



LTSA Eclipse and SENSORIA Help Guide

LTSA Eclipse and SENSORIA Help Guide

© 2007 DoC Imperial College London

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: November 2007 in (wherever you are located)

Publisher

*Department of Computing,
Imperial College London*

Managing Editor

Dr. Howard Foster

Technical Editors

Dr. Howard Foster

Special thanks to:

SENSORIA IST Project for supporting our work financially and providing real case studies for us to test our ideas with....

Table of Contents

Part I Introduction	1
Part II Glossary and Acronyms	1
Part III Installation	1
Part IV Tutorials	1
1 MSC Trace from BPEL Process	1
2 MSC Trace of UML Interactions	6
Part V Plug-in Reference	10
1 Analysis	10
Model Checking	10
LTSA	10
Info	10
Functions	10
Web Services	11
WS-Engineer	11
2 Modeller	11
MSC	11
UML	11
Part VI Appendix	11
Index	0

1 Introduction

The LTSA Eclipse plug-ins for the SENSORIA Case Tool are a set of services providing interfaces to the LTSA suite of model-checking tools. The SENSORIA Case Tool provides a framework for offering these services and can be used to script the execution of service methods in a workflow style. The set of LTSA services and functions currently available includes the following:

- Core LTSA safety, liveness property model checking
- WS-Engineer (WS-BPEL and WS-CDL) FSP models and property checking
- LTSA MSC (MSC from LTSA traces and from MSC XML)
- LTSA UML (LTSA and FSP models and analysis from UML XMI Models)

2 Glossary and Acronyms

Acrony Description

m

FSP	Finite State Processes
LTSA	The Labelled Transition System Analyser
MSC	Message Sequence Chart

3 Installation

The LTSA Eclipse SENSORIA Case Tool plug-ins require the LTSA Eclipse plug-in, and can be installed as features from the main LTSA Eclipse Update site (which is used via the **Help -> Software Updates -> Find and Install** option in the Eclipse IDE).

Please see the LTSA Eclipse help guide for further details of installing the core LTSA Eclipse and associated plug-ins.

4 Tutorials

Featured tutorials include:

