

# The Web Service Modeling Toolkit

Mick Kerrigan and Adrian Mocan

Semantic Technology Institute (STI) Innsbruck,  
Universität Innsbruck, Austria  
`firstname.lastname@sti2.at`

**Abstract.** The development of software is not an easy task and the availability of adequate tool support is an important step towards reducing the effort that a developer must put into the Software Development Cycle. As an emerging technology, it is vital that Semantic Web Services can be quickly and easily created by developers to ensure that this new technology can be easily adopted. In this demo the process of developing Semantic Web Service descriptions, through the WSMO paradigm, using the Web Service Modeling Toolkit (WSMT) will be presented.

## 1 Introduction

Web services are quickly becoming one of the most important technologies for business to business integration; however some of the promises claimed by Service Oriented Architectures in terms of loose decoupling and easy reusability have failed to be met, as the ability to dynamically find and link Web services at runtime cannot be realistically performed with syntactic technologies. Semantic Web Services are the extension of ontologies to describe Web services in such a way that they can be dynamically *discovered*, *composed*, *ranked*, *selected*, *mediated* and *invoked* at runtime. Such functionality reduces the amount of effort that a developer must spend building an application using a Service Oriented Architecture and improves the overall quality of that application as new services, which may be cheaper or have better performance, become immediately available to the application once they are published by the provider.

The life of the Semantic Web Service developer is not such an easy one though. As noted in 1994 by the Standish group chaos report[10], 31% of all software development projects in their survey failed, i.e. were canceled at some point during the development cycle, another 53% were completed but over-budget, over the estimate time or provided fewer functions and features than originally intended, thus leaving only 16% of all software projects surveyed as completing successfully. Given that the software projects were using well established technologies like Java, C# or Visual Basic with good tool support provided by development environments equivalent to the Eclipse Java Development Toolkit<sup>1</sup>, SharpDevelop<sup>2</sup> and Visual Studio<sup>3</sup>, it would be surprising if the Semantic Web Service developer could do much better without any tool support at all.

<sup>1</sup> <http://www.eclipse.org/jdt/>

<sup>2</sup> <http://www.icsharpcode.net/>

<sup>3</sup> <http://msdn.microsoft.com/vstudio/>

The Web Service Modeling Toolkit (WSMT)[6,7] is an integrated development environment for Semantic Web Services that enables developers to develop Ontologies, Web Services, Goals and Mediators through the Web Service Modeling Ontology (WSMO)[3] formalism. The WSMT is implemented as a collection of plug-ins for the Eclipse<sup>4</sup> framework such that it can be integrated with other toolkits like the Java Development Toolkit JDT or the Web Tools Platform (WTP)<sup>5</sup> so that a developer can develop his java code, Web services and Semantic Web Services side by side in the one application. The main aim of the WSMT is to support the developer through the full Software Development Cycle of his Semantic Web Service from requirements, through design, implementation, testing, and deployment such that the process of developing Semantic Web Services can become cheaper to perform and remove many of the tedious activities that the developer must currently perform.

## 2 The Web Service Modeling Toolkit

The WSMT has been under development since early 2005 and is made up of three main areas of functionality:

- **Creation and Management of WSML Artifacts:** The ability to quickly and cheaply create and test WSMO Ontologies, Web Services, Goals and Mediators through the Web Service Modeling Language (WSML)[8] is key to the successful creation of Semantic Web Services. The WSMT provides the **WSML perspective** with multiple editors[5] for creating and testing WSMO descriptions, conversion tools to and from RDF and OWL, embedded reasoners for testing the behavior of ontologies in their target environment, and embedded discovery engines for ensuring that Goals and Web Services match each other as expected.
- **Creation and Management of Mediation Mappings:** One of the key challenges in semantics is the interoperability of ontologies. In the Semantic Web Service field this becomes even more important when the service requester and service provider use different ontologies to describe the same domain. The WSMT provides the **Mapping Perspective**[1] within which mediation mappings between two or more ontologies can be created at design time, such that they can later be executed at runtime. The tools in this perspective guide the developer through the process of creating mappings using visual cues, suggestion algorithms and embedded testing functionality.
- **Interfacing with Semantic Execution Environments:** Crucially once all the artifacts related to a Semantic Web Service have been created these artifacts need to be deployed the execution environment within which they will be used. The **SEE Perspective** provides functionality for interfacing with Semantic Execution Environments like the Web Service Execution Environment (WSMX)[4] and IRSIII[2]. Artifacts can be stored to and retrieved from

