SWWS Studio - a WSMO compliant editor

Marin Dimitrov, Zlatina Marinova, Peter Radkov

OntoText Lab. / SIRMA
135 Tsarigradsko Shose Blvd., Sofia 1784, Bulgaria
{marin.dimitrov, zlatina.marinova, peter.radkov}@ontotext.com

Abstract. The Web Services Modelling Framework (WSMF) and the Web Services Modelling Ontology (WSMO) provide a unique, highly innovative perspective onto Semantic Web and Web Service technologies. We present in this paper the *SWWS Studio* – a prototype that supports and elaborates that perspective, making the technology easy to use and transparent for the end user. At present the *SWWS Studio* is comprised of two components: a graphical designer for modelling WSML specifications and a graphical tool for modelling compositions of services.

1 Introduction

Robust, mature and easy-to-use tools play crucial role for the easy adoption of any new technology. In fact the overall value of a technological innovation can be severely undermined by the lack of proper tools that support it.

The Web Services Modelling Framework (WSMF) [5] and the Web Services Modelling Ontology (WSMO) [9] provide a unique, highly innovative perspective onto Semantic Web and Web Service technologies. We present a prototype that supports and elaborates that perspective, making the technology easy to use and transparent for the end user. In particular, we present the *SWWS Studio*[1] developed within the scope of the EU-funded SWWS project [2].

2 Features

SWWS Studio is comprised of two components: a graphical designer for modelling WSML [7] specifications (Service Editor) and a graphical tool for modelling compositions of services (Service Composer).

2.1 Service Editor

The Service Editor (Fig. 1) provides a graphical interface for creating WSMO descriptions. Definitions of all WSMO elements - ontologies, goals, mediators and services - can be easily created using the editor. Element descriptions can be exported in the following formats:

[1] http://swws.ontotext.com

– WSML [7]
– WSMO F-Logic format [6]
– OWL-DL format for WSMO defined for the purpose of the SWWS project

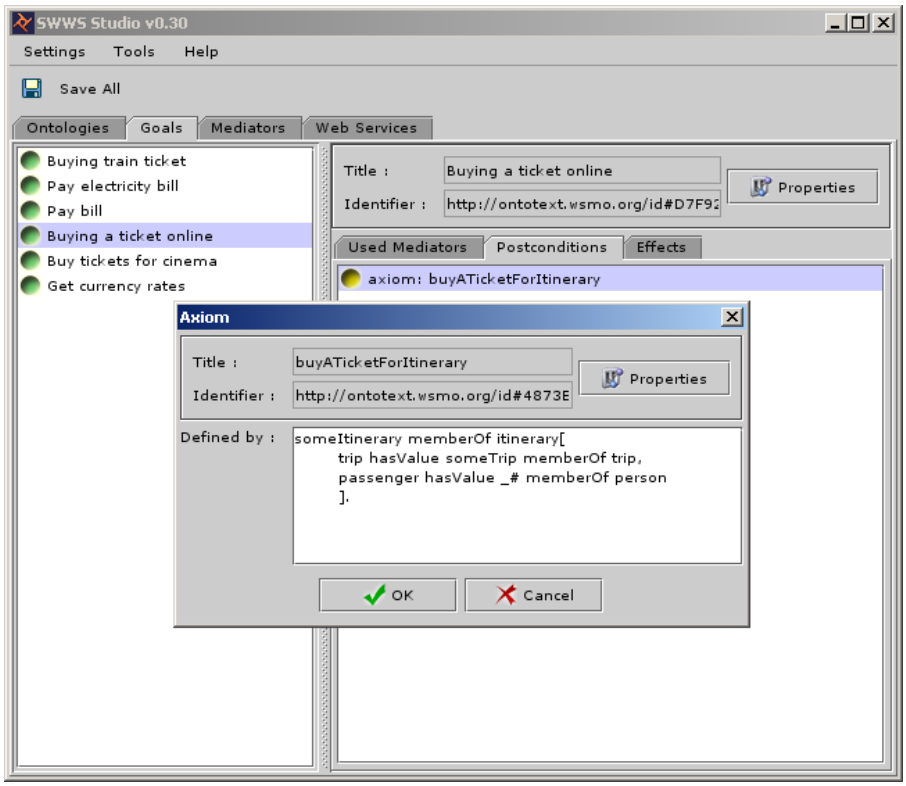

Fig. 1. SWWS Studio - creating a goal description

The Service Editor can be used as a front-end browser to a remote WSML registry.

2.2 Service Composer

The Service Composer (Fig. 2) is used for defining service orchestrations, e.g. the interaction with other services in order to achieve the service functionality. At present the orchestration definitions created with *SWWS Studio* can be exported as an OWL-S process model [4] but future versions will support other language representations when defined within WSMO.

