Multi Domain-Specific Modeling of the Security Concerns of Service-Oriented Architectures

Juan Pedro Silva Gallino, Miguel de Miguel, Javier F. Briones, Alejandro Alonso
Departamento de Ingeniería Telemática
Universidad Politécnica de Madrid (UPM)
Madrid, Spain.
Email: {psilva, mmiguel, jfbriones, aalonso}@dit.upm.es

Abstract—**Service-Oriented Architectures (SOA), and Web Services (WS), the technology generally used to implement them, achieve the integration of heterogeneous technologies, providing interoperability, and yielding the reutilization of pre-existent systems. Model-driven development methodologies provide inherent benefits such as increased productivity, greater reuse, and better maintainability, to name a few. Efforts on achieving model-driven development of SOAs already exist, but there is currently no standard solution that addresses non-functional aspects of these services as well. This paper presents an approach to integrate these non-functional aspects in the development of web services, with an emphasis on security.**

Keywords-**MDA; WS; SOA; AC; WS-SecurityPolicy;**

I. Introduction

Service-Oriented Architectures (SOA), of great popularity nowadays, and Web Services (WS), the technology generally used to implement them, achieve the integration of heterogeneous technologies, providing interoperability, and yielding the reutilization of pre-existent systems.

At the same time, Model Driven Applications (MDA) [1] arises as a new paradigm that provides, among other benefits, a greater understanding of the system as a whole and a platform-independent development, improving reusability of design, and simplifying the evolution of the system, thus increasing productivity.

WS-* standards provide a strong foundation for the development of services. WS-Policy and related specifications offer the possibility of considering extra-functional concerns of those services. However, the syntax of these specifications is not designed to be read/written by humans, and an abstraction of such languages is desirable.

Although multiple implementation technologies exist to facilitate the development of WS and SOA systems, it was identified that there exists a lack of a design solution that:

- Addresses functional and non-functional (NF) concerns of SOAs independently, fostering reuse.
- Describes each concern in a convenient manner.
- Is modular enough so that the functional model is not polluted with non-functional information.
- Provides a practical way of analyzing the system for non-functional characteristics in early stages.

Some approaches applying MDA to the development of Web Services already exist, but most of these approaches do not offer thorough support for NF configuration, code generation, Web Services Description Language (WSDL), and WS-Policy. This paper presents ongoing work towards achieving a solution that includes all these previously mentioned concepts.

II. Integrating the Non-Functional Concerns

Leveraging the work presented in [2], a complete solution addressing NF aspects of service architectures was defined. This section presents the process proposed to integrate these non-functional aspects in the development of web services.

A. Development Process

Figure 1 shows the overall structure of the designed solution, consisting of a chain of model transformations and compositions. Every box in the figure indicates a different type of model, including:

- Business model indicating non-functional intentions.
- Functional model and non-functional aspects models.
- Intermediate Meta Model (iMM) model (PIM 2 Level).
- WSDL, WS-Policy, and XML models (PSM Level).

The objective of the iMM metamodel is achieving an abstraction both of input modeling techniques/metamodels, and output target platforms. It is also designed for holding in one unique model, all information necessary to generate the different output artifacts, and to perform analyses on.

The presented approach proposes the development of each individual concern as an independent model. Moreover, it suggests the use of different DSLs, each one appropriate to the particular concern in hand. Such models are later composed into a complete model of the system by weaving them together. An implementation using a possible set of metamodels is presented in [2].

As illustrated in figure 1, repositories containing different expert-made NF models could provide the NF input. Appropriate models are automatically selected from the repositories and offered to the modeler. As a result, the developer need not be savvy in the particular NF concern (e.g, security).

B. Sequence of Operation

The sequence of steps in the proposed approach, which extends the one in [2], follows. Tool support, with extensive use of default value options to allow an end-to-end automatic process, is desirable.

