509 Version 1. Some *KerberosTicket* instances are also defined, including instances that represent Kerberos Version 5 AP-REQ (Application Request) and GSS (Generic Security Service) Kerberos Version 5 AP-REQ.

The same scheme is used for XML tokens. The *Assertion* class is a *XMLSecurityToken* subclass. It represents security assertions. *SAMLAssertion* is defined as a specialization for this class. *SAML-v1.1* and *SAML-v2.0* are specified as instances of the *SAMLAssertion* class. These instances represent different versions of the Security Assertion Markup Language (SAML), including SAML Version 1.1 and SAML Version 2.0.

The ontology offers a flexible approach to support interoperability, which is a requirement in Web service environments. It can be extended with additional message security techniques and technologies by including new classes and properties.

4.2. Security policy

In the proposed approach, the specification and enactment of Web service business processes are based on policies.

At specification time, service providers define policies describing security properties of their Web services. Moreover, service consumers define policies stating security properties that should be offered by Web services included into business processes. These policies are enriched with annotations based on the security ontology.

At enactment time, the policies of a provider and a consumer are intersected to compute an effective security policy. This policy indicates the interoperability between the participants in terms of security.

The ontology offers a base for reasoning over policy specifications. Therefore, it supports rich policy intersections to guarantee the building of business processes with suitable security levels for the consumers.

The basic structure of policies is compliant with the WS-Policy normal form, which is shown in Figure 2.

```
<p:Policy>
   <p:ExactlyOne>
      ( <p:All>
         ( <Assertion ...> ... </Assertion> )*
      </p:All> )*
   </p:ExactlyOne>
</p:Policy>
```

Figure 2. Basic policy structure.

In Figure 2, *p* is a prefix for the WS-Policy namespace URI. In addition to the components included into the normal form, other general-purpose components can facilitate policy manipulation. A policy includes the following components:

- *Policy*: the root element that indicates a policy.
- *Name*, *Id*: two kinds of policy identification may be used. Either the policy is associated with an absolute URI, using the *Name* attribute, or it is associated with a reference within the enclosing document, using the *Id* attribute.
- *PolicyReference*: the *PolicyReference* element may be used to include the content of a policy into another policy.
- *Service*: a provider policy includes a *Service* element to describe details of the service implementation for which the policy has been specified. A consumer policy includes this element to specify details of the business process or component service to which the policy applies.
- Operators: in a policy, policy alternatives are grouped into an *ExactlyOne* operator. The *All* operator represents a policy alternative and groups the alternative assertions.
- Assertions: policy assertions are elements that represent consumer security requirements and service security capabilities. A policy assertion may contain nested assertions and a nested policy.

It is in the assertion components that a policy is specialized. Assertions use concepts from the security ontology in opposition to the current WS-Policy approach, which uses assertions specified in WS-Policy supplementary specifications. Thus, the ontology defines a common security vocabulary that is shared among service consumers and providers.

Figures 3 and 4 show examples of assertions extracted from a policy for a service implementation.

Figure 3 presents a token assertion. It indicates that the service uses a X.509 Version 3 token (Line 01) with the base64 format (Line 03). The *Id* attribute (Line 02) specifies the local identification of the token element. The *sec* and *u* prefixes are associated with the namespace URIs of the OWL security ontology and the WS-Security-Utility XML Schema definition, respectively.

```
01  <sec:X.509-v3
02     u:Id = "X.509Token"
03     EncodingFormat = "sec:Base64"/>
```

Figure 3. Token assertion example.

The encryption assertion in Figure 4 indicates that the body of messages (Line 06) sent by the service is encrypted using the 3DES algorithm (Line 01). Line 03 shows that the encryption mechanism uses the token defined in Figure 3.

```
01  <sec:3DES>
02     <sec:Token>
03        <sec:Reference URI = "#X.509Token"/>
04     </sec:Token>
05     <sec:EncryptedParts>
06        <sec:Body/>
07     </sec:EncryptedParts>
08  </sec:3DES>
```

Figure 4. Encryption assertion example.

Policy operations defined in the WS-Policy specification may be used for processing security policies. For example, the intersection operation is used to determine services whose security policies are suitable for a given consumer policy.