1. Illustrated MSC Trace from the safety check of a BPEL process
2. MSC Trace of a UML Interaction Diagram

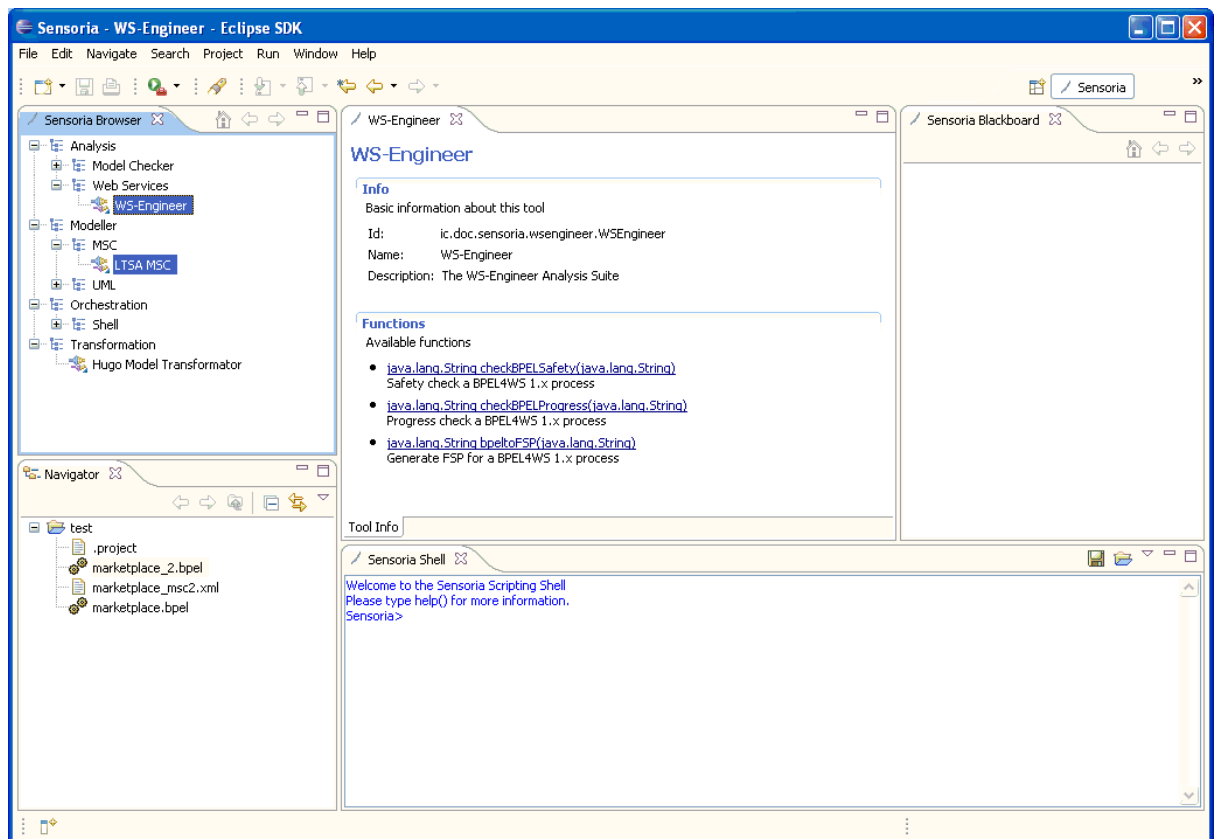
4.1 MSC Trace from BPEL Process

This tutorial demonstrates producing a Message Sequence Chart trace from the analysis of a Business Process Execution Language for Web Service (BPEL4WS) process specification. Each step is illustrated with an example figure.

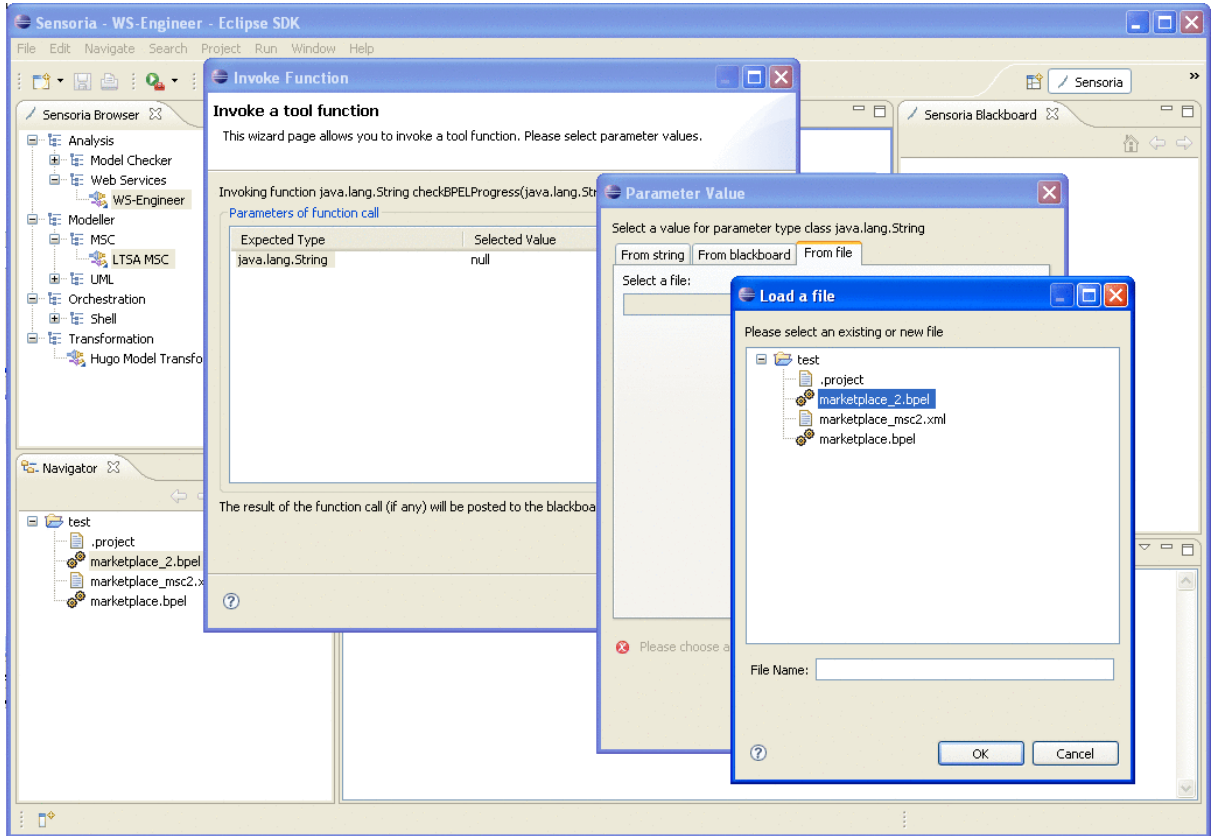
1. Locate the SENSORIA Browser view and check that the following plug-ins

have been installed.

