MODELLING AND DEPLOYING SECURITY POLICIES

Xabier Larrucea
European Software Institute-Tecnalia, Parque Tecnologico de Zamudio 204, Zamudio, Spain

Rubén Alonso
Visual Tools, Gijon Science & Technology Park - C/ Los Prados 166, Gijon, Spain

Keywords: Method Engineering, SOA, Security Policy.

Abstract: Web Services (WS) platform is one of the most widely accepted implementations of Service Oriented Architectures (SOA). There is a huge amount of specifications related to the so-called WS-*. In this context Security Policies specification and deployment support is still immature and it needs improvement. This paper is focused on the growing importance of security, the increase of collaboration amongst organizations and the emergent need of modelling SOA and security aspects. This paper presents a methodology and modelling framework based on Eclipse platform for designing security aspects in SOA and a derivation mechanism in order to automatically generate web service security elements. This approach is illustrated with an example.

1 INTRODUCTION

Non functional aspects (NFA) have an evident impact in the resulting architecture and they are specialized for a specific platform. WS-Security is an open standard which defines means to add end-to-end security to the Web Services. Concretely, it includes mechanisms for enduing Web Services with integrity (through digital signature), confidentiality and authentication. These mechanisms are defined in the headers of SOAP messages as series of elements described in the standard. On the one hand, XML Encryption and XML Signature are the elements providing the services with confidentiality and integrity. On the other hand, the named Security Tokens allow authentication and authorization in Web Services platform. These security tokens allow the definition of the authentication and authorization methods of messages, which can comprise from the classic user/password authentication to X.509 certificates or only authorize users with Kerberos tickets

The wide range of available authentications mechanisms and, encryption and digital signature algorithms have favoured the development of a policy standard, specially oriented to security.

A standard known as and the WS-Policy extension for including security related asserts. Therefore, WS-Security Policy. (WS-security policy 1.2, 2007) is the base to be used for including in our servers constraints and requirements to be fulfilled by the clients to access to the services and vice versa, since a client could demand, for example, an encrypted response from the server.

This paper is an extension of previous work such as (Larrucea et al., 2008) where some mechanisms are developed for modelling security specifications. This paper is structured as follows. The first section describes mechanisms for modelling security policies. The second section describes some recommended tasks used for applying the mechanisms. The third section represents a case study where we have evaluated our approach. And finally a conclusion section is provided in order to summarise the approach.

2 INDEPENDENT SOA SECURITY SPECIFICATION

In previous work (Larrucea et al., 2007) a Model Driven Architecture (MDA®) is combined with Service Oriented Architecture (SOA) in order to solve interoperability issues focused on

Larrucea X. and Alonso R.
MODELLING AND DEPLOYING SECURITY POLICIES.
DOI: 10.5220/0001825504110414
In *Proceedings of the Fifth International Conference on Web Information Systems and Technologies (WEBIST 2009)*, page
ISBN: 978-989-8111-81-4

functionalities. In (Benguria et al., 2006) a metamodel is defined in order to design SOA models. This metamodel has been recently proposed as an OMG standard as response to the UML Profile and Metamodel for Services (UPMS) Request For Proposal (RFP) (UML Profile and Metamodel for Services, 2006). This metamodel called PIM4SOA (Platform Independent Model for Service Oriented Architectures) represents the Platform Independent Model (PIM) level in the Model Driven Architecture promoted by the Object Management Group (OMG). PIM4SOA metamodel is structured in 4 metamodels interrelated which are described deeply in (Benguria et al., 2006).

UML™ Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms specification is used for describing and specifying security policies. In fact, the QoS metamodel described in this specification represents the metamodel that we are going to use as basis. QoS metamodel can be used to model “*non functional aspects like: latency, throughput, capacity, scalability, availability, reliability, safety, confidentiality, integrity,[...].”*.

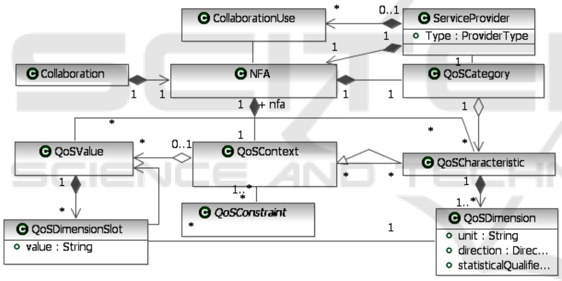


Figure 1: QoS enhanced metamodel.

3 METHOD ENGINEERING SUPPORT

Based on (Larrucea, 2008) we have adopted a method engineering approach for designing a methodology supporting our approach. We have defined a set of method fragments representing building blocks for methodology definition. In our work we have been focused on task definition but they are other method fragment to consider such as roles and guidelines.

Previous sections are focused on the mechanisms such as metamodels for modelling security policies. The main purpose of this section is to provide some guidelines for modeling and deploying security policies.