The intersection operation matches consumer and provider policies. Policies are compatible if there is at least one pair of compatible alternatives between them. Policy alternatives are compatible if the capability assertions of one alternative satisfy the requirement assertions of the other alternative. The compatibility between assertions is determined by using OWL-based operators.

Thus, security domain knowledge can be considered. For instance, if a service consumer includes the *MessageConfidentiality* ontological concept into the policy associated with a component service of a business process, then the required service must offer message confidentiality protection. The service implementation for which the policy assertion in Figure 4 was defined is considered a suitable service for this consumer. The domain knowledge captured by the security ontology allows determining that the implementation supports message confidentiality, as required by the consumer.

5. WS-BPEL extension

This section describes the mechanisms for the specification and enactment of business processes.

5.1. Business process specification

A consumer describes a business process using WS-BPEL. In order to include the consumer security requirements into the process description, the WS-BPEL description is extended with references to policies.

Service consumers may specify policies with general and specific security requirements for business processes. A policy with specific requirements applies to a specific Web service included into a business process. Policies with general requirements apply to the whole business processes.

When a policy is included into the context of a service in a process, it is enforced during the selection of an implementation for the service to which it applies.

Below, Figure 5 shows an example of a requirement extracted from a policy. The policy is associated with a service, which is a component of a business process.

```
01  <sec:MessageConfidentiality>
02     <sec:EncryptedParts>
03        <sec:Body/>
04        <sec:SecurityHeader/>
05     </sec:EncryptedParts>
06  </sec:MessageConfidentiality>
```

Figure 5. Specific requirement example.

The requirement in Figure 5 indicates that the service must guarantee the confidentiality (Line 01) of the body (Line 03) and security headers (Line 04) of its messages.

In the case of a general security policy, that is, a policy with security requirements that apply to a whole business process, the policy is enforced during the selection of each service included into the process.

An example of a requirement extracted from a general security policy is shown in Figure 6. The policy is associated with all the services that compose a process.

```
01  <sec:DSA-SHA1>
02     <sec:SignedParts>
03        <sec:Body/>
04        <sec:TimestampHeader/>
05     </sec:SignedParts>
06  </sec:DSA-SHA1>
```

Figure 6. General requirement example.

Figure 6 includes a security requirement of message integrity protection, which indicates that the Web services must sign the body (Line 03) and the time-stamp header

(Line 04) of messages using the DSA-SHA1 algorithm (Line 01).

Security policies with requirements for both specific services and whole business processes are attached to WS-BPEL descriptions according to a mechanism defined by the WS-PolicyAttachment standard.

WS-PolicyAttachment defines the *PolicyURIs* attribute to attach WS-Policy policies to an arbitrary XML element. In the approach, *PolicyURIs* attributes allow security policies to be attached to WS-BPEL elements that represent business processes and services included into processes.

The *PolicyURIs* attribute includes a list of URIs. Each URI identifies a WS-Policy policy. If more than one policy is listed in a *PolicyURIs* attribute, the policies need to be merged to form a single policy. The resultant policy is then associated with the element that contains the *PolicyURIs* attribute. The policy merge operation in the WS-Policy standard combines policies into a single policy, which includes the content of the original policies.

Below, examples of policy attachments extracted from a business process are shown. The *b* and *a* prefixes are associated with the namespace URIs of WS-BPEL and the application example, respectively.

Figure 7 shows a fragment of a business process description with a component service. The service has a policy associated with it (Line 07). This policy specifies security requirements applied only to the operation (Line 04) invoked on the service (Line 03). This service is provided by the participant (Line 02) indicated in the invocation activity described in this example.

```
01  <b:invoke
02      partnerLink="Seller"
03      portType="a:Purchasing"
04      operation="Purchase"
05      inputVariable="sendOrder"
06      outputVariable="getResponse"
07      p:PolicyURIs="…securityPolicy">
```

Figure 7. Specific policy attachment example.

An example of a general policy attachment, that is, the attachment of a policy that applies to a business process as a whole, is shown in Figure 8.

```
01  <b:process
02      name="purchaseOrderProcess"
03      targetNamespace="…purchase"
04      p:PolicyURIs="…policies#Tokens
05          …policies#Integrity">
```

Figure 8. General policy attachment example.