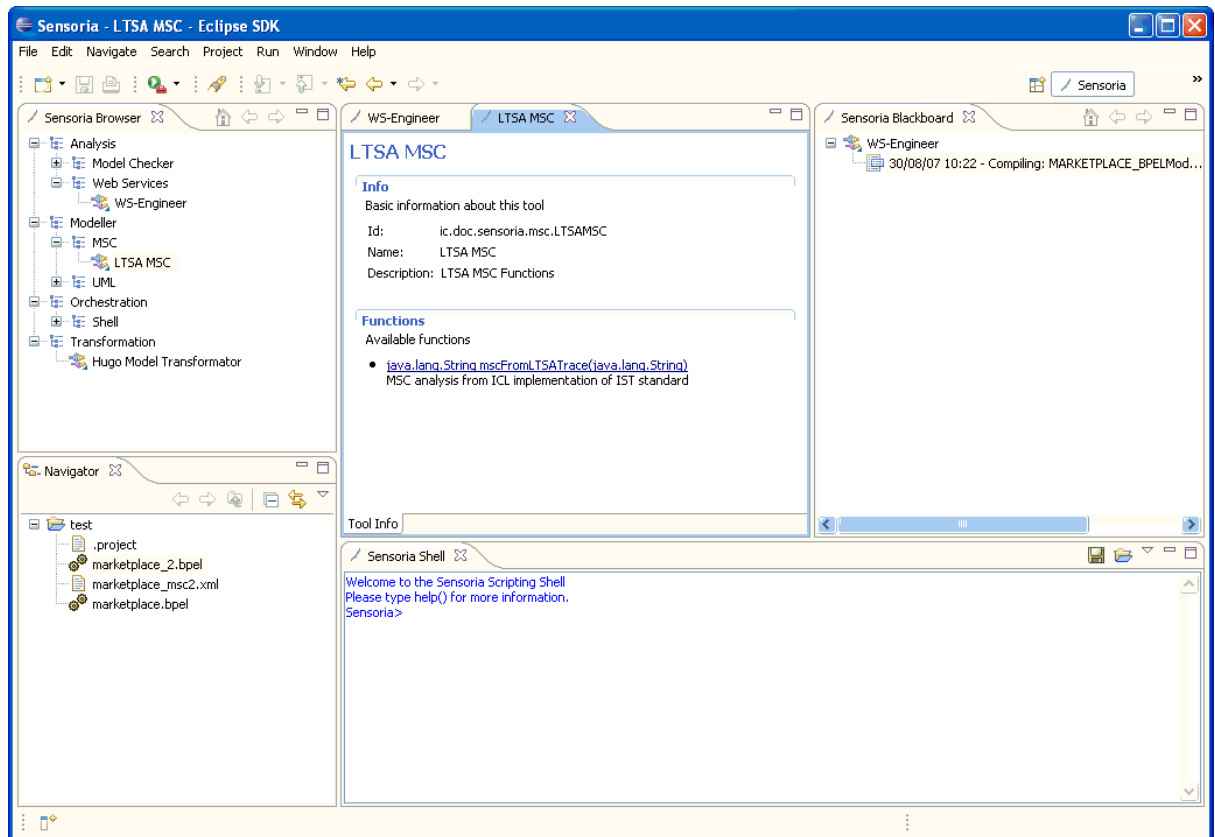
Analysis -> Web Services -> WS-Engineer
Modeller -> MSC -> LTSA MSC



