D9.4v0.1 WSMX Invoker

WSMX Working Draft 13 June 2005

This version:
> http://www.wsmo.org/TR/d9/d9.4/v0.1/20050613
Latest version:
> http://www.wsmo.org/TR/d9/d9.4/v0.1/
Previous version:
> http://www.wsmo.org/TR/d9/d9.4/v0.1/20050426/
Editors:
> Mick Kerrigan
Authors:
> Mick Kerrigan

This document is also available in non-normative PDF version.

Table of contents

1. Introduction

1.1 Scope

The Web Service Execution Environment (WSMX) [Cimpian et al., 2005] is a reference implementation of the Web Service Modeling Ontology (WSMO) [Roman et al., 2004] that enables users to dynamically discover and select Web Services, mediate between the Ontologies used by the user and the Web Service, mediate between the external choreographies of the user and the Web Service and invoke the selected Web Service to achieve the users Goal. The Web Service Modeling Language (WSML) [de Bruijn et al., 2004], which is a formalism of WSMO, is used as input to and output from the WSMX System. A user describes his or her Goal in WSML and this Goal is matched against the Web Service descriptions, also described in WSML, that have been registered by service providers.

The Web Services Modeling Toolkit (WSMT) [Kerrigan, 2005a] is a lightweight framework for the creation and deployment of tools for Semantic Web Services. Current tools for the WSMT include the WSML Editor [Kerrigan, 2005b] and the WSMX Data Mediation Mapping Tool [Mocan, 2005]. The WSML Editor is a tool for the creation, editing and publishing of Ontologies, Web Services, Mediators and Goals described in the WSML language, while the WSMX Data Mediation Mapping Tool is used to create mappings between disparate Ontologies to resolve heterogeneity issues, these mappings can then be used by the WSMX

System at runtime to perform instance transformation from instances of one Ontology to another.

The WSMX Invoker is a tool for invoking the entrypoints of the WSMX System using WSML documents. It replaces the existing "back-end application" used in WSMX demos. The tool will use the inbuilt communication manager of the WSMT to perform invocations of the asynchronous service entrypoints of WSMX and will use the communication managers inbuilt receiving component to receive any results sent by the WSMX System (see.[Kerrigan, 2005a] for more information on the WSMT communication manager).

1.2 Purpose of this Document

The purpose of this document is to outline the WSMX Invoker plug-in for the Web Service Modeling Toolkit (WSMT). The WSMX Invoker is a graphical tool for invoking the Web Service entrypoints of the WSMX execution environment. It is also possible to monitor incoming and outgoing messages to/from the WSMT communication component.

1.3 Downloading the WSMX Invoker

The WSMX Invoker is distributed as a plug-in for the Web Services Modeling Toolkit (WSMT). The latest version of the WSMX Invoker can be obtained by downloading the WSMT. The WSMT is available at http://www.sourceforge.net/projects/wsmx/.

2. WSMX Invoker

The WSMX Invoker is currently a basic tool that essentially exposes the functionality of the WSMT Communication Manager (both invoker and receiver) in a graphical way, more information on the WSMT Communication Manager can be found in [Kerrigan, 2005a]. The WSMX Invoker is comprised of three tabs:

2.1 Send Message Tab

This tab in the WSMX invoker is used to send messages to Web Services that have been configured in the WSMT's preferences dialog. All of the available services are displayed to the user using a set of drop downs, which allow the user to choose the system, context, service, port-type and operation to be invoked. The user can then also specify a value for each of the input parts that the selected operation expects. Currently only primitive soap data types are supported for the input parts, however future work will look at using complex types with the Web Service Invocation Framework (in the WSMT receiver), the support set of complex types should be at least enough to support those used in the Web Service entrypoints of the WSMX System.
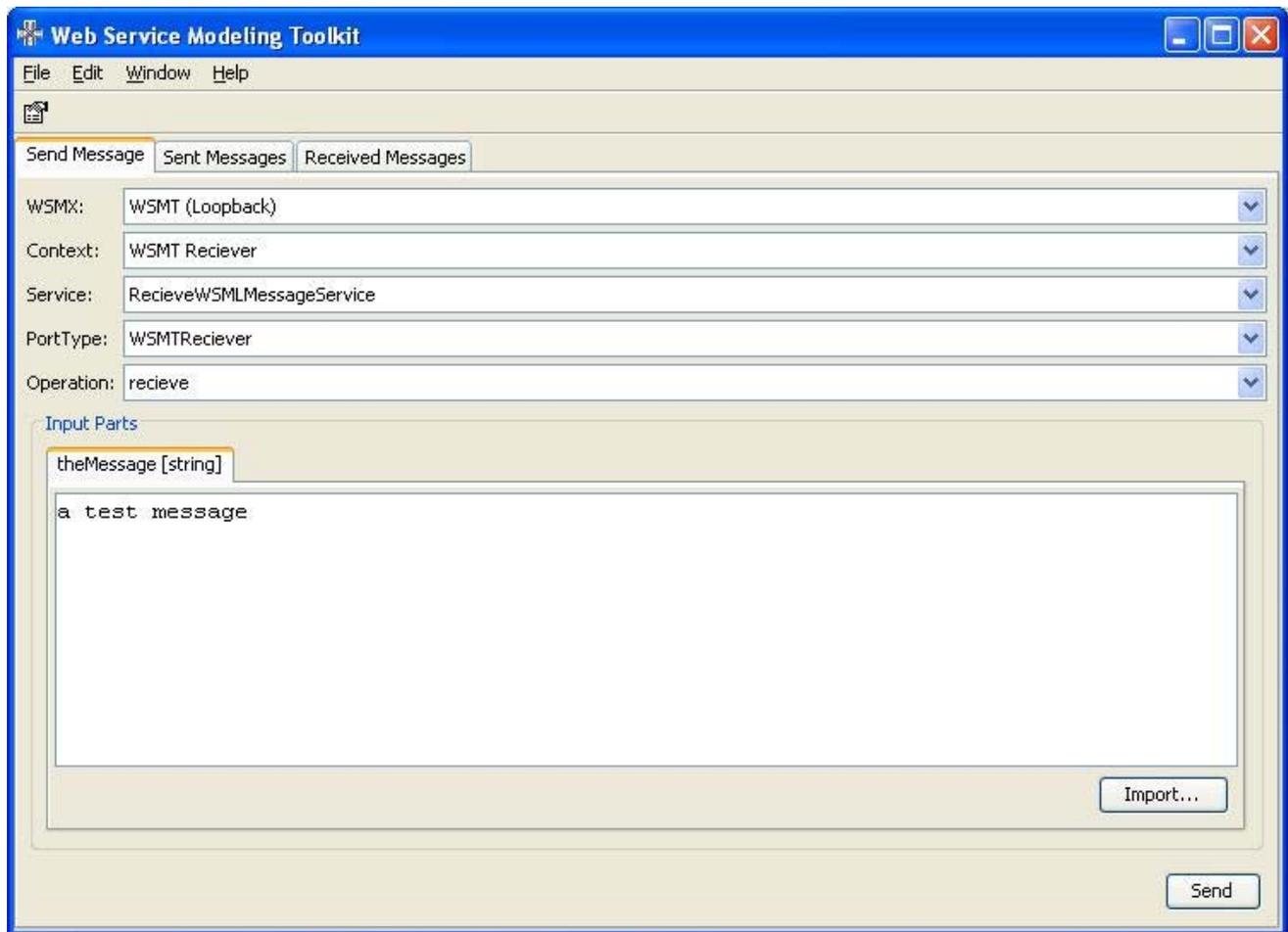
Figure 1. Send Message Tab