---

<sup>4</sup> <http://www.eclipse.org>

<sup>5</sup> <http://www.eclipse.org/webtools/>

these environments, or can be used to invoke the functionality of the Semantic Execution Environments, for example discovering services that match, or fully invoking the best service that matches, a provided Goal.

The WSMT development team is also in the process of branching out into new areas including the encoding of business processes semantically in WSML through the many projects we are involved in. These new tools will form a number of new key WSMT perspectives that will support the developer in creating Semantically Enabled Service Oriented Architectures.

### 3 Contents of the Demo

The demo of the Web Service Modeling Toolkit takes the user through the different tools that are available for developing those artifacts related to Semantic Web Services and how these tools interact with each other. The demo begins by introducing the user to the different editors available for creating WSML Ontologies, Web Services, Mediators and Goals based on the expertise of the user.

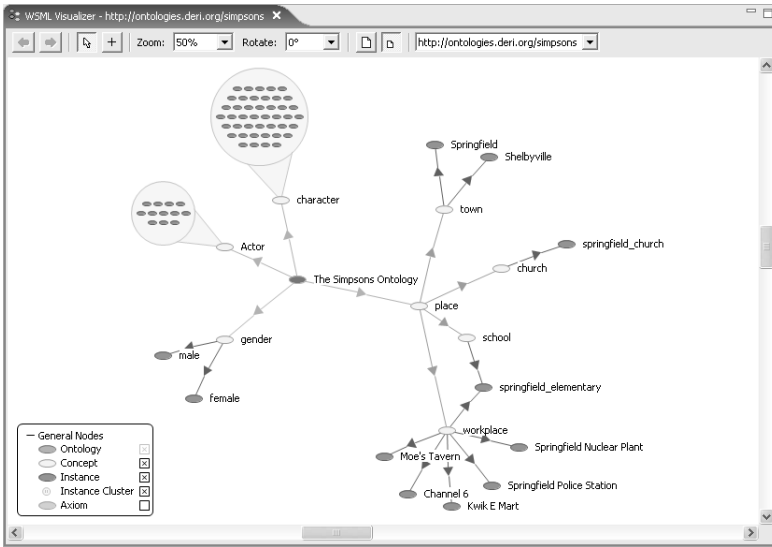


Fig. 1. An Ontology in the WSML Visualizer

The WSML Visualizer[5], as can be seen in figure 1, can be used by novice users who know very little about ontologies to get an overview of WSMO descriptions or by expert users to dig deeper into the intricacies of the relationship between elements in the semantic description. Crucially the WSML Visualizer also has fully embedded editing support allowing the user to directly manipulate

the semantic description vis the graph representation. The WSML Form Based Editor[7] provides a more functional view of the semantic description providing custom forms to the user which can be filled in to create and maintain semantic descriptions. Finally the WSML Text Editor[7] is available to the user who is more experienced with the Human-readable syntax of WSML, it provides functionality like syntax highlighting, syntax and content assistance, code folding, and bracket highlighting. At this stage the user is also introduced to the validation support available in the WSMT, which is used to give immediate feedback to the developer whenever they make mistakes.

Now that the user is familiar with the different types of editing support available for WSML descriptions, and has an understanding of how to create and maintain them, they are introduced to the WSML Reasoner View[7] within which queries can be executed over the currently visible WSMO Ontology. The reasoner view gives the developer the opportunity to execute WSML queries over WSML-Flight and WSML-Rule ontologies using the IRIS, MINS or KAON2 reasoners and over WSML-DL ontologies. The user is introduced to the Query template preferences, that enables the creation of complex logic programming queries by a domain expert and are later displayed to the end user as natural language queries, where drop downs and text fields can be used for configuring them. These query templates enable those that are not familiar with logics to access the power of reasoning over ontologies.