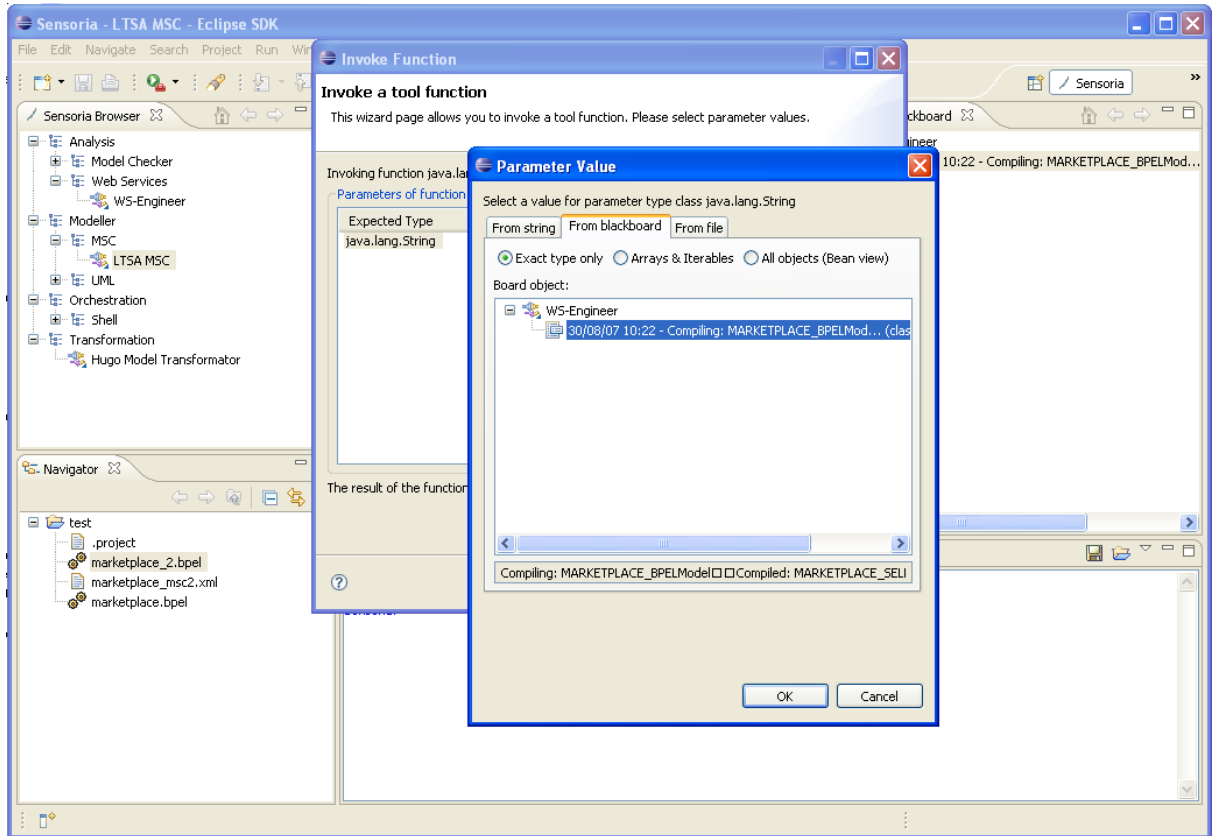
2. Select and invoke the `checkBPELProgress` method on the WS-Engineer Tool info view, and then use the invoke function wizard to select a BPEL process file or string.



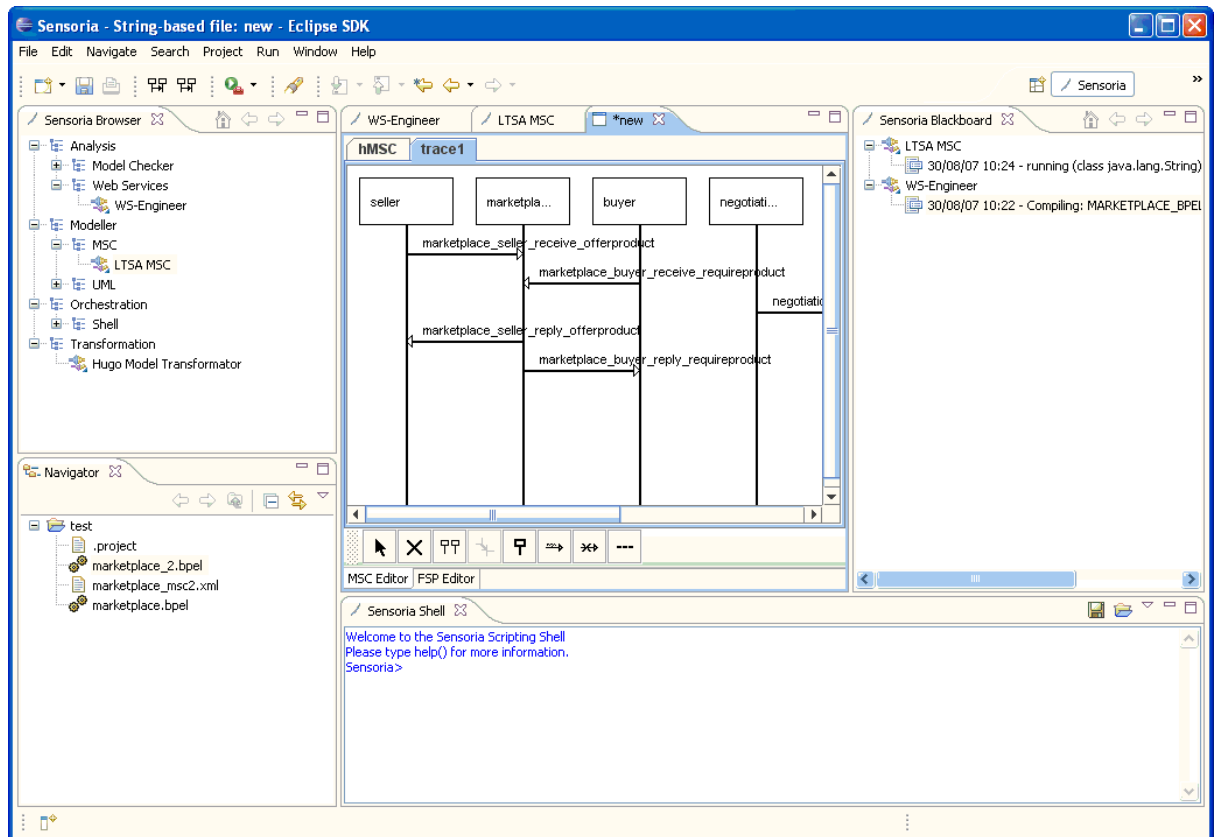
3. If the call was successful, check that the result is on the SCT Blackboard view.



