

IRS-III Browser User Guide

<http://kmi.open.ac.uk/project/irs>

IRS-III allows you to create and edit WSMO knowledge models for Goals Web Services and Mediators; publish Lisp code, Java code, Web applications and WSDL descriptions against those descriptions; and invoke Services through Achieve-Goal. In the following we give you some instructions for doing that using the IRS-III browser. Additionally, we provide some snapshots of an example for publishing a java class using existing descriptions. We hope this will get you started on creating your own semantic descriptions. Also, have a look at the Test applications provided for examples using the Java API.

1. Starting IRS-III Browser/Editor

- Double-click **IRSIIIBrowser** bat file.
- Enter a valid **IRS Server** (anfield.open.ac.uk) and **Port** number (3000). (fig 1.)
- Enter a valid **Username** (wsmo) and **Password** (wsmo) (fig. 1)

2. Using IRS-III Browser/Editor

- Double-click a tree node to expand/collapse the ontology tree.
- Select an ontology under which you want to create a new Goal, WS or Mediator
- Click a tree node (ontology, goal, WS, Mediator, class or slot) to see its description in the **Details** panel.
- Double-click any highlighted item in the **Details** panel to see its definition.
- Under *Knowledge Models* menu item select the types of ontologies (**Goal, WS, Mediator, Domain, Application**) to be loaded in the ontology tree.
- Select in the toolbar the types of knowledge items (**Classes, Goals, WS, Mediators, inherited items**) to be seen under the selected ontology.

3. Developing Semantic Services using IRS-III

1. Find or create an appropriate ontology (fig. 2)
2. Find or create an appropriate goal description. (fig. 2).
3. Find or create an appropriate mediator description (fig. 3)
4. Find or create a number of related WSs. (fig. 4).
5. Link the WSs to the goal created in 2 through mediators (capability used mediators) (fig. 4).
6. Use the WSs for publishing services (Java or Lisp code) (fig. 5).
7. Invoke services through Achieve Goal (fig. 6).

Example

We are going to semantically publish and invoke the KMIMultiply java class included in the IRS-III distribution package using existing Goal, WS and Mediator descriptions. Please start the IRS-III browser/Editor.



Figure 1. IRS-III Login

1. Double-click ontology **european-travel-service-descriptions** and select goal **multiply-goal**. Under Goal/WS/Mediator menu item, select Edit Goal/WS/Mediator. Make sure you enter inputs and output.