1) A business process model, annotated with abstract non-functional intents, is presented as input to the process. The intents at CIM level drive the selection of the appropriate NF models.
2) A functional PIM model is derived from the CIM model. Alternatively, the user can provide an input functional model at PIM level (PIM 1) [2].
3) By means of composition models, the user indicates which non-functional properties should be associated to which resources in the functional model. Ideally, pre-filled composition models should be proposed.
4) At this stage, the platform-independent iMM model (PIM 2) contains all (functional and NF) data.
5) Next, the PIM2 model is transformed into platform dependant (PSM) models. Optional platform information may be fed to the transformation.
6) Finally, service and policy descriptors are derived from the PIM 2 model. Resulting WSDL documents hold references to the appropriate policies defined in the different generated WS-Policy documents. Simultaneously, code templates and non-functional configurations are automatically created.

The idea behind the approach is that each model in the repositories address one or more non-functional concerns. Experts in the particular NF domain create such models, and also indicate how the different elements map onto the target platform. Appropriate metadata is associated to these models for automatic discovery support.

The benefits are twofold: it frees the developers from the need of non-functional expertise, and allows greater reuse of (functional **and** non-functional) designs. NF models may be reused among different projects. Also, different NF characteristics may be applied to the same functional design to generate different resulting systems.

III. Conclusions And Future Work

This paper has presented a piece of work focused on the achievement of an integrated, model-driven solution for the development of service-oriented architectures with policy awareness and access control support.

Multi domain-specific models, independently addressing non-functional concerns, and being independently mapped into target platforms, foster concerns' models reuse and favor that each concern be addressed in a convenient manner. Additionally, the modularity of the approach permits that each concern design, functional or not, is not polluted with extraneous information. Finally, the integrated model offers a practical way of measuring and analyzing the system in early stages of the development.

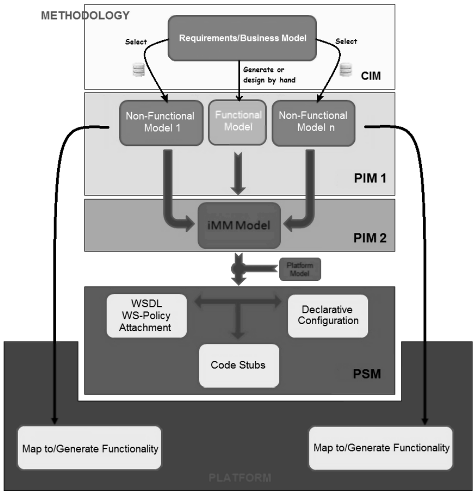
METHODOLOGY
Requirements/Business Model
Select
Select
Generate or design by hand
CIM
Non-Functional Model 1
Functional Model
Non-Functional Model n
PIM 1
iMM Model
PIM 2
Platform Model
WSDL WS-Policy Attachment
Declarative Configuration
Code Stubs
PSM
Map to/Generate Functionality
Map to/Generate Functionality
PLATFORM

Figure 1. Structure of the proposed solution.

The proposed solution presents great flexibility in its evolution regarding the appearance of new standards or technologies. Moreover, the WS-Policy framework itself has been defined in a generic fashion, allowing for many standards to be formulated based on it. Consequently, all standards under its umbrella will automatically be supported by this solution. This leverages its use for developing not only security-aware services, but also include reliability, timing constraints, secure exchange, transactions, etc.

With respect to future work, in the short term it will be focused on completing a working prototype. In a longer term, it is planned to consider other policy aspects under standardization process. As previously mentioned, the approach is flexible enough to support any WS-Policy-based standard, but it is not able to use the information intrinsic to the assertion for its benefit (shaping of code generation, configuration of target platform, e.g.).

On a different token, the shifting from a monolithic-metamodel-based generation approach towards a composition-based generation approach (in which any metamodel could be used to generate artifacts based on a set of weaving associations) could provide an alternative line of research.

References

[1] Miller, J.: MDA Guide Version 1.0. 1. Object Management Group (2003)

[2] Silva Gallino, J.P., de Miguel, M.A., Briones, J.F., Alonso, A.: Model-Driven Development of a Web Service-Oriented Architecture and Security Policies. In: 2010 13th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing. pp. 92–96. Carmona, Spain (2010)

[3] Peri Tarr, Harold Ossher, William Harrison, and Stanley M. Sutton Jr. N degrees of separation: multi-dimensional separation of concerns. *International Conference on Software Engineering*, pages 107 – 119, 1999.