4. Use the result on the SCT Blackboard as Input to the LTSA MSC plug-in method `mscFromLTSATrace`.



5. If successful, a new LTSA MSC editor window will display with a graphical trace of the BPEL interactions. If the call is not successful, please check the result on the SCT Blackboard.



4.2 MSC Trace of UML Interactions

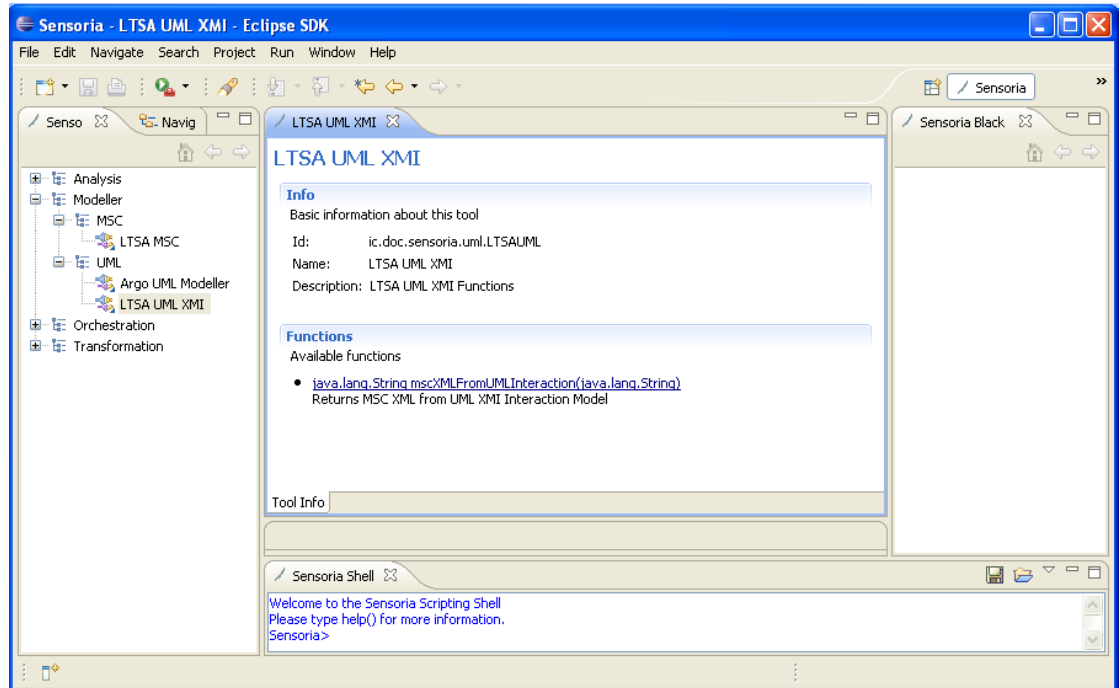
This tutorial demonstrates producing a Message Sequence Chart trace from the analysis of a UML2 interaction specification (exported to UML XMI 2 standards). Each step is illustrated with an example figure.