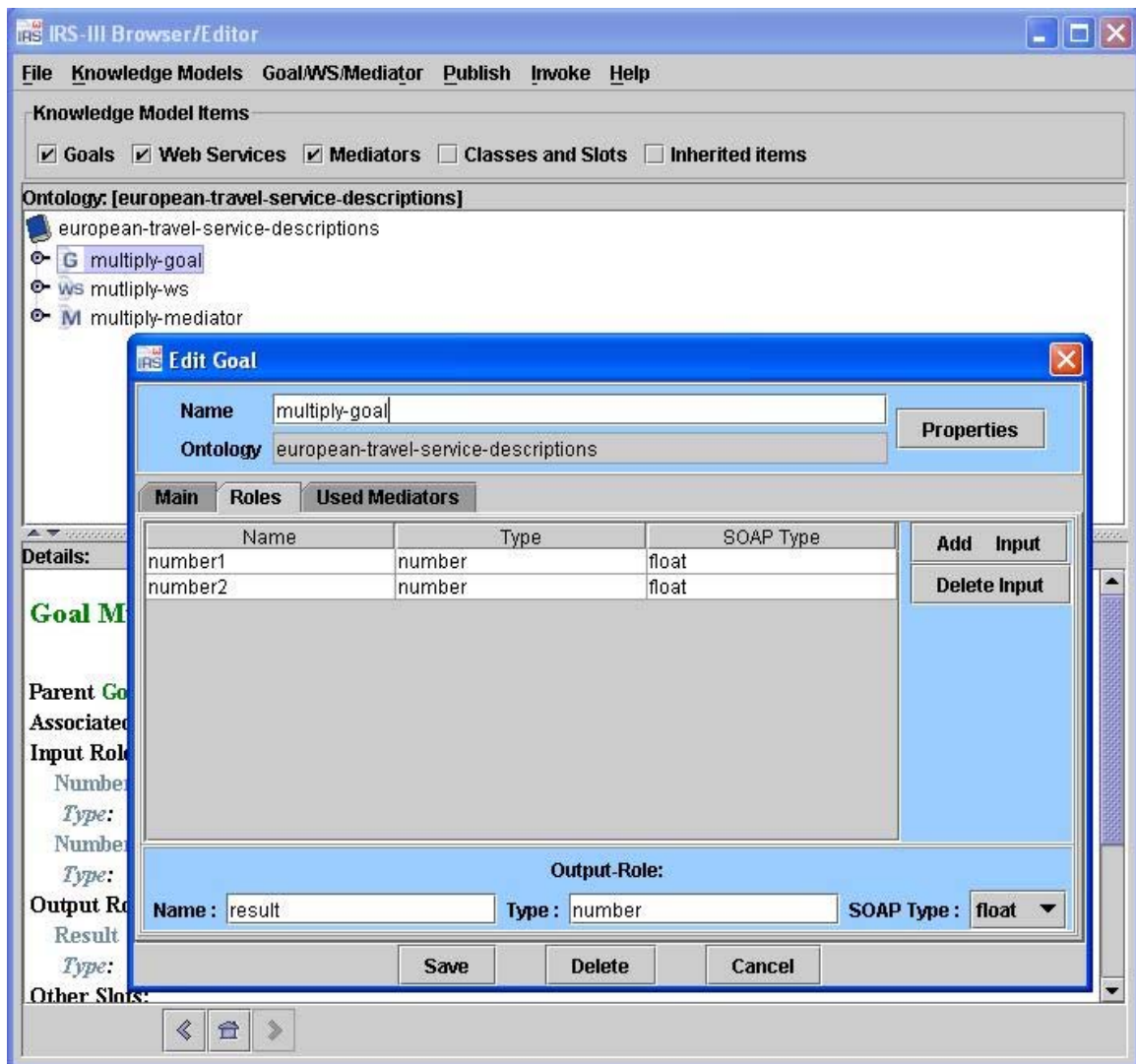


Figure 2. Editing existing Goal description.

2. Double-click ontology **european-travel-service-descriptions** and select mediator **multiply-mediator**. Under Goal/WS/Mediator menu item, select Edit Goal/WS/Mediator. Make sure Source component slot refers to the Goal to be solved.

