1 Configuration

The WSMT (Web Service Modeling Toolkit) is installed within the SHAPE_Toolsuite. WSMT may work in conjunction with Web Service Execution Environment (WSMX)[1]. You may consider configuring WSMT in order to make it aware of the execution environment.

Step 1. Run SHAPE_Toolsuite Eclipse bundle (see section **Error! Reference source not found.**)

Step 2. Open SEE perspective of WSMT.

Step 3. On the left panel, right click on the “server” and open “properties” to set the IP and ports in order to connect the WSMT to Web Service Execution Environment (WSMX). In this case, IP is ‘localhost’, HTTPport is 8080 and servlet port is 8050

Note: WSMT is an ongoing development will have many updated versions in near future. The latest versions of the WSMT can be downloaded from http://sourceforge.net/projects/wsmt

2 Browsing ontologies, goals and Web Service Descriptions in WSMT

The WSMT is a part of the SHAPE_Toolsuite Eclipse bundle. In order to start using it, change the perspective to WSML.
After the WSMT is started, on the left side of WSMT GUI, the “Navigator” view shows all the projects in your workspace. For this tutorial session, the WSMT contains all the WSMO Ontologies, Web Services and Goals needed to execute the use case in the project called “WSMO Tutorial”. The workspace also contains some other WSML examples that you can use to understand the WSMT functionality.

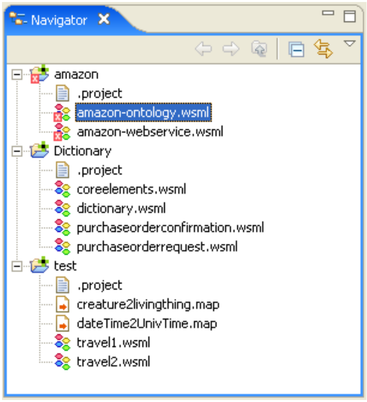
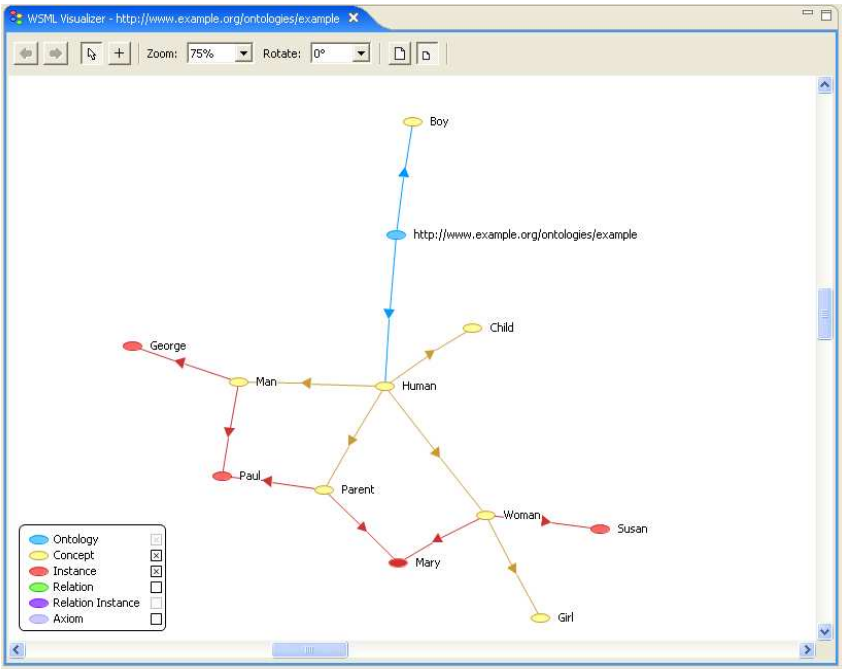


Figure: WSMT Navigator

There are two main editors for viewing and manipulating WSMO descriptions in the WSML formalism, which are the required features of WSMT in context of SHAPE. These are the WSML Visualizer and the WSML Text Editor and instructions for using them are provided in sections 3.1 and 3.2.