Note that this tutorial requires sources files downloaded from here: <http://www.doc.ic.ac.uk/ltsa/eclipse/sensoria/tutorials/umlmscexample1.zip>

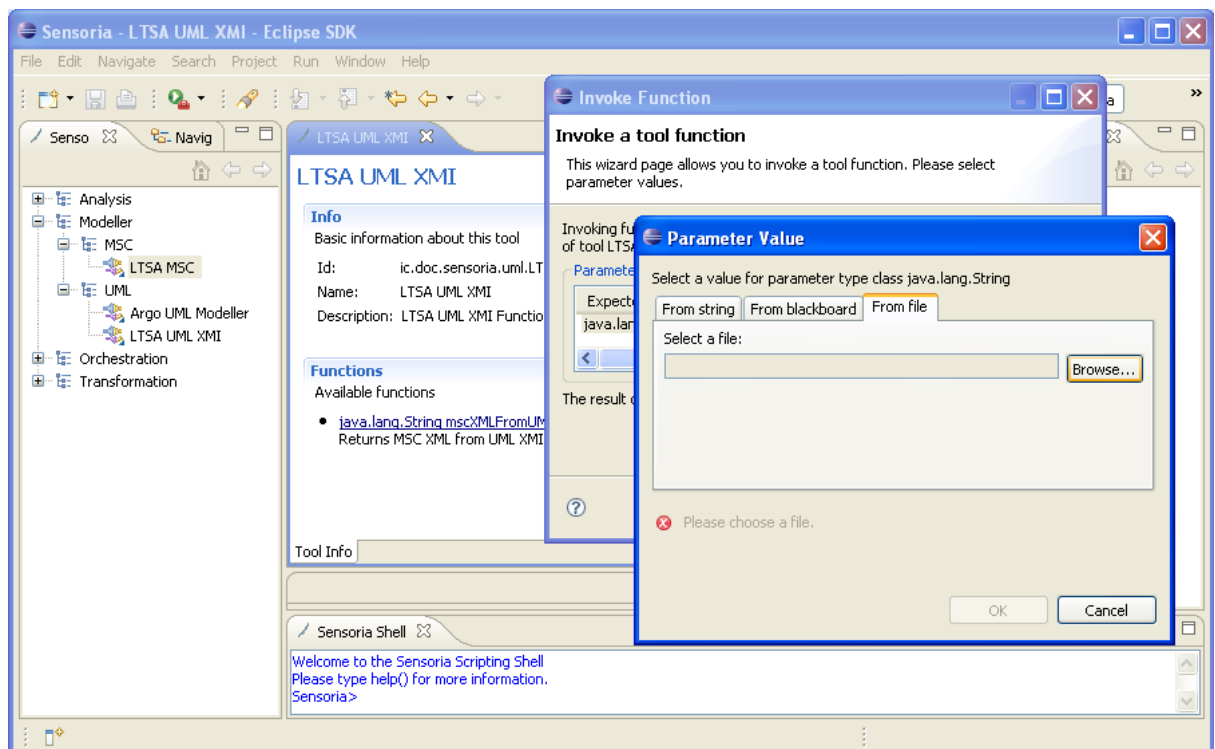
1. Locate the SENSORIA Browser view and check that the following plug-ins have been installed.

