PANACEA Taverna tutorial

WP 3

Author	Marc Poch
Affiliation	UPF
Status	internal report
Version	3.0
Release	draft
Date	2012-05-05
History:	

Abstract: This document is a short tutorial for Taverna workflow editor and engine.

Contents

1. Introduction

This document is a tutorial for PANACEA users willing to run and design PANACEA workflows. It can be used as a guideline for users to access all the necessary information and material to start working.

The tutorial is divided according to the three development phases of the project: each of them adding new functionalities, corrections or improvements.

2. Platform version 1

Taverna is the workflow editor and engine used to develop workflows in PANACEA. It can combine SOAP and REST web services and can use plugins specific for certain tools or formats.

2.1. Technical description summary

Name: Taverna Workbench

Version: 2.4

OS: Windows, Mac, Linux

Documentation:
http://www.taverna.org.uk/documentation
http://www.mygrid.org.uk/dev/wiki/display/taverna/Documentation+and+Videos

2.2. Import SOAPLAB services

SOAPLAB web services, if deployed with the “typed interface” parameter activated, can be imported to Taverna as any other SOAP web service: using the WSDL file. On the other hand, there is a plugin specially designed for Soaplab which makes it easier to use these web services on Taverna and makes workflows much simpler.

To use the plugin you only need to click on “Import new services” and then “Soaplab service…”. Enter the Soaplab Server address with the web services you want to use like **http://{host}/soaplab2-axis/services/**. Don’t forget the final slash (/).

Afterwards the web services deployed on that Soaplab web server will be ready to be used in Taverna. Then, they can be easily added to the workflow by a simple drag and drop to the workflow diagram. If an error occurs, check that you didn't forget the "services/" part of the address and try to open the Spinet web client (if it's not working then the web service is down).

2.3. Building the workflow

There are numerous examples and videos which are recommended to study to learn how to build workflows in Taverna.

Tip: To connect the output of a web service to the input of the following web service we use the **output_url** and **the input_url** in PANCEA.

2.4. XPATH: getting specific data from XML files

In some situations you will need specific data from a XML file in your workflow. There are two options to use XPATH in Taverna:

1. Use the **"Xpath from text" local processor** which requires a bit of xpath knowledge.
http://www.mygrid.org.uk/dev/wiki/display/taverna/XPath+From+Text
2. The **"Xpath service"** from service templates. It is really easy to use and requires no xpath knowledge . You may need to update Taverna to be able to use this feature (Advanced menu -> updates).
http://www.mygrid.org.uk/dev/wiki/display/taverna/XPath

2.5. Lists

In most situations our workflow will be used to process many files. To handle that situation Taverna has "lists".

There is very good documentation about lists in:
http://www.mygrid.org.uk/dev/wiki/display/taverna/Lists+and+iterations

For complex lists situations, the following documentation can be very helpful:

http://www.mygrid.org.uk/dev/wiki/display/taverna/List+handling

3. Platform version 2

3.1. Taverna extra memory

Large files and more complex workflows make it necessary to assign more memory to Taverna to ensure it can handle all of the work.

In PANACEA, we use **800 MB** of memory and **280 MB** for classes.

To modify the memory parameter of Taverna use the following instructions depending on the OS:

http://www.mygrid.org.uk/dev/wiki/display/taverna/Increase+memory+allocation

3.2. Parallelization

When processing a large amount of files it is always interesting to add some kind of parallelization option to make global performance much better. From the PANACEA point of view, the idea would be to call web services many times instead of waiting for the previous document to finish.

Taverna can handle parallelization and queues inside a workflow. The workflow designer only needs to specify how many parallel processes of each web service the engine must call.

Follow these instructions:

http://taverna.knowledgeblog.org/2010/12/13/parallel-service-invocations/

Before setting the parallelization parameter in your workflows check the web service documentation (in the Registry, for example) to see the maximum amount of parallel jobs you are allowed to make by the service provider. Remember that the server's resources are limited. Use this parameter responsibly and if in doubt never put this parameter higher than 3.

3.3. Retries

For big workflows and many files it is important to make workflows more robust. Taverna has a "retry" system that can be configured for any web service inside the workflow.
You should use "retries" to try to execute one web service more times in the case that it fails.
http://www.youtube.com/watch?v=-jWQH9_u370

3.4. Polling techniques with Soaplab

Taverna has a timeout limit. If the answer from the web service takes too long, Taverna thinks that something went wrong. To avoid this you must use "polling". This can be very important for long lasting executions due to large data or long processes.

"Polling" is a system that allows the client to periodically make requests to the Soaplab server to check if the task has finished. This way the client doesn't have to wait, it simply makes short questions which are answered very quickly by the server avoiding the timeouts.

To make use of the Soaplab "polling" system in Taverna follow these instructions:
http://www.mygrid.org.uk/usermanual1.7/soaplab_processor.html#soaplab_polling

3.5. Useful workflow examples

How to create temporary files: http://www.myexperiment.org/workflows/1604.html

How to add conditions to a workflow: http://www.myexperiment.org/workflows/157.html

4. Platform version 3

It's important to use Taverna workbench 2.4 due to its robustness.