PANACEA 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 PANACEA users.

Contents

1. Introduction 2

2. Users 2

2.1. PANACEA Service Users 2

2.1.1. Find web services (the registry) 3
2.1.2. Run web services. 3
2.1.3. Design and run workflows. 3
2.1.1. Find and share already designed workflows 3
2.2. PANACEA Service Providers 3
2.2.1. Deploy web services 3
2.2.2. Register and annotate web services. 3
3. Travelling Object 4
4. Common interfaces 4
5. FAQs 4
5.1. Soaplab FAQs 4
5.1.1. General 4
5.1.2. Installation / ACD files development 4
5.1.3. Deployment and configuration 5
5.2. Taverna FAQs 5
5.2.1. General 5
5.2.2. Designing workflows 6
5.2.3. Running workflows 7
5.3. PANACEA registry (Biocatalogue instance) FAQs 7
5.3.1. Registering web services 7
5.3.2. Searching web services 7

1. Introduction

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

2. Users

PANACEA users can be divided in two groups: “PANACEA Service Users” and “PANACEA Service Providers”.

2.1. PANACEA Service Users

These users are the ones willing to obtain language resources, analysis, etc. They want to make use of the tools but they do not have or need a deep knowledge of the technologies behind the web services. They make use of the tools because of their functionalities. Their basic tasks are listed in the following sections.

2.1.1. Find web services (the registry)

The registry is the first point of entry to the platform. It is where users must go to locate the web services that provide the desired functionalities. PANACEA registry can be found on http://registry.elda.org.

2.1.2. Run web services.

Once a web service of interest is located, the user may want to try it before using it in a workflow or larger experiment; or perhaps the desired task simply requires only one web service, therefore needing only a simple execution. As most of the web services in PANACEA are deployed using Soaplab, the best way to test them is by using the "Spinet" web client. Spinet web client is a simple web form and it can be accessed following the link "**Spinet:**" on the registry or using something similar to http://{host}/soaplab2-axis/ . For example, the "hunalign" web service can be found on the registry. Its wsdl is: http://www.cngl.ie/panacea-soaplab2-axis/services/panacea.hunalign?wsdl. The URL of the Spinet of that web service provider is http://www.cngl.ie/panacea-soaplab2-axis.

Alert! Keep in mind that web services' results can differ depending on the parameter settings.

2.1.3. Design and run workflows.

Once a web service has been tested and fulfills the desired expectations of the user, it can be combined to other web services in a workflow. To find the optimum and desired results of a web service we recommend testing it before integrating it to a workflow. Spinet web client is a simple and easy way to test parameters.

"Taverna" is the tool used in PANACEA to create and run workflows. It can be downloaded from http://www.taverna.org.uk/ where manuals and tutorials can be found.

Tips and interesting links: see **5.2 Taverna FAQs** below.

2.1.1. Find and share already designed workflows

Users may want to share their web services or find already designed ones. PANACEA has a social network portal to share, find and discuss workflows: http://myexperiment.elda.org .
This portal is also used to share files with source code, data, etc.

2.2. PANACEA Service Providers

PANACEA Service Providers are researchers, institutions, companies, etc. who want to make their tools available for everyone.

2.2.1. Deploy web services

PANACEA has chosen Soaplab to be the software to deploy web services. It can be used to deploy command line applications as web services.

Other technologies like Apache Axis or apache CXF[1] can also be used in PANACEA. However, all deployed web services should be compliant with the Common Interface (CI) design (if there is a CI for that particular web service). All kind of web services can be registered on the registry (http://registry.elda.org).

Tips and links: see **5.1 Soaplab FAQs** below.

2.2.2. Register and annotate web services.

Service Providers must be registered to the registry to operate (http://registry.elda.org). All web services should be registered and annotated (categorized and tagged) so users can find them.

[1] http://axis.apache.org/axis2/java/core/ and http://cxf.apache.org/

The registry has an automatic status check system for web services. It shows whether the web service is up and running or not.

Tips and links: see **5.3 PANACEA registry (Biocatalogue instance) FAQs** below.

3. Travelling Object

Interoperability between web services is granted by using a common data format. This format is called “Travelling object” (TO) and is defined in D.3.1 Section 6.1. *Travelling object. Corpus and data format.*

Using the TO is not always necessary but in some cases it is the only way to link processes. In order to convert from one format to another, users can use converters. There are several format converters deployed which can be found on the registry.