2.2 Sent Messages Tab

It is possible for any tool implemented as a plug-in to the WSMT to register itself for notifications of messages sent by the WSMT's invoker. This tab in the WSMT Invoker listens for every message sent out by the WSMT and displays these messages in a table ordered by the date when they where sent.
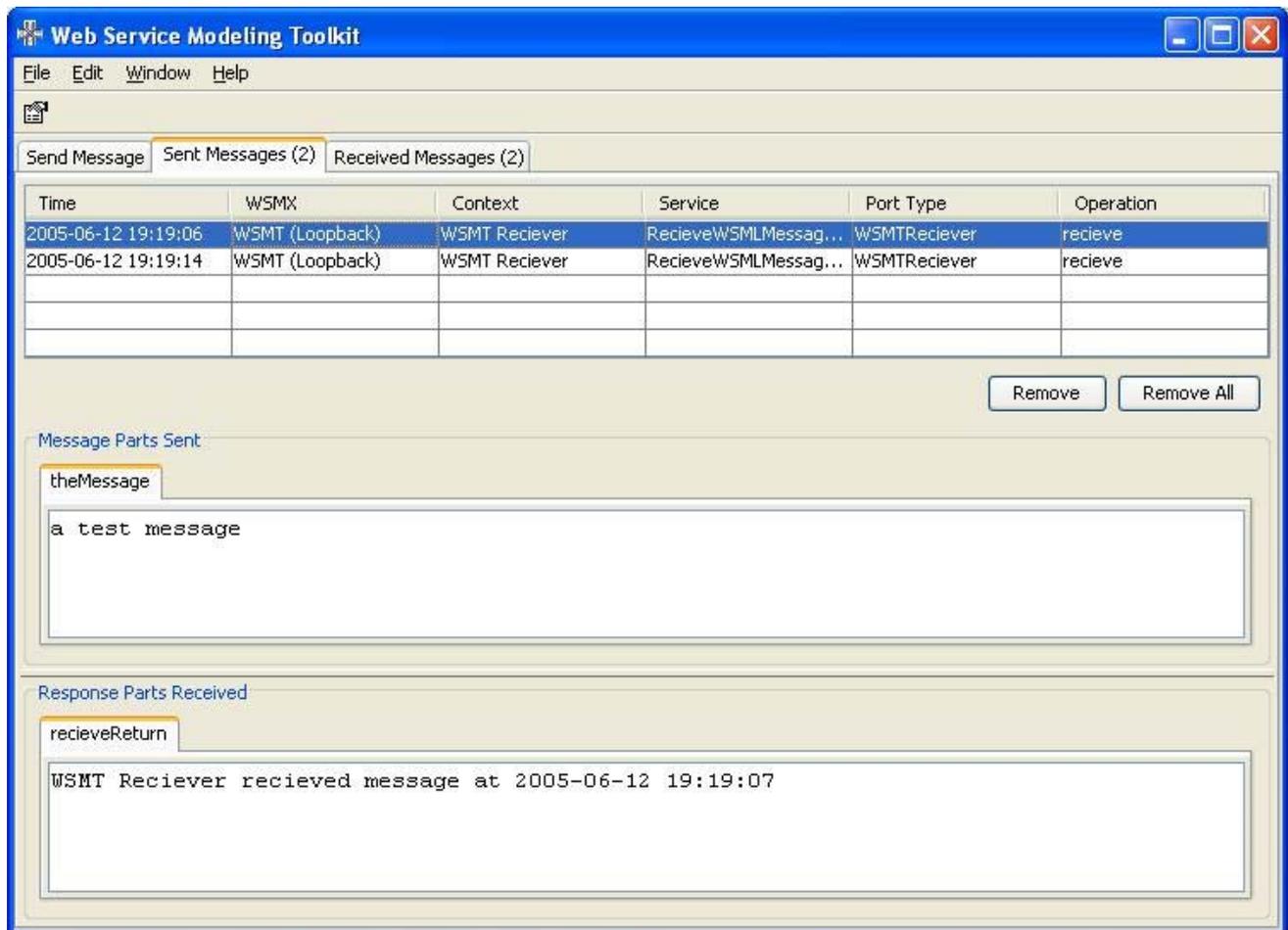
Figure 2. Sent Messages Tab