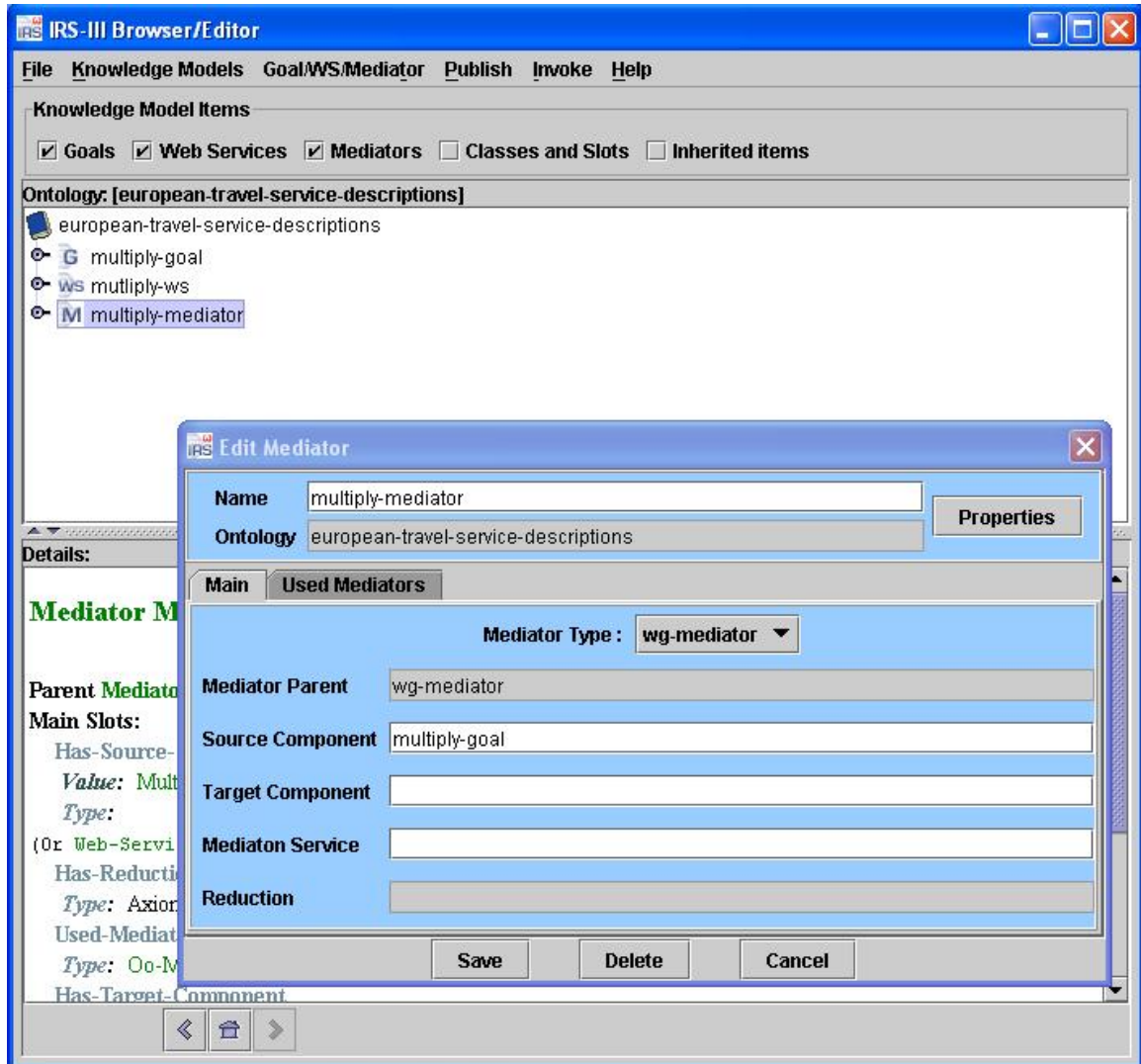


Figure 3. Editing existing Mediator description.

3. Double-click ontology **exchange-rate-provider-ontology** and select WS **multiply-ws**. Under Goal/WS/Mediator menu item, select Edit Goal/WS/Mediator. Inputs and output will be inherited from the Goal (connected via mediator). The WS capability used mediator is the one defined before.