The process description fragment shown in Figure 8 indicates that the process has two policies. One policy includes requirements regarding tokens (Line 04) and the other includes requirements concerning message integrity (Line 05). The policies must be enforced for all the services that compose the process (Line 01).

Business process policies must be combined into one resulting policy. The same applies to service policies. When both policy types are specified, the process policy must be merged with the service policy, for each component service. The policy resultant from this process is used for selecting a service implementation.

5.2. Business process enactment

In the approach, business processes are enacted by a business process engine and a broker. The engine uses the broker to select service implementations in a service registry, as shown in Figure 9.

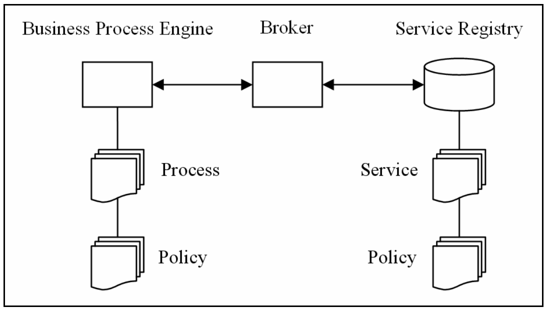

Figure 9. Business process enactment.

During process enactment, when a service associated with a security policy is reached, the process engine sends to the broker a request for a service implementation. The request includes functional requirements that indicate the type of implementation required to provide the service. It also includes the policy associated with the service that indicates the consumer requirements, including general requirements and requirements for the specific service.

The broker selects a service implementation considering the consumer security policy and the policies of the service implementations registered in the registry. In order to select a service, the broker discovers services with the required functionality in the UDDI registry. This is done by using the UDDI discovery mechanism. Then, a service among the ones discovered in the first step is selected intersecting security policies.

To build a business process that satisfies the consumer security requirements, for each service included into the business process, the consumer policy is intersected with the provider policies that describe the security capabilities of the services found in the first step.

Thus, the broker selection is based on functionality and security policy. After selecting services, the broker sends their addresses to the business process engine, which invokes the services.

6. Related work

Much research has been done in the area of Web service QoS. There are studies that deal with security of single services and security in the context of business processes. Works considering both are discussed below.

In [24], a Web service architecture employs brokers to support service selection based on QoS. Maximilien and Singh [19] propose the use of a QoS ontology together with a QoS policy language.

Web service QoS ontologies are described in [31, 10]. The DAML-QoS ontology [31] is realized using DAML+OIL, a language built from the DARPA Agent Markup Language (DAML) in an effort to combine components of the Ontology Inference Layer (OIL). QoSOnt [10] is realized using OWL. It uses the power of knowledge representation in OWL to allow reasoning.

Kagal et al [17] use the Semantic Web technology to handle authorization and privacy policies for Web services. An approach based on domain knowledge is proposed. Shields et al [26] present an approach for the specification of access control policies using an ontology.

These contributions do not consider Web service business processes. There has been a growing interest in service composition based on QoS attributes, as shown by the following studies.

QoS-oriented and broker-based frameworks are proposed in [22, 29]. They perform Web service selection for improving business process QoS.

In [15], a mechanism for determining the QoS of a service composition uses workflow patterns to represent structural elements of compositions. It is employed to monitor business process enactment [16].

Montagut and Molva [20] present a process to build transactional processes. The business process considers transactional requirements defined by service consumers.

Transaction processing and security, for instance, are general QoS attributes, that is, they apply to services of different domains. The use of non-functional attributes specific to a service domain can also be considered to compose services. In [9], an approach that allows the use of application-specific QoS attributes is proposed.

These studies do not focus on security, an aspect that has not been deeply investigated so far. Recently, some efforts have been conducted in this area. Below, they are presented.

Approaches based on extended calculus [6, 28] are proposed to specify security characteristics and select services that satisfy security requirements of processes.

Karjoth et al [18] introduce the concept of service-oriented assurance, in which services articulate their offered security capabilities. Services with specified capabilities provide guarantees about their security features. Thus, it is possible to build business processes by selecting services with required security levels.

Bartoletti et al [7] and Han and Khan [13] present frameworks for security-oriented service composition using different techniques for verifying if service compositions meet security goals.

An approach to enhance the security of business processes is proposed in [27]. Service consumers can specify security policies, which may be joined into business processes at runtime.