The control flow constructs[2] supported by the Service Composer include:

[2] Refer to [10] for an extensive classification of workflow patterns

– *Sequence* – sequential execution of activities
– *Parallel split + Synchronisation* – parallel execution of activities which converge into a single activity.
– *Exclusive choice* – a point where only one out of many alternative activities is chosen based on a specified condition.
– *Iteration* – structured cycles allowing a group of activities to be executed several times.

In addition to the above constructs the control flow of a service orchestration is comprised of one or more *proxies* that are the elementary units of orchestration.

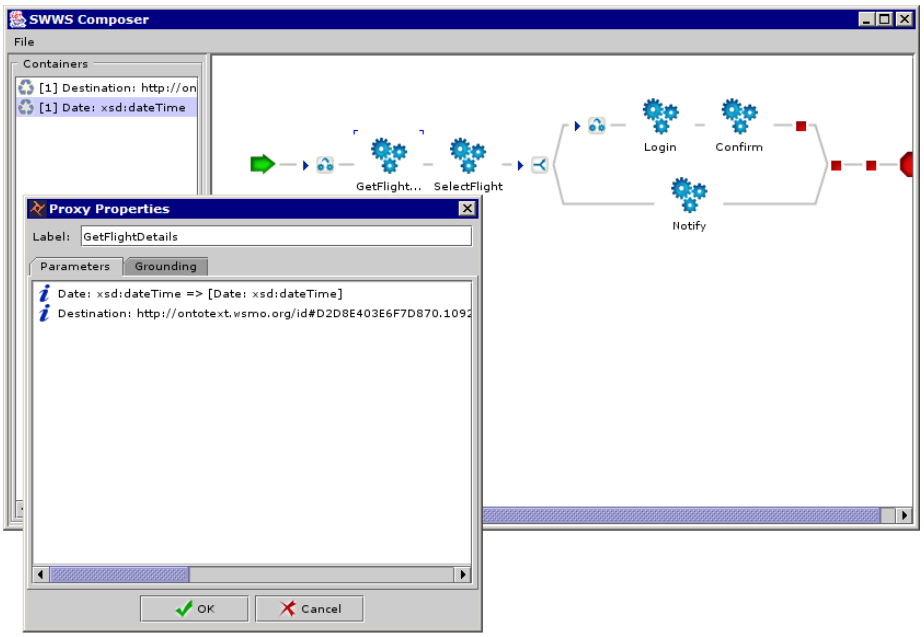

Fig. 2. SWWS Studio - creating a service orchestration

In *SWWS Studio* proxies are associated with input and output parameters, typed according to either the built-in XML Schema datatypes [3] or concepts in a specified ontology. The mappings between outputs and inputs are specified with the help of *containers* - state variables that can be used for storing data values from output ports and retrieving the values into the input ports.

3 Future Work

Our future work will be focused in these directions:

- *WSMO Studio* - an Eclipse [1] compliant implementation based on *SWWS Studio*. Such an implementation will ease the integration of 3^{rd} party components in the *Studio* and improve its reusability and extensibility
- support for the latest WSMO Standard drafts
- new functionality such as a choreography editor compliant with [8]

References

1. The Eclipse project. http://www.eclipse.org.
2. SWWS - Semantic Web Enabled Web Services, project IST-2002-37134. http://swws.semanticweb.org.
3. P. V. Biron and A. Malhotra. XML schema part 2: Datatypes. W3C recommendation, W3C, 2001. Available at http://www.w3.org/TR/xmlschema-2/.
4. The OWL Services Coalition. OWL-S: Semantic markup for Web Services, 2003. Available at http://www.daml.org/services/owl-s/1.0/owl-s.html.
5. D. Fensel and C. Bussler. The web service modeling framework (WSMF). *Electronic Commerce Research and Applications*, 1(2), 2002.
6. H. Lausen and D. Roman. WSMO F-Logic Syntax, v0.1, March 2004. Available at http://www.wsmo.org/2004/d16/d16.4/v0.1/20040324/.
7. E. Oren, M. Kifer, J. de Bruijn, H. Lausen, D. Roman, and M. Felderer. BNF Grammar for WSML User Language, v0.2, April 2004. Available at http://www.wsmo.org/2004/d16/d16.1/v0.2/20040418/.
8. D. Roman, M. Dimitrov, and M. Stolberg. Choreography in WSMO, v0.1, July 2004. Available at http://www.wsmo.org/2004/d14/v0.1/20040717/.
9. D. Roman, U. Keller, and H. Lausen. Web Service Modeling Ontology - Standard, v0.2, March 2004. Available at http://www.wsmo.org/2004/d2/v0.2/20040306/.
10. W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros. Workflow patterns. *Distributed and Parallel Databases*, 14(3):5–51, 2003.