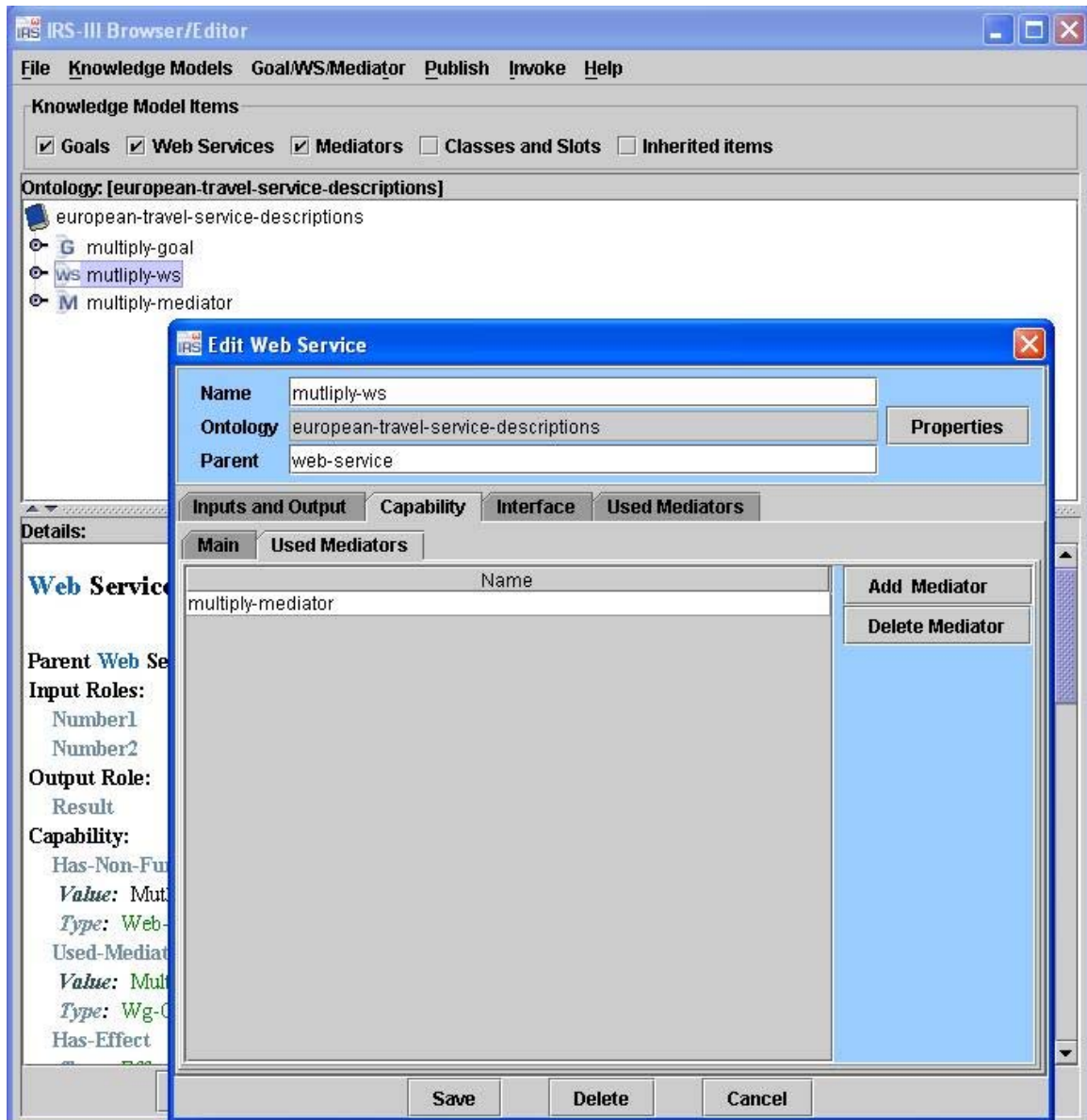


Figure 4. Editing existing WS description.

4. Double-click ontology **exchange-rate-provider-ontology** and select WS **multiply-ws**. Under *Publish* menu item, select *Java Class and Method*. Make sure the java Publisher (IRSPublisher.war) is deployed and the host (default is anfield.open.ac.uk) is correct. KMIExchange class file must be under /shared/classes in the same web server (e.g. Tomcat) as the publisher.