A method using the Semantic Web technology for modeling security requirements and a mechanism that considers them to build processes are presented in [8].

These solutions do not support integration into the Web service architecture. Differently from this paper, some important standards are not used:

- WS-Policy: WS-Policy offers a flexible framework for expressing Web service policies. It has been submitted to the World Wide Web Consortium (W3C) for standardization. Policies are a suitable choice for specifying security characteristics and policy intersection may improve service selection.
- OWL: OWL is a W3C recommendation. Ontologies may be used to capture security semantics. Policy intersection considering only the syntax of policies may not identify all possible effective policies.
- WS-Security: WS-Security is an OASIS standard that provides an extensible framework for message security. It supports the use of different security models, which is a requirement for Web services.

7. Conclusions

The Web service technology offers benefits to areas such as e-business, e-government, among other areas, due to its interoperability support. An important benefit provided by the Web service technology is the possibility of building business processes by composing Web services. However, there are still open issues hindering its wide scale deployment.

Particularly, in the context of Web service business processes, there is still a lack of facilities to deal with security issues. For instance, the current approach for building Web service business processes, supported by WS-BPEL, does not consider security characteristics.

In this paper, an approach that combines WS-BPEL, WS-Policy and OWL was introduced to build business processes that are secure according to consumer requirements. Policies are used to specify service security capabilities and consumer security requirements in business processes, and to select service implementations according to security policies for business process enactment. A security ontology helps specifying semantics-enriched policies.

The main contribution of this paper is the use of semantic policies to enable the consideration of security characteristics during the building of business processes.

Future work includes the execution of tests to evaluate the approach in scenarios with different security constraints. Moreover, the use of the approach with other QoS attributes and the inclusion of facilities for mobile Web services may also be considered.

Acknowledgements

This project is supported by FAPESP.

8. References

[1] G. Alonso, F. Casati, H. Kuno, and V. Machiraju. *Web Services: Concepts, Architectures and Applications*. Springer, 2004.

[2] A. Alves, et al. WS-BPEL Version 2.0. OASIS, Apr. 2007. http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html, accessed on 11/2007.

[3] S. Bajaj, et al. Web Services Policy 1.2 - Attachment. W3C, Apr. 2006. http://www.w3.org/Submission/2006/SUBM-WS-PolicyAttachment-20060425/, accessed on 11/2007.

[4] S. Bajaj, et al. Web Services Policy 1.2 - Framework. W3C, Apr. 2006. http://www.w3.org/Submission/2006/SUBM-WS-Policy-20060425/, accessed on 11/2007.

[5] M. Bartel, J. Boyer, B. Fox, B. LaMacchia, and E. Simon. XML Signature. W3C, Feb. 2002. http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/, accessed on 11/2007.

[6] M. Bartoletti, P. Degano, and G. L. Ferrari. Enforcing Secure Service Composition. In *Proc. of the IEEE Workshop on Computer Security Foundations*, pages 211-223. IEEE, 2005.

[7] M. Bartoletti, P. Degano, and G. L. Ferrari. Plans for Service Composition. In *Proceedings of the Workshop on Issues in the Theory of Security*, pages 20-35. 2005.

[8] R. Bishop, B. Carminati, E. Ferrari, and P. C. K. Hung. Security Conscious Web Service Composition with Semantic Web Support. In *Proc. of the Workshop on Security Technologies for Next Generation Collaborative Business Applications*. 2007.

[9] G. Canfora, M. Di Penta, R. Esposito, F. Perfetto, and M. L. Villani. Service Composition (re)Binding Driven by Application-Specific QoS. In *Proc. of the Intl. Conference on Service-Oriented Computing*, pages 141-152. IEEE, 2006.

[10] G. Dobson, R. Lock, and I. Sommerville. QoSOnt: A QoS Ontology for Service-centric Systems. In *Proceedings of the EUROMICRO Conference on Software Engineering and Advanced Applications*, pages 80-87. IEEE, 2005.

[11] D. Geer. Taking Steps to Secure Web Services. *IEEE Computer*, 36(10):14-16, 2003.

[12] C. Geuer-Pollmann and J. Claessens. Web Services and Web Service Security Standards. *Information Security Technical Report*, 10(1):15-24, 2005.