Having introduced ontologies as the basis upon which Semantic Web Services are built, the user is brought back to the concept of Goals and Web Services with the WSML Discovery View[6]. The WSML Discovery View provides access to a number of underlying discovery engines with which the developer can ensure that a specified Goal matches the expected set of Semantic Web Services. This is especially important as a Semantic Web Service that does not match the expected Goal is essentially unfindable by service requesters and thus will never be used in applications. The discovery view can also give a competitive advantage to new service providers on the market, allowing them to test that their new Semantic Web Services match the sample Goals provided by existing providers on the market. Thus they can attempt to take business from its competitors by providing services with better quality of service or lower cost.

As already mentioned the importance of the ability to mediate between different ontologies is necessary in the heterogeneous world of the web. Thus the user is introduced to the WSMT Mapping perspective and particularly the View Based Editor[9] for the semi-automatic creation of ontology to ontology mappings for runtime instance transformation[1] with the WSMX[4] environment. At runtime these mappings are used within WSMX to transform instances provided by the service requester into the terms expected by the service provider and vice versa. Alongside the view based editor the user is presented with the MUnit view, which can be used for creating unit tests for mediation mappings. With this view the developer can define tests consisting of a set of source instances and a set of target instances. The tests can be executed whenever the ontologies or mappings evolve in order to be sure that the mappings are still valid.

## Acknowledgements

The work is funded by the European Commission under the projects ASG, DIP, enIRaF, InfraWebs, Knowledge Web, Musing, Salero, SEKT, SEEMP, SemanticGov, Super, SHAPE, SWING and TripCom; by the FFG (Österreichische Forschungsförderungsgesellschaft mbH) under the projects Grisino, RW, SemNetMan, SeNSE, TSC, OnTourism.

## References

1. Mocan, A., Cimpian, E.: An Ontology-based Data Mediation Framework for Semantic Environments. *International Journal on Semantic Web and Information Systems (IJSWIS)* 3(2), 66–95 (2007)
2. Cabral, L., Domingue, J., Galizia, S., Gugliotta, A., Norton, B., Tanasescu, V., Pedrinaci, C.: IRS-III: A Broker for Semantic Web Services Based Applications. In: *Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*, Athens, Georgia, USA (2006)
3. Fensel, D., Lausen, H., Polleres, A., de Bruijn, J., Stollberg, M., Roman, D., Domingue, J.: *Enabling Semantic Web Services – The Web Service Modeling Ontology*. Springer, Heidelberg (2006)
4. Haller, A., Cimpian, E., Mocan, A., Oren, E., Bussler, C.: WSMX - A Semantic Service-Oriented Architecture. In: *Proceedings of the International Conference on Web Services (ICWS 2005)*, Orlando, Florida, USA (July 2005)
5. Kerrigan, M.: WSMOViz: An Ontology Visualization Approach for WSMO. In: *Proceedings of the 10th International Conference on Information Visualization (IV 2006)*, London, England (July 2006)
6. Kerrigan, M., Mocan, A., Tanler, M., Bliem, W.: Creating Semantic Web Services with the Web Service Modeling Toolkit (WSMT). In: *Proceedings of the workshop on Making Semantics Work For Business (MSWFB2007) at the 1st European Semantic Technology Conference (ESTC2007)*, Vienna, Austria (May 2007)
7. Kerrigan, M., Mocan, A., Tanler, M., Fensel, D.: The Web Service Modeling Toolkit - An Integrated Development Environment for Semantic Web Services (System Description). In: *Proceedings of the 4th European Semantic Web Conference (ESWC2007)*, Innsbruck, Austria (June 2007)
8. Lausen, H., de Bruijn, J., Polleres, A., Fensel, D.: WSMML - A Language Framework for Semantic Web Services. In: *Proceedings of the W3C Workshop on Rule Languages for Interoperability* (April 2005)
9. Mocan, A., Cimpian, E.: Mapping creation using a view based approach. In: *1st International Workshop on Mediation in Semantic Web* (2005)
10. Standish Group. *The CHAOS Report*. Technical report, Standish Group (1994)