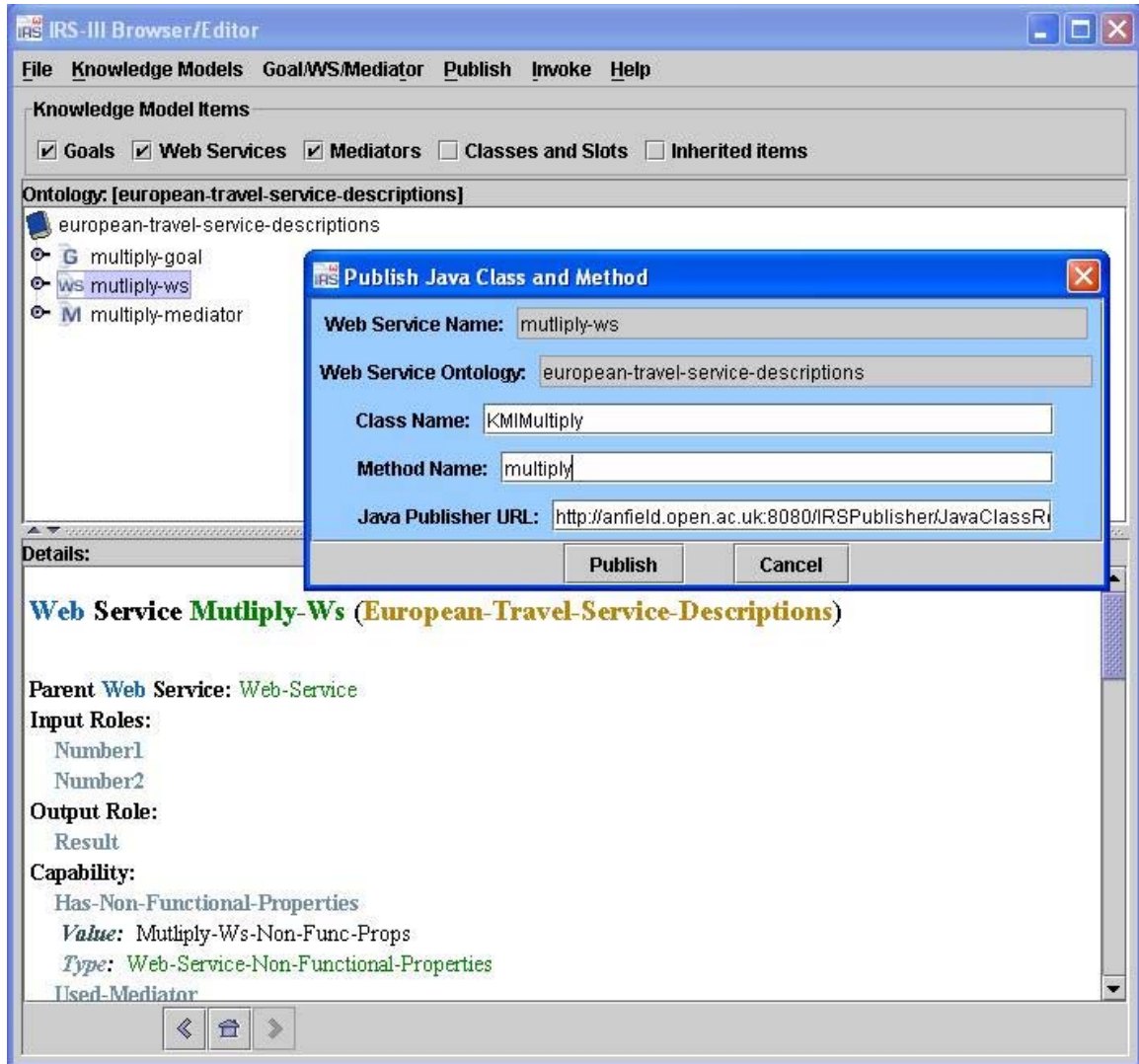


Figure 5. Publishing java class KMIMultiply and method multiply.

5. Double-click ontology **europaean-travel-service-descriptions** and select goal **multiply-goal**. Enter values for inputs. Under *invoke* menu item, select *AchieveGoalk*.