[13] J. Han and K. M. Khan. Security-Oriented Service Composition and Evolution. In *Proceedings of the Asia Pacific Software Engineering Conference*, pages 71-78. IEEE, 2006.

[14] T. Imamura, B. Dillaway, and E. Simon. XML Encryption. W3C, Dec. 2002. http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/, accessed on 11/2007.

[15] M. C. Jaeger, G. Rojec-Goldmann, and G. Muhl. QoS Aggregation for Web Service Composition using Workflow Patterns. In *Proc. of the IEEE Intl. Enterprise Distributed Object Computing Conference*, pages 149-159. IEEE, 2004.

[16] M. C. Jaeger, G. Rojec-Goldmann, and G. Muhl. QoS Aggregation in Web Service Compositions. In *Proceedings of the IEEE International Conference on E-Technology, E-Commerce and E-Service*, pages 181-185. IEEE, 2005.

[17] L. Kagal, M. Paolucci, N. Srinivasan, G. Denker, T. Finin, and K. Sycara. Authorization and Privacy for Semantic Web Services. *IEEE Intelligent Systems*, 19(4):50-56, 2004.

[18] G. Karjoth, B. Pfitzmann, M. Schunter, and M. Waidner. Service-Oriented Assurance - Comprehensive Security by Explicit Assurances. In *Proceedings of the Workshop on Quality of Protection*. 2005.

[19] E. M. Maximilien and M. P. Singh. A Framework and Ontology for Dynamic Web Services Selection. *IEEE Internet Computing*, 8(5):84-93, 2004.

[20] F. Montagut and R. Molva. Augmenting Web Services Composition with Transactional Requirements. In *Proc. of the IEEE Intl. Conf. on Web Services*, pages 91-98. IEEE, 2006.

[21] A. Nadalin, C. Kaler, R. Monzillo, and P. Hallam-Baker. Web Services Security: SOAP Message Security Version 1.1 Specification. OASIS, Feb. 2006. oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf, accessed on 11/2007.

[22] C. Patel, K. Supekar, and Y. Lee. A QoS Oriented Framework for Adaptive Management of Web Service Based Workflows. In *Proc. of the Intl. Conference on Database and Expert Systems Applications*, pages 826-835. Springer, 2003.

[23] P. F. Patel-Schneider, P. Hayes, and I. Horrocks. OWL Web Ontology Language Semantics and Abstract Syntax. W3C, Feb. 2004. w3.org/TR/owl-semantics/, accessed on 11/2007.

[24] M. A. Serhani, R. Dssouli, A. Hafid, and H. Sahraoui. A QoS Broker Based Architecture for Efficient Web Services Selection. In *Proceedings of the IEEE International Conference on Web Services*, pages 113-120. IEEE, 2005.

[25] N. Shadbolt, W. Hall, and T. Berners-Lee. The Semantic Web Revisited. *IEEE Intelligent Systems*, 21(3):96-101, 2006.

[26] B. Shields, O. Molloy, G. Lyons, and J. Duggan. Using Semantic Rules to Determine Access Control for Web Services. In *Proc. of the Intl. WWW Conf.*, pages 913-914. ACM, 2006.

[27] H. Song, Y. Sun, Y. Yin, and S. Zheng. Dynamic Weaving of Security Aspects in Service Composition. In *Proceedings of the IEEE International Symposium on Service-Oriented System Engineering*, pages 189-196. IEEE, 2006.

[28] D. Xua, Y. Qi, D. Hou, Y. Chen, and L. Liu. A Formal Model for Security-Aware Dynamic Web Services Composition. In *Proc. of the International Conference on Computational Science and its Applications*, pages 139-143. IEEE, 2007.

[29] T. Yu and K. Lin. A Broker-Based Framework for QoS-Aware Web Service Composition. In *Proceedings of the IEEE International Conference on E-Technology, E-Commerce and E-Service*, pages 22-29. IEEE, 2005.

[30] J. Zhang. Trustworthy Web Services: Actions for Now. *IT Professional*, 7(1):32-36, 2005.

[31] C. Zhou, L.-T. Chia, and B.-S. Lee. DAML-QoS Ontology for Web Services. In *Proc. of the IEEE Int'l Conference on Web Services*, page 472-479. IEEE, 2004.