[1] Web Service Execution Environment (WSMX) www.wsmx.org

3 WSML Visualizer



WSML Visualizer - http://www.example.org/ontologies/example

Zoom: 75%

Rotate: 0°

Figure: WSMT Visualizer

The WSML Visualizer provides a graphical representation of the information contained within a WSML document, using a directed graph. In this part of the tutorial we will look at the Simpsons ontology, which is a schema and some instances related with the Simpsons television show. This example wsml file can be found in *"SHAPE_Toolsuite Installation Folder / examples / the-simpsons-ontology.wsml"*. Copy it in a *Simpsons* project.

Step 1. Open "The Simpsons" example project by clicking the plus on the left hand side of the folder icon.

Step 2. Right click on "the-simpsons-ontology.wsml" in the navigator view, choose "Open With → WSML Visualizer"

In the visualizer you will see the blue node at the centre that represents the ontology contained within this WSML file. The other nodes in the representation are concepts (yellow) and instances (red). The graph is automatically layed-out using a spring layout algorithm however the representation is fully interactive and by clicking on a node and dragging it the representation can be moved to obtain the best layout.

Step 3. Manipulate the graph using the dragging feature to get the best representation.

The visualization provided in this editor is fully interactive and allows the user to edit the WSMO Ontology, Web Service, Goal or Mediator being visualized. In the following steps we will add a new concept, some attributes and an instance to the Simpsons ontology. The concept that we shall add will represent the concept of a public park and the instance will represent the park located in Springfield.

Step 4. Right click on the yellow concept node called "place" and choose "Add → Sub Concept"

Step 5. In the "New Concept ID" dialog enter "public_park" and select OK (NOTE: a new concept node is added to the visualization).

The visualizer separates the complexities in the ontology into multiple levels, in order to start manipulating the attributes of a concept we need to “step down” to the concept semantic level. We do this by double clicking on a concept.

Step 6. Double click on the “public_park” concept node (NOTE: the view is constrained to information about the selected concept).

Step 7. Right click on the “public_park” concept node and choose “Add → Attribute”

Step 8. In the “New Attribute ID” dialog enter “hasLocation” and select OK (NOTE: we now have an error in our ontology as the attribute hasLocation has no range).

Step 9. Right click on the brown “hasLocation” attribute node and choose “Add → Attribute Range (Concept)”

Step 10. In the “Select a Concept” dialog, expand the “place” concept in “The Simpsons Ontology”, choose “town” and select OK. (NOTE: the error in the ontology is now removed.

Now that we have created our concept and added an attribute we can create an instance of that concept. Note that there are two attributes available for use on the instance. The first is the hasLocation that we just created and the second is the hasName attribute inherited from the “place” super concept of “public park”.

Step 11. Right click on the “public_park” concept again and choose “Add → Instance”

Step 12. In the “New Instance ID” dialog enter “springfield_public_park” and select OK

Step 13. Double click on the new “springfield_public_park” instance node to move to the instance semantic level.

We have created our instance, but we still need to assign values to the attributes of the instance.

Step 14. Right click on “springfield_public_park” and choose “Add Attribute”

Step 15. Choose the “hasLocation” attribute from the “public_park” concept and select OK. (NOTE: You will automatically be shown the “Select an Instance” dialog).

Step 16. Choose the “Springfield” instance from the “town” concept and select OK.