Modeller -> MSC -> LTSA MSC

Modeller -> UML -> LTSA UML XMI

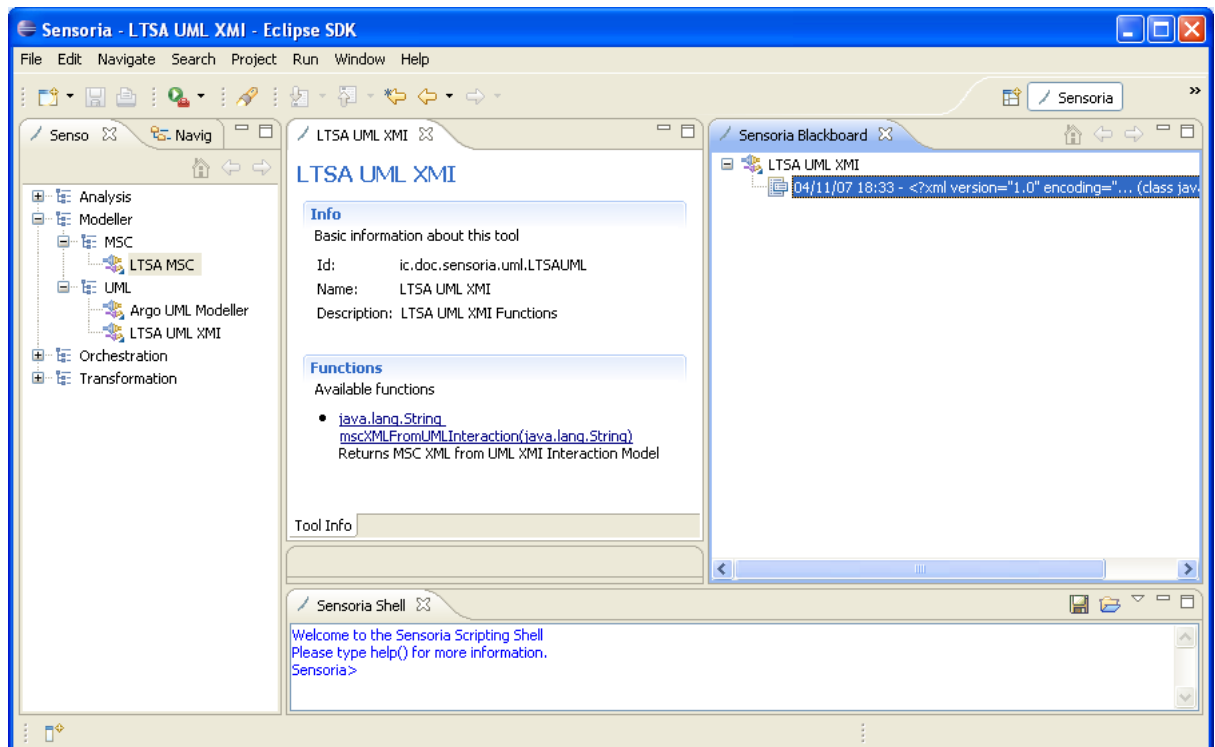


2. Select the LTSA UML XMI plug-in and execute the `mscXMLFromUMLInteraction` service method, using a valid UML2 XMI document containing the UML Interaction.

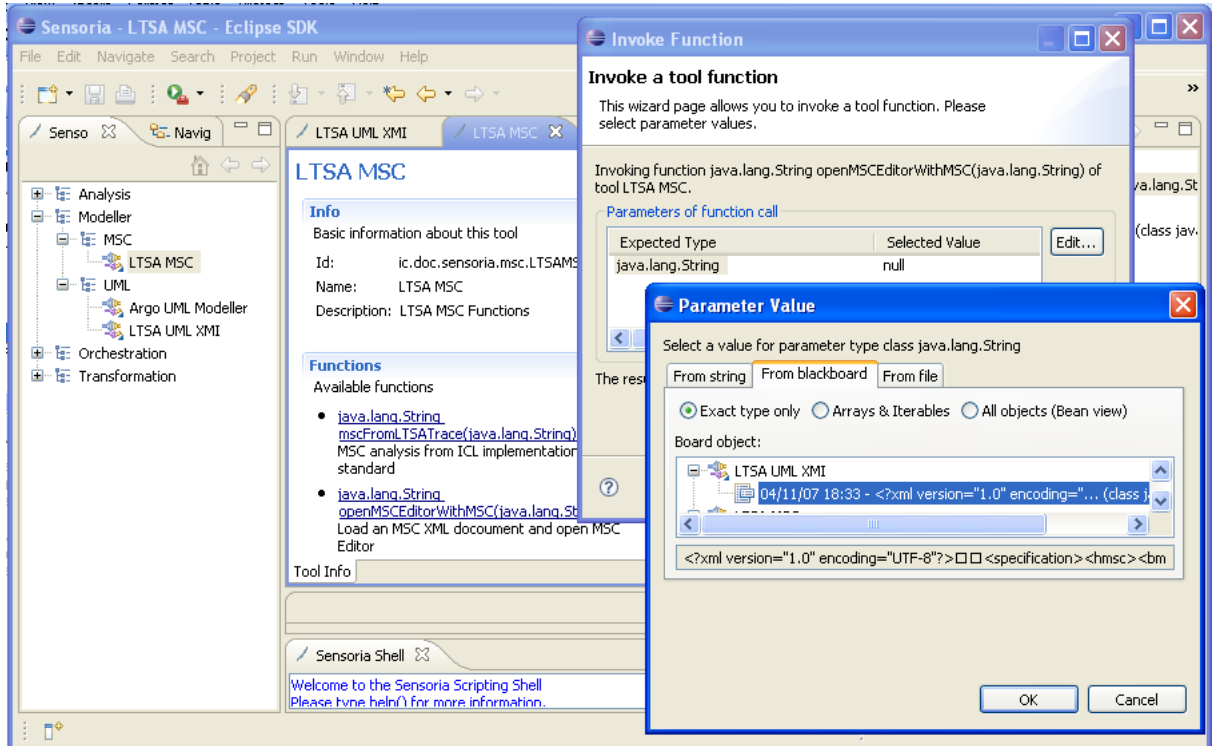


3. If successful, an XML document will be placed on the Sensoria Blackboard. If the method is not successful, an error message will be placed on the

