Taverna Components
Semantically annotated and sharable units of functionality

Alan Williams, Donal Fellows, Finn Bacall,
Stian Soiland-Reyes, Khalid Belhajjame,
David Withers, Carole Goble

School of Computer Science, University of Manchester, UK
http://www.taverna.org.uk/
What is a Component?

- Something that can be put into a workflow
  - Well described — what the component does
  - Behaves “well” — conforms to agreed good practice policy
  - Curated — someone looks after it
  - Produces and consumes data in agreed formats
  - Fails in described ways — meaningful error messages
  - Produces agreed type of provenance
- Documentation
- Example usage
Usefulness of Components

- Predictable good behaviour
  - Should conform to a defined agreement
- Hide complexity
- Guaranteed to work together
- Can (in theory) check that data in a run conforms to the component specification
What is the Agreement?

- The agreement is a condition of being in a “component family”
- Different domains, or even different uses within a domain, have different agreements
  - Astronomical data does not use the same formats as biodiversity data
  - Digital library components do not do the same tasks as physiology components
- Agreement is formalized as a “component profile”
Support in Taverna

- Prototype Plugin for Taverna 2.4
- Full part of Taverna 2.5 Workbench
  - One of the key features
- Will be supported in Taverna 2.5 Server
  - Requires components to be published to a repository
    - Can be done with restricted access permissions
Component Architecture
Implementation

- A component family is:
  - a pack on myExperiment, or
  - a directory on your local machine
- A component is defined by a workflow (in a pack) in a component family pack
- Components are versioned by myExperiment’s versioning
- Semantic Annotations are stored in RDF as part of the workflow definition
- Collated semantics, including workflow structure, are combined on myExperiment
myExperiment Component Pack

- Definition, description, documentation
- Contains:
  - Workflow “realizing” the component
  - Example data
  - Documentation
  - Dependency specification
  - ...

Component Pack

- Realizing Workflow
- Sample Data
- Usage Manual

BOSC 2013, Berlin
Finding Components

Pack: SCAPE Image Characterisation Component

Created at: 04/04/13 @ 12:39:14  Last updated: 20/05/13 @ 12:07:25

Title: SCAPE Image Characterisation Component

These components handle extraction of characteristics from an image.

Items (5)

- Pack: Extract Image Dimensions with SIPS (Donal Fellows)
- Pack: Extract PNG dimensions (Donal Fellows)
- Pack: Extract TIFF dimensions (Donal Fellows)
- Pack: Extract JPEG-2000 dimensions (Donal Fellows)
- File: Characterisation Component (Donal Fellows)

Tags from Items (3)

- component
- component profile
- SCAPE

Shared with Groups (1)

- SCAPE

Featured in Packs (0)
Using Components

- Component Families shown in the service panel of Taverna Workbench
- Components can be included within a Taverna workflow
  - Like any service
- Components are not simply the same as nested workflows
  - Think of them as nested workflows that:
    - Obey a set of rules
    - Hide what is nested inside (and you should not care)
Component Creation

- Create Components by Annotating a Workflow
  - Choice of a component family and so profile
  - Semantic annotation from the specified ontologies
  - Validation against the profile
  - Component saved into the component family

- Can Annotate:
  - Workflow itself
  - Workflow input/output ports
  - Individual services inside workflow

- Extensions to myExperiment for
  - Pack snapshots
  - Semantic collation
  - Semantic searching
Semantic Annotation

Workflow explorer  Details  Validation report

Component Extract JPEG-2000 dimensions
Workflow Extract_JPEG_2000_im
Annotations

Semantic Annotations

Annotation type: handlesMimetype
image/jp2  Change  Delete

Annotation type: fits
Characterisation  Change  Delete

Add/change annotation
Enter a value for the annotation 'fits'

Turtle annotations

```xml
<http://purl.org/DP/components#fits>
  <http://purl.org/DP/components#Characterisation> ;
<http://purl.org/DP/components#handlesMimetype>
  "image/jp2"^^<http://www.w3.org/2001/XMLSchema#string> .
```
Semantic Annotation — Aside

- Semantic annotation will not be limited to components
- Will be a general feature of Taverna workflows and workflow runs
Effect on Workflows

- Use of components will allow
  - Component developers to work on the component
  - Component users to upgrade (or revert) the component versions
  - A workflow to remain ‘unchanged’ (if the component interfaces remain the same)
    - Powerful and dangerous!
  - Proxies for components (re-run and re-play)
- Components are “black boxes” in the workflow and workflow runs
Leveraging Semantics — RDF Description

```rdfs
@base <http://ns.taverna.org.uk/2010/workflowBundle/8d2f9ef0-09ca-4103-b4fd-0ee0a40d8263/workflow/Imagemagick_convert/> .
@prefix wfdesc: <http://purl.org/wf4ever/wfdesc#> .
@prefix wf4ever: <http://purl.org/wf4ever/wf4ever#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix scape: <http://purl.org/DP/components#> .

<>
scape:fits scape:MigrationAction ;
scape:migrates [   
scape:fromMimetype "image/tiff" ;
scape:toMimetype "image/tiff" ;
a scape:MigrationPath ] ;
wfdesc:hasDataLink <datalink?from=in/compression&to=processor/convert/in/compression>,  
    <datalink?from=in/from_uri&to=processor/convert/in/from_uri>,  
    <datalink?from=in/to_uri&to=processor/convert/in/to_uri>,  
    <datalink?from=processor/convert/out/STDERR&to=out/status&mergePosition=1>,  
    <datalink?from=processor/convert/out/STDOUT&to=out/status&mergePosition=0> ;
wfdesc:hasInput <in/compression>, <in/from_uri>, <in/to_uri> ;
wfdesc:hasOutput <out/status> ;
wfdesc:hasSubProcess <processor/convert/> ;
a wfdesc:Description, wfdesc:Process, wfdesc:Workflow ;
rdfs:label "Imagemagick_convert_" .

<datalink?from=in/compression&to=processor/convert/in/compression>
  wfdesc:hasSink <processor/convert/in/compression> ;
  wfdesc:hasSource <in/compression> ;
a wfdesc:DataLink .
```

BOSC 2013, Berlin
Leveraging Semantics — Searching
Future Work

- Treat current workflow as an RDF model
  - Enable semantic searching
- User-friendly searching and compound object creation
- Creation of workflows from templates
  - Replacing template objects with dynamically-found components
- Collation of provenance from components
- Enable third-party component repositories
Acknowledgements

- This work has been supported by the EU 7th Framework Programme, via:
  - The BioVeL Project, contract 283359
    - http://www.biovel.eu/
  - The SCAPE Project, contract 270137
    - http://www.scape-project.eu/
  - The WF4Ever Project, contract 270192
    - http://www.wf4ever-project.org/

Taverna Workflow System
http://www.taverna.org.uk/

BOSC 2013, Berlin