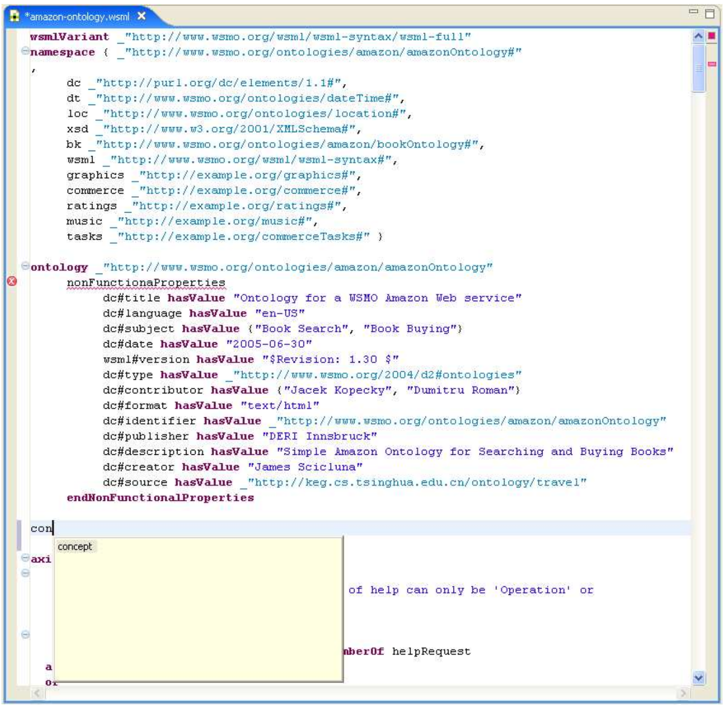
Step 17. Right click on “springfield_public_park” and choose “Add Attribute”

Step 18. Choose the “hasName” attribute from the “place” concept and select OK (NOTE: You will automatically be shown the “Specify a Data Value” dialog).

Step 19. In the Data Value text field enter “Springfield Public Park” and select OK.

Step 20. Save the ontology by using the Ctrl-S shortcut or “File → Save”.

4 WSML Text Editor



```
wsmlVariant _"http://www.wsmo.org/wsml/wsml-syntax/wsml-full"
namespace { _"http://www.wsmo.org/ontologies/amazon/amazonOntology#"
,
     dc _"http://purl.org/dc/elements/1.1#",
     dt _"http://www.wsmo.org/ontologies/dateTime#",
     loc _"http://www.wsmo.org/ontologies/location#",
     xsd _"http://www.w3.org/2001/XMLSchema#",
     bk _"http://www.wsmo.org/ontologies/amazon/bookOntology#",
     wsml _"http://www.wsmo.org/wsml/wsml-syntax#",
     graphics _"http://example.org/graphics#",
     commerce _"http://example.org/commerce#",
     ratings _"http://example.org/ratings#",
     music _"http://example.org/music#",
     tasks _"http://example.org/commerceTasks#" }

ontology _"http://www.wsmo.org/ontologies/amazon/amazonOntology"
     nonFunctionaProperties
          dc#title hasValue "Ontology for a WSMO Amazon Web service"
          dc#language hasValue "en-US"
          dc#subject hasValue {"Book Search", "Book Buying"}
          dc#date hasValue "2005-06-30"
          wsml#version hasValue "$Revision: 1.30 $"
          dc#type hasValue _"http://www.wsmo.org/2004/d2#ontologies"
          dc#contributor hasValue {"Jacek Kopecky", "Dumitru Roman"}
          dc#format hasValue "text/html"
          dc#identifier hasValue _"http://www.wsmo.org/ontologies/amazon/amazonOntology"
          dc#publisher hasValue "DERI Innsbruck"
          dc#description hasValue "Simple Amazon Ontology for Searching and Buying Books"
          dc#creator hasValue "James Scicluna"
          dc#source hasValue _"http://keg.cs.tsinghua.edu.cn/ontology/travel"
     endNonFunctionalProperties

con
axi
                                        of help can only be 'Operation' or

                                        mberOf helpRequest
  a
  or
```

Figure: WSMT Text-based Editor

The WSML Text Editor shows the contents of a WSML document, in the WSML human readable syntax, to the user with syntax highlighting, content assistance and many other features that help advanced users who are familiar with WSML to create and manage their WSML documents more efficiently.

Step 21. Right click on “the-simpsons-ontology.wsml” in the navigator view, choose “Open With → WSML Text Editor”

Step 22. Use the Ctrl-F shortcut to bring up the find dialog

Step 23. In the find enter “public_park” and click Find.

You can now see the human readable syntax of the concept that we created in the visualizer.

Step 24. In the find dialog click Find again.

You can now see the human readable syntax of the instance that we created in the visualizer.

5 Importing SoaML model

This section is to be completed once the SoaML transformation is completed. However, expected steps are as follows:

1. A function to import SoaML model
2. A function to transform SoaML model into WSML
3. Display generated WSML in Text based editor or formed-based editor