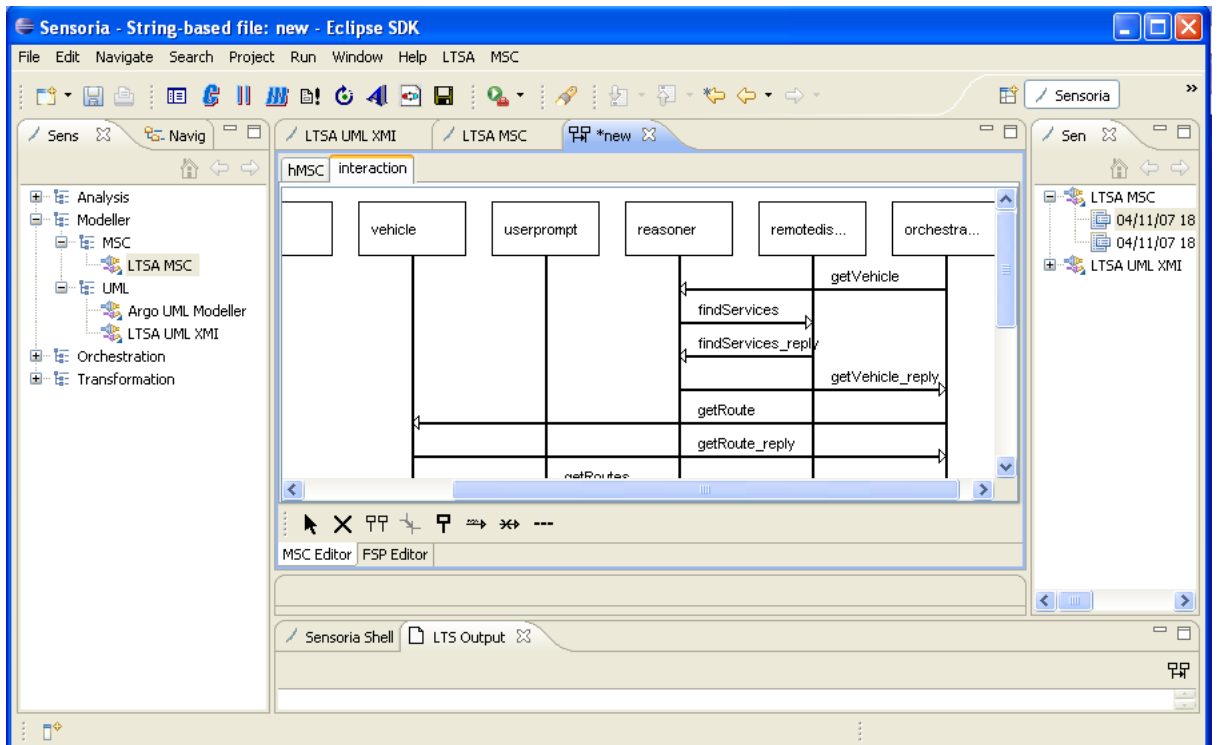
Sensoria Blackboard.



4. Select the LTSA MSC plug-in and execute the `openMSCEditorWithMSC` service method, using a successful result of the LTSA UML XMI service method in 3.



5. If successful, the LTSA MSC Editor will be opened displaying a trace from the XML document generated in 3. If the method is not successful, an error message will be placed on the Sensoria Blackboard.



5 Plug-in Reference

These plug-in references refer to specific details about each LTSA SENSORIA plug-in provided.

The plug-ins are categorized by Software Engineering topic. For example, by Analysis plug-ins or by Modelling plug-ins.

5.1 Analysis

The Department of Computing, Imperial College London provides the core LTSA model checker in the Analysis services within the SENSORIA browser.

5.1.1 Model Checking

5.1.1.1 LTSA

ID: ic.doc.sensoria.ltssa
Name: LTSA
Description: Core LTSA Functions plug-in

5.1.1.1.1 Info

ID: ic.doc.sensoria.ltssa
Name: LTSA
Description: Core LTSA Functions plug-in

5.1.1.1.2 Functions

Analyse

Provides a method to analyse a Finite State Process model.

Input: An FSP model (type: String)

Output: A Trace (type: String)

Safety

Provides a method to safety check a Finite State Process model.

Input: An FSP model (type: String)

Output: A Trace (type: String)

Progress

Provides a method to progress check a Finite State Process model.

Input: An FSP model (type: String)
Output: A Trace (type: String)

Compile

Provides a method to compile an FSP source model to a Finite State Machine.

Input: An FSP model (type: String)
Output: Result of compilation (type: String)

5.1.2 Web Services

Enter topic text here.

5.1.2.1 WS-Engineer

Enter topic text here.

5.2 Modeller

Enter topic text here.

5.2.1 MSC

Enter topic text here.

5.2.2 UML

Enter topic text here.

6 Appendix