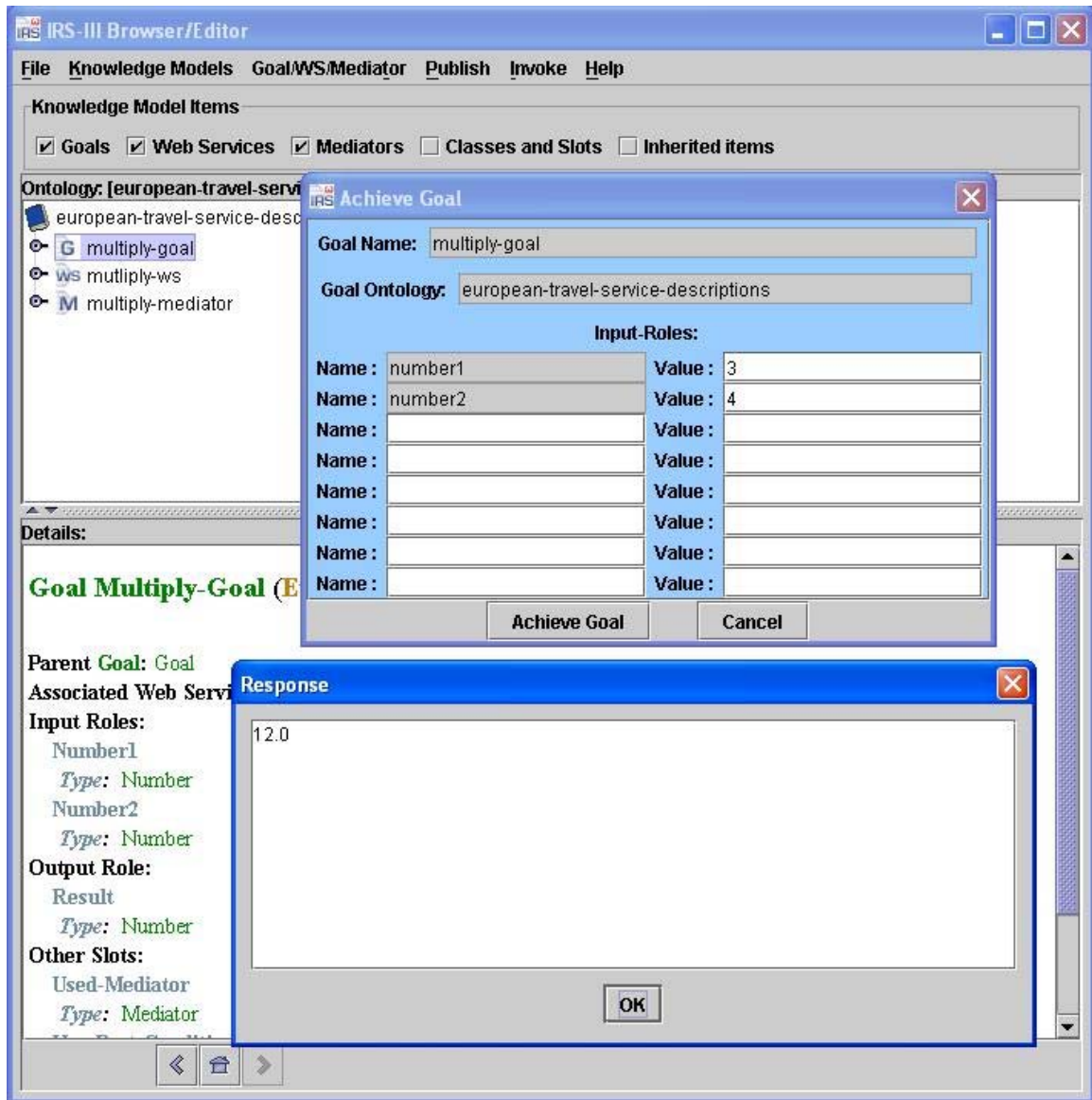


Figure 6. Invoking **multiply-goal** goal.