2.3 Received Messages Tab

Similar to the WSMT Invoker it is possible for a plug-in to the WSMT to register itself for notifications of messages received by the WSMT's receiver. The received messages tab listens for all messages received by the WSMT and displays them in a table ordered by the data when they where received.
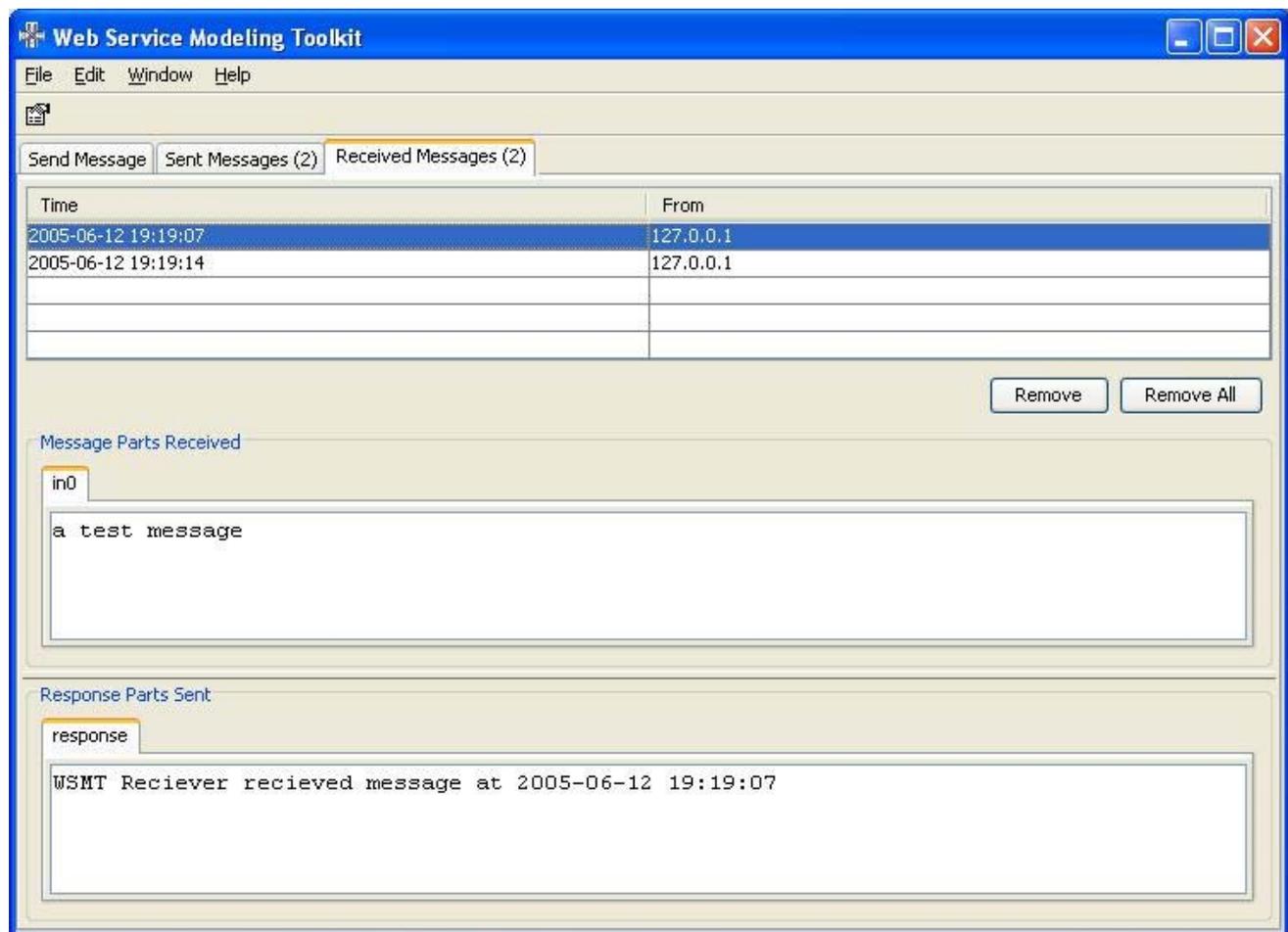
Figure 3. Received Messages Tab

3. Future Work

As more features are added to the WSMT Communication Manager more functionality will be added to the WSMT Invoker to reflect these new features. Two important changes to the invoker component of the communication manager is the support for complex types using the Web Service Invocation Framework and the inclusion of WS Addressing in the soap message headers to allow the services called to call the WSMT back.

References

[Cimpian et al., 2005] E. Cimpian, T. Vitvar, and M. Zaremba, Overview and Scope of WSMX, http://www.wsmo.org/TR/d13/d13.0/

[de Bruijn et al., 2004] J. de Bruijn, H. Lausen, and D. Fensel, The WSML Family of Representation Languages, http://www.wsmo.org/TR/d16/d16.1/

[Kerrigan, 2005a] M. Kerrigan, Web Services Modeling Toolkit (WSMT), http://www.wsmo.org/TR/d9/d9.1/

[Kerrigan, 2005b] M. Kerrigan, WSML Editor, http://www.wsmo.org/TR/d9/d9.2/

[Mocan, 2005] A. Mocan, WSMX Data Mediation, http://www.wsmo.org/TR/d13/d13.3/

[Roman et al., 2004] D. Roman, H. Lausen, U. Keller (eds.): Web Service Modeling Ontology (WSMO), http://www.wsmo.org/TR/d2/

Acknowledgement

The editors would like to thank to all the <u>members of the WSMX working group</u> for their advises and inputs to this document.

<u>webmaster</u>