

## Chapter XVIII

# Modeling of Web Services using Reaction Rules

**Marko Ribarić**

*Mihailo Pupin Institute, Serbia*

**Shahin Sheidaei**

*Simon Fraser University, Canada*

**Milan Milanović**

*University of Belgrade, Serbia*

**Dragan Gašević**

*Athabasca University, Canada*

**Adrian Giurca**

*Brandenburgische Technische Universität  
Cottbus, Deutschland*

**Sergey Lukichev**

*Brandenburg University of Technology at  
Cottbus, Germany*

**Gerd Wagner**

*Brandenburg University of Technology,  
Germany*

### ABSTRACT

*The development process of Web services needs to focus on the modeling of business processes rather than on low-level implementation details of Web services, and yet it also needs to incorporate the support for frequent business changes. This chapter presents the UML-based Rule Language (URML) and REVERSE Rule Markup Language (R2ML), which use reaction rules (also known as Event-Condition-Action rules) for modeling Web services in terms of message exchange patterns. Web services that are being modeled in this way can easily be integrated in the wider context of modeling orchestration and choreography. In order to achieve proposed solution, we have developed a plug-in for the Fujaba UML tool (so called Strelka) and a number of model transformations for round-trip engineering between Web services and reaction rules. Also, the paper presents mappings of models of Web services with reaction rules into the Drools rule language, thus enabling the run time execution semantics for our rule-based models.*

## 1. INTRODUCTION

Web services provide a communication interface and a workflow management protocol for business systems. Their goal is set to change the Web from a static collection of information, to a dynamic place where different software components (business logics) can be easily integrated (Arroyo et al., 2004). Some of the factors that constrain web services from achieving that goal can be named as follow: when developers try to enrich the existing Web services with some new functionality, they have to implement necessary changes manually, which may lead to potential execution errors; furthermore, business rules, due to their declarative nature, are used for reflecting dynamic changes representing desired alteration in the business process. When business rules are used in a networked business, they are often large and complex, which makes them difficult to manage and change. There is a lack of automatic mechanism for updating Web services based on the business rules changes (Ribaric et al., 2008).

The need for a more straightforward way to express, and manage business rules as a separate part of Web services' composition has been recognized by researchers for the numerous benefits that it provides (Charfi & Mezini, 2004). However, integrating rule-based systems in a service-oriented environment is a complex task, due to the fact that both worlds have their own paradigms (Rosenberg & Dustdar, 2005). Many different factors involve in this integration. However, the most important one of these factors is the architecture of service oriented systems. In this architecture, layers play a great role. The layering architecture is needed to be represented and deployed in the world of rules which become challenging and could not be simple done. This integration approach should be loosely coupled, meaning that it is reasonable to expose business rules as services. The way of modeling Web services presented in this paper facilitates this integration process, and those services can be more easily integrated in the wider context of modeling orchestration and choreography.

A promising way to solve these problems is to use a high-level modeling approach combined with the use of rules. A modeling approach will allow developers to focus on a problem domain rather than on an implementation technology. This is why we propose using an approach based on Model Driving Engineering (MDE). Although, there have been several attempts to leverage MDE principles to model Web services, they are still very low level oriented, as they again focus on technical details covered either by WSDL (Bezivin et al., 2004) (Vara et al., 2005) or OWL-S (Timm & Gannod, 2005) (Gronmo et al., 2005). The use of rules implying that they are defined in a declarative way, they can dynamically reflect business logic changes at run-time without the need to redesign the whole system. Since Web services are used for integration of business processes of various stakeholders, it is important for them to reflect changes in business logic, or policies, as good as possible.

The solution that we propose is to use rules to represent and model business processes from the perspective of message exchange. That is, our modeling approach enables one to model Web services from the perspective of the underlying business logic regulating *how* Web services are used regardless of the context *where* they are used. To do so, our proposal is to leverage message-exchange patterns (MEPs) as an underlying perspective integrated into a Web service modeling language. This perspective has already been recognized by Web service standards (Chinnici et al., 2007), and our contribution is to raise this to the level of models.

As the leading Rule Modeling language we benefit from UML-based Rule Modeling Language (URML) (Lukichev & Wagner, 2006a) for our approach. URML having strong ties with R2ML (REVERSE Rule Markup Language) empowers us to model rules in various languages. URML supports modeling of domain vocabularies in addition to integrity, derivation, production and reaction rules. Business rules, being similar

23 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:  
[www.igi-global.com/chapter/modeling-web-services-using-reaction/35869](http://www.igi-global.com/chapter/modeling-web-services-using-reaction/35869)

## Related Content

---

### Rules Capturing Events and Reactivity

Adrian Paschke and Harold Boley (2009). *Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches* (pp. 215-252).

[www.irma-international.org/chapter/rules-capturing-events-reactivity/35861](http://www.irma-international.org/chapter/rules-capturing-events-reactivity/35861)

### Enhancing RUP Business Model with Client-Oriented Requirements Models

Maria C. Leonardi (2003). *UML and the Unified Process* (pp. 80-115).

[www.irma-international.org/chapter/enhancing-rup-business-model-client/30539](http://www.irma-international.org/chapter/enhancing-rup-business-model-client/30539)

### Keyword Search on XML Data

Ziyang Liu and Yi Chen (2010). *Advanced Applications and Structures in XML Processing: Label Streams, Semantics Utilization and Data Query Technologies* (pp. 143-159).

[www.irma-international.org/chapter/keyword-search-xml-data/41503](http://www.irma-international.org/chapter/keyword-search-xml-data/41503)

### Mapping Policies to Web Rules: A Case of the KAoS Policy Language

Nima Kaviani, Dragan Gasevic and Marek Hatala (2009). *Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches* (pp. 564-595).

[www.irma-international.org/chapter/mapping-policies-web-rules/35875](http://www.irma-international.org/chapter/mapping-policies-web-rules/35875)

### Native XML Programming: Make Your Tags Active

Philippe Poulard (2009). *Open and Novel Issues in XML Database Applications: Future Directions and Advanced Technologies* (pp. 151-180).

[www.irma-international.org/chapter/native-xml-programming/27781](http://www.irma-international.org/chapter/native-xml-programming/27781)