In fact we have identified some tasks and work products as a basis for these guidelines. These method fragments are categorized by the metamodel aspects described on section 2:

Services

- Identify services in your software architecture: we need to identify which part of the architecture is accessed as a service
- Identify services publicly available or consumed by third parties. This kind of services are called critical services
- Relate services among them as consumer-provider relationship

Processes

- Identify flows between services in your software architecture: we need to identify which are the enacted processes that will be executed
- Identify public flows: there are some flows that are followed by consumer and users.
- Identify private flows: flows controlled and not accessed by third parties or external users

Information

- Identify Information exchanged in your software architecture
- Identify public Information: we define work products according to this public information
- Identify private Information: we define work products according to this private information

QoS

- Define QoSCategories: this element classifies kind of characteristics
- Define QoSCharacteristics: this element represents a security element and it is associated to a service.
- Relate critical services with QoSCharacteristics
- Assign values to QoSCharacteristics
- Define Integrity for each critical service
- Define Confidentiality for each critical service
- Define Availability for each critical service

All tasks should be analysed from the following point of view because security mechanism is based on messages exchanged between services:

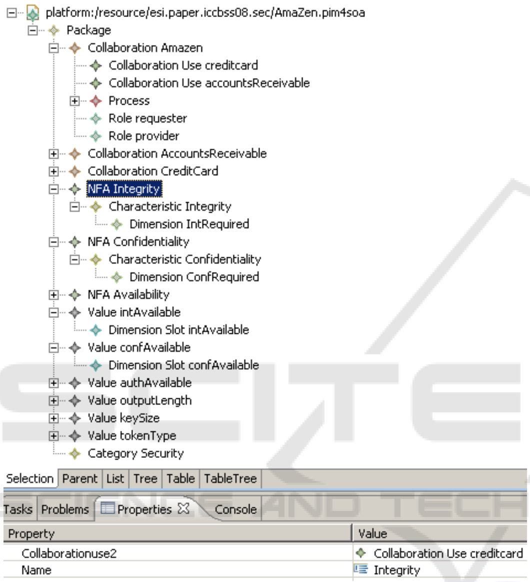
- At *Service level*
- At *Endpoint level*
- At *Operation level*
- At *Message level*

4 CASE STUDY: ONLINE BOOKSHOP

We have applied this approach using as a basis the following scenario. This scenario describes a service architecture defined for an online bookshop.

In this scenario integrity and confidentiality aspects for the CreditCard Web Service are proved. In this context petitions are signed and the content of messages are encrypted.

These security elements help to validate that credit card number is not exposed and the message has not been modified by third parties.



Property	Value
Collaborationuse2	Collaboration Use creditcard
Name	Integrity

Figure 2: Cases study snapshot.

Figure 2 represents a typical snapshot of an Eclipse based plug-in. In this scenario there are three services specifications (collaboration nodes): the main service called Amazen and the other two services: AccountsReceivable and CreditCard. The purpose is to show the definition of these services, the definition of security policy elements and the relationship between them.

NFA Integrity represents the integrity requirement and it is used to prevent any modification of interchanged messages. This element is selected and its properties as described below are *Collaborationuse2* and *Name*. These properties specify the name and the relationship with the usage of a service. This relationship is used to attach the generated WSDL and WS-Security policies.

This example shows a WSDL definition (*CreditCard*) with an embedded policy (*SecureCreditCard*) obtained from the model defined in the scenario description and it is applied to the messages for the operation called *ValidateCreditCard.*

5 CONCLUSIONS

This paper presents an approach providing a holistic view of a service oriented architecture environment. This approach allows the definition of functional and non functional aspects in a coherent way based on metamodel and a method engineering support. In fact we have provided a way to define and deploy security policies specifications in SOA. Using this approach we can generate web service artefacts such as WSDL, XSD, BPEL and WS-SecurityPolicy as a whole.

This approach is implemented using the Eclipse platform and facilities, and therefore it is a modular and extendable approach. However and as it is shown in the example we need to develop a graphical environment and this is one of the current improvements we are focused on. Moreover the modelling side (PIM4SOA) could be extended with parts of the WS-SecurityPolicy1.2 specification that are out of scope of the approach presented in this paper. Another future work is to set up the relationships at higher levels of abstraction such as Business Process Definition Metamodel.

In addition the approach presented in this paper is aligned with OMG standardisation initiatives such as (UML Profile and Metamodel for Services, 2006) where a service metamodel and UML profile is being developed.

ACKNOWLEDGEMENTS

This work is being supported in part by MetaMethod project (TIN2006-15175-C05-04) funded by Ministerio de Educacion y Ciencia, Spain government and by the Vulcano project (Proyecto Singular Ministerio Industria, Turismo y Comercio-TSI-020301-200822).

REFERENCES

Benguria, G. Larrucea, X., Elvesaeter, B.Neple, T., Beardsmore, A, Friess M. “A Platform Independent Model for Service Oriented Architectures”. I-ESA conference 2006, March 22-24, Bordeaux. Springer

Larrucea, X. 2008. Method Engineering Approach for Interoperable Systems Development. Journal on Software Process: Improvement and Practice. Vol 13 issue 2. Pages 127-133. ISSN: 1077-4866.

Larrucea, X. Alonso, R. 2008 ISOAS: Through an independent SOA security specification. International Seventh IEEE Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems Pages 92-100. 2008 ISBN:978-0-7695-3091

Larrucea, X. Benguria G, Schuster S. 2007. "MDSOA for Achieving Interoperability," iccbss, p. 247. Sixth International IEEE Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems (ICCBSS'07)

UML Profile and Metamodel for Services, 2006 Request For Proposal (RFP-OMG). http://www.omg.org/docs/soa/06-09-09.pdf

UML™ Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms. 2004, Object Management Group. http://www.omg.org/docs/ptc/04-09-01.pdf .

WS-security policy 1.2 (OASIS) 2007. http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/ws-securitypolicy-1.2-spec-cs.pdf

World Wide Web consortium. Web Services Policy 1.5, 2007 Attachment http://www.w3.org/TR/ws-policy-attach/