The converter from / to some tools in-house formats into TO is the “panacea_conversor” (http://registry.elda.org/services/115). Most workflows in http://myexperiment.elda.org make use of this web service and can be used as an example.

4. Common interfaces

Common interfaces (CI) are used to make web services easily interchangeable. All web services share the same interface at some level. In PANACEA, this means they have the same mandatory parameters. The Common Interface documentation can be found in http://www.panacea-lr.eu/en/info-for-professionals/documents/.

All deployed web services should be CI compliant if their functionality has a definition on the CI.

5. FAQs

These FAQs can be found also in the PANACEA website:

http://panacea-lr.eu/en/info-for-professionals/faqs/

5.1. Soaplab FAQs

5.1.1. General

- I have a command line application. How can I make it a web service?
There are different solutions. We use Soaplab in PANACEA.

- **What is Soaplab?**
It is a software tool to automatically turn command line applications into web services.

- Where can I get Soaplab?
Soaplab software and documentation can be found in http://soaplab.sourceforge.net/soaplab2/

- Which Soaplab version is PANACEA using?
The latest! soaplab2-2.3.2.

- Which web application container / server do I need to use to deploy Soaplab web services?
Apache Tomcat. In PANACEA we use Tomcat 6. http://tomcat.apache.org

5.1.2. Installation / ACD files development

- When I use *ant* commands I get timeouts when connecting to other sites?
Probably you are behind a **proxy**. Use **ant -autoproxy** instead of just **ant** in all your commands.

- How can I create my ACD files?
Follow the instructions in http://soaplab.sourceforge.net/soaplab2/MetadataGuide.html

- Where do I put my ACD files that describe my command line applications?
In the **src/etc/acd/sowa** folder in your soaplab2 installation folder.

- What kind of input parameter should I use for large data?
In PANACEA, we use **infile** parameters with "**comment: default_for_direct**". This way the input data can be provided with a **URL** (link) or direct data (a stream of characters). Direct data in Spinet web client can be provided directly typing in a text box or by uploading a file with "Browse..." button.

5.1.3. Deployment and configuration

- How can I create the necessary XML metadata files from ACD files?
Use *ant gen* to create metadata from ALL ACD files. In PANACEA we use **ant gensowa** to generate XML from files in the SOWA directory (src/etc/acd/sowa)**.**

- Which protocol should I use: *axis1* or *jaxws*?
In PANACEA, we use axis1 because the Taverna workflow editor is not 100% compatible with jaxws.

- How can I deploy web services?
Use **ant jaxdeploy** or **ant axis1deploy**. In PANACEA, we use ant axis1deploy because axis1 is compatible with Taverna.

- Should I use typed interface?
Yes. More applications will be able to call your web services.
http://soaplab.sourceforge.net/soaplab2/TypedServices.html

- How can I configure Soaplab to create web services with typed interface?
Set **typedinterface.enable = true** in your build.properties file.

- How can I create the WAR file without automatically deploying it?
Use **ant jaxwar** or **ant axis1war** instead of ant jaxdeploy or ant axis1deploy.
Running the Soaplab services

- How can I test my Soaplab web services?
An easy way to test them is by using the Spinet web client: http://localhost:8080/soaplab2/
or http://{your_host}:{your_port}/soaplab2

- How can I run Soaplab web services? Which web services clients can I use?
You can use **Taverna** workflow editor (http://www.taverna.org.uk/), **Spinet** web client or **perl** programs such as: http://www.ebi.ac.uk/Tools/webservices/soaplab/guide. **More** information can be found in http://soaplab.sourceforge.net/soaplab2/ClientsOverview.html.

- What is *detailed_status* output?
It indicates the exit status of your command line program. If a program ends correctly its exit status is 0 otherwise its value is 1. Soaplab captures this value and uses the detailed_status to show it. If exit status is 1, Soaplab will end with an error message.

5.2. Taverna FAQs

5.2.1. General

- Which taverna version do PANACEA members use?
Taverna 2.4

- **Where can I find documentation about Taverna workbench, local processors, etc.?**
http://www.mygrid.org.uk/dev/wiki/display/taverna/Documentation+and+Videos

5.2.2. Designing workflows

- **How can I import Soaplab web services from a service provider?**
Use the Soaplab option after “Import new services” button. If the Soaplab server you want to use is: http://{host}/soaplab2-axis, the address for Taverna is **http://{host}/soaplab2-axis/services/**. Do not forget the final slash (/).

