Embedding workflows in your website

Aleksandra Pawlik
myGrid Team
University of Manchester

VLIZ, 2014-10-06 / 2014-10-08
http://www.taverna.org.uk/
Seen it already

• You have already seen an example of a website that runs Taverna workflows
  • the BioVeL portal

• How was it done?
  Taverna Player
  • Looks after the running of the workflows
  • Surfaces them for a website
Lightweight embedding

• Embed like a YouTube video
• Embedded workflow is passed the URI of data

<iframe src="http://portal.org/runs/new?
  embedded=true&
  workflow_id=1&
  input_uri=http://scratchpad.org/taxa/1234/data"
>
</iframe>

• This level of integration is lightweight
  • Science showcases
  • One off analyses

https://github.com/myGrid/taverna-player/wiki/Embedding
Tightness of integration

**Lightweight embedding**
- Run a specified workflow
  - Chosen by the host website’s administrator
- Results are not stored in the host
- Workflow run retains host app look and feel

**Tight integration**
- Run any workflow
  - That the host website is authorized to see
- Results are available for further analysis
- Workflow appears as part of the host website

**Common**
- Workflows are run within Taverna Player in the host app
- Interactions are presented to the user
- Results can be downloaded
Taverna Player

• A Ruby on Rails plugin library
  • Hooks into host application’s
    • Workflow model
    • Authentication and authorization system
  • Provides a REST interface

• Talks to Taverna Server’s REST interface
  • Uploads the workflow, sets inputs
  • Presents workflow interactions to the user
  • Retrieves results, logs and provenance data
Taverna Player

• Surfaces a workflow run in three ways:
  • As a Web interface in the browser
    • In the host application
  • As an embeddable widget
    • In any Web page (c.f. YouTube videos)
  • As a REST-based Web Service

• All look-and-feel and styling is derived from the host application
  • Rails’s hierarchical layouts and views
Taverna Player

• Total workflow run isolation
  • A worker per run
  • State passed via database

• Scaling
Taverna all together
Workflows in Scratchpads

• Virtual Research Environments
• Hosted websites for biodiversity data
• Virtual research & publication platform
• Curated data and analysis
• Completely open access & open source
• Modular & flexible
Workflows in Scratchpads - cont
IPython Notebook

- IPython Notebook
  - originally developed by Fernando Perez of University of Berkeley
  - browser-based environment for interactive computing
  - [http://ipython.org/](http://ipython.org/)
- write, edit and re-run Python scripts
- interactive data visualization
- report presentation
- save, record, share notebook runs
Taverna Player Client

- Uses Taverna Player and its Server to run workflows within an IPython Notebook
- Data passed from the Notebook to the executing Taverna Workflows
- Workflow run’s requests for data answered within the Notebook using Taverna’s interaction service
- Results retrieved from the run and fed back into the Notebook

- Tested with BioVeL workflows for data refinement and ecological niche modelling.
Taverna in IPython example

Client Creation

```python
In [3]: from tavernaplayerclient import *
In [4]: c = Client('http://dev.at.biovel.eu', 'player', 'player')
In [5]: workflows = c.workflows
In [6]: for w in c.workflows: print w.identifier, w.title
```

Interaction with Run

```
In [20]: esw = c.get_workflow(20)
In [21]: esw_run_template = esw.run_template
In [22]: print esw.description
```

Organization of Data

```
In [23]: namelist = "now,2050"
In [24]: layerlist = projection_urls[0] + "," + projection_urls[1]
In [25]: esw.run("Example esw run",
                   "namelist": namelist,
                   "layerlist":layerlist,
                   "png_size": "800")
```
Taverna and IPython architecture

**Flows**

<table>
<thead>
<tr>
<th>Flow</th>
<th>Notebook -&gt; Player -&gt; Portal</th>
<th>Notebook -&gt; Player -&gt; Notebook</th>
<th>Notebook -&gt; Player -&gt; Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorize Client</td>
<td>Start run</td>
<td>Interact with run</td>
<td>Return results</td>
</tr>
<tr>
<td>List workflows</td>
<td>Notebook -&gt; Player -&gt; Portal</td>
<td>Notebook -&gt; Player -&gt; Notebook</td>
<td>Server -&gt; Player -&gt; Notebook</td>
</tr>
<tr>
<td>Get workflow and run template</td>
<td>Notebook -&gt; Player -&gt; Portal</td>
<td>Notebook -&gt; Player -&gt; Notebook</td>
<td>Server -&gt; Player -&gt; Notebook</td>
</tr>
</tbody>
</table>
Summary

• Taverna Player is very flexible
• Examples of integration into
  • Ruby on Rails – BioVeL Portal
  • Drupal – Scratchpads
  • Python – IPython Notebook
• Different levels of integration
  • Simple iframe to
  • Communication via REST API
• More information at https://github.com/myGrid/taverna-player/wiki