- **I am trying to add a Soaplab server and I get an error message?**
Be sure you add the final slash to the address! **http://{host}/soaplab2-axis/services/**.
Verify the Spinet web client is working.

- **What should I use to connect the inputs and outputs from different Soaplab processors: *output or output_url*?**
In PANACEA, we recommend **output_url** to deal with big files.

- **Can I inspect PANACEA catalogue from Taverna?**
Yes! Go to your Taverna *Preferences* and configure the Biocatalogue instance that you want to use.

- **How can I get specific information from an xml data file? Is there a local processor to do that?**
There are two options:

1. Use the **“Xpath from text” local processor** which will require a little xpath knowledge.
http://www.mygrid.org.uk/dev/wiki/display/taverna/XPath+From+Text
2. The **“Xpath service”** from service templates. Really easy to use and no xpath knowledge required. 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

- **I am working with some processors whose outputs are lists. This creates a problem of undesired nested lists. How can I solve this?**
You can use the “**flatten list**” local processor to reduce the list one level.
http://www.mygrid.org.uk/dev/wiki/display/taverna/Flatten+list

- **I am using nested workflows with list inputs. The output is creating several nested lists with a lot of repeated outputs! What can I do?**
The outputs you are seeing are the result of processing all possible combinations of inputs (cross product). You can get more information here:
http://www.mygrid.org.uk/dev/wiki/display/taverna/List+handling
If you want to pair up items from input lists you can follow these steps:
http://www.mygrid.org.uk/dev/wiki/display/taverna/List+handling#Listhandling-Pairingupvalueswithdotproduct

- **I am using a web service with several input lists and parameters and I get validation errors. I want to pair list elements and always use the same parameters.**
The solution is to combine cross product and dot product with list handling:
Single parameters that use cross product and a sub dot product must be created to combine the list parameters.
http://www.mygrid.org.uk/dev/wiki/display/taverna/List+handling#Listhandling-Combinedcrossproductanddotproduct

-I run a slow process or a big file without problems in Spinet but I get a timeout error in Taverna. What can I do?

Taverna has a timeout limit. If an answer from the web service takes too long Taverna thinks something went wrong. To avoid this you must use “polling”. Follow these instructions:
http://www.mygrid.org.uk/usermanual1.7/soaplab_processor.html#soaplab_polling

-When there is an error in one web service the data is lost. What can I do?
You should use “retries” to execute one web service more times in the case that it fails.
http://www.youtube.com/watch?v=-jWQH9_u370

-Is there any way to execute a task in parallel?
Yes! You can configure web services to be executed in parallel. You must be careful since the machines have limited resources. Follow the examples and read the following document:
http://taverna.knowledgeblog.org/2010/12/13/parallel-service-invocations/

5.2.3. Running workflows

- How can I improve Taverna’s performance (not web services performance!)?
Deactivate “**in-memory storage**” and “Enable provenance capture” option in “**data and provenance**” located in Taverna Preferences.
You can also assign more memory to Taverna by following these instructions: http://www.mygrid.org.uk/dev/wiki/display/taverna/Increase+memory+allocation. We are using the following configuration: *exec java –Xmx800m -XX:MaxPermSize=280m*.

- Can I inspect intermediate results?
Yes, if you have that option activated in “**data and provenance**” located in Taverna Preferences. To inspect the results click on the processor you are interested in within the results view.

- How can I count the execution time?
In the results view click on the “**Progress report**” tab and you will find detailed information about each processor / web service: iterations, **Average time / iteration**, etc.

5.3. PANACEA registry (Biocatalogue instance) FAQs

5.3.1. Registering web services

- How should I register my Soaplab2 web services: SOAP(wsdl) or soaplab server?
You should use the **Soaplab server**. The registry will automatically gather all of your web services and register them.

- Is it really necessary to categorize all my web services?
No, but it is recommended. Categorization facilitates the users find services.

- Is it important to use tags?
Yes. Tags are a very versatile and useful tool to annotate web services. With tags, users can find web services without being experts. For example, there is no need to know the category.

5.3.2. Searching web services

- How can I search for web services?
The Biocatalogue instance has tags, categories, providers, and more parameters to search web services.
The “Search” field on the web site can be used with wildcards to make for more flexible queries i.e. asterisk (*) or question mark (?). For example **crawl*** would match **crawler** and **crawling**.

- Is it possible to know if a service is up and running?
Yes! Next to every service there is a “ball” indicating if the service is available. Move your mouse over it to get extra information such as “last checked”.