

Project site:
<http://www.taverna.org.uk/>

Source code:
<https://github.com/mygrid/scufl2>
<http://taverna.googlecode.com/>

License: GNU Lesser General Public License (LGPL) 2.1



Root file: Primary document
the bundle (ePub OCF/Adobe
UCF), Alternative representations
of same workflow bundle allowed
(Turtle, JSON, HTML, etc)

Unique identifier which can
be used as prefix for all
relative references in bundle

Suggests main workflow and
main profile, but executor
could (e.g. by parameter) run a
different workflow or profile

Additional (non-executable)
annotations and metadata

Profile gives implementation
bindings, alternative profiles
can customize execution of
workflow steps for different
environments (e.g. desktop,
server, cloud)

Hybrid of RDF/XML and XML
schema - If the xsi:type is
given, documents can be
parsed and generated as
regular XML, using xpath, etc.

The schema ensures the
document can still be parsed as
RDF/XML as well. Pure RDF
writers can omit the xsi:type

Scufl2 – because a workflow is more than its definition



Stian Soiland-Reyes, Alan R Williams, Stuart Owen, David Withers and Carole Goble

School of Computer Science, University of Manchester, UK

{stian.soland-reyes, alan.r.williams, stuart.owen, david.withers, carole.a.goble}@manchester.ac.uk

- Taverna's future workflow and data format
- Goal: Simplify third-party reading, writing, annotating and extending
- Scufl2: specification, schema, ontology, Java API and conversion tool

"Structured" zip-file
(can be unpacked to be
exposed on the web)

the-workflow-bundle.scufl2

mimetype
application/vnd.taverna.scufl2.workflow-bundle
META-INF/container.xml

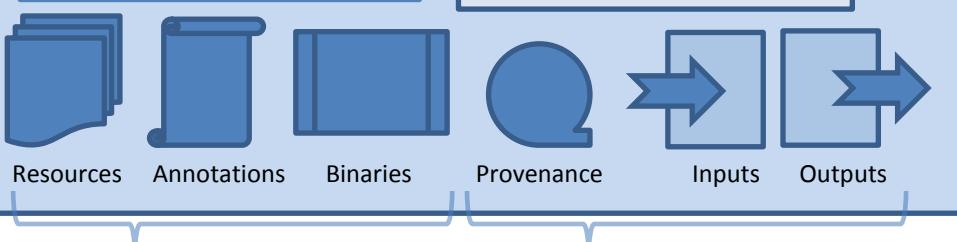
<manifest version="1.0" xmlns="urn:oasis:names:tc:opendocument:xmlns:container">
<rootfiles>
 <rootfile full-path="workflowBundle.rdf" media-type="application/rdf+xml" />
 <rootfile full-path="WorkflowBundle.ttl"
 media-type="text/turtle" /> Alternative repr. -->
</rootfiles>
</manifest>

workflowBundle.rdf

<rdf:RDF xmlns="http://ns.taverna.org.uk/2010/scufl2#">
 <xsi:type>WorkflowBundle</xsi:type>
 <owl:versionInfo>2010-08-17.08:57:31Z</owl:versionInfo>
 <owl:imports>HelloWorld</owl:imports>
 <owl:imports>WorkflowServer</owl:imports>
 <owl:imports>Workflow</owl:imports>
 <owl:imports>Workflow</owl:imports> <!-- plus each nested workflow -->
 <owl:profile rdf:resource="profile/tavernaworkbench/" />
 <profiles>
 <profile rdf:about="profile/tavernaworkbench/" />
 <profile rdf:resource="profile/tavernaworkbench.rdf" />
 </profiles>
 <annotation rdf:resource="annotation/user_annotation.rdf" />
 <rdf:seeAlso rdf:resource="annotation/myExperiment-wf-765.rdf" />
</WorkflowBundle>

profile/tavernaworkbench.rdf

<rdf:RDF xmlns="http://ns.taverna.org.uk/2010/scufl2#">
 <xsi:type>HelloWorld</xsi:type>
 <owl:versionInfo>2010-08-17.08:57:31Z</owl:versionInfo>
 <owl:imports>HelloWorld</owl:imports>
 <owl:imports>WorkflowServer</owl:imports>
 <owl:imports>Workflow</owl:imports>
 <owl:imports>Workflow</owl:imports> <!-- plus each nested workflow -->
 <owl:profile rdf:about="processor/HelloWorld/" />
 <processors>
 <processor rdf:resource="processor/greeting/HelloWorld/" />
 <activateConfiguration rdf:resource="configuration/HelloWorld/" />
 <activity rdf:about="activity/HelloWorld/" />
 <execute rdf:resource="execute/HelloWorld/" />
 <nameHelloScript />
 <inputActivityPort />
 <inputActivityPort />
 <outputActivityPort />
 <outputActivityPort />
 </processors>
 <processes>
 <process rdf:about="processor/greeting/HelloWorld/" />
 <nameHello />
 <rdf:resource="activity/HelloWorld/" />
 <bindProcessor rdf:resource=".../HelloWorld/HelloWorld/processor/HelloWorld/" />
 <bindProcessor rdf:resource=".../HelloWorld/HelloWorld/processor/HelloWorldConfig/" />
 <bindProcessor rdf:resource=".../HelloWorld/HelloWorldConfig/" />
 <optionPath rdf:resource="HelloWorldConfig/HelloWorld/HelloWorldConfig" />
 </processes>
 <configuration rdf:about="configuration/HelloWorld/" />
 <rdf:resource="http://ns.taverna.org.uk/2010/activity/beanShellConfig/" />
 <nameHello />
 <activity rdf:about="activity/HelloWorldScript" />
 <beanShellScript>HelloWorldHelloWorld />
 <optionPath rdf:resource="HelloWorldConfig/HelloWorld/HelloWorldConfig" />
 <configuration rdf:about="configuration/HelloWorldConfig/" />



The bundle (as a ZIP file) can be extended to include arbitrary resources. These could be referenced internally by workflow activities, it could be PDFs, spreadsheets, etc.

Wf4Ever: A Research Object (RO) captures enough data and provenance about results to make data and methods reproducible, verifiable, shareable, reusable and repeatable. The Scufl2 workflow run bundle forms such an RO for Taverna workflow results – with the goal of becoming an executable paper.

Embedding provenance, input and output data of a workflow run can provide example use and expected outputs. Changing the root-file to point to the provenance makes the bundle a workflow run bundle, where the workflow definition is secondary.

Similarly a data bundle represents primarily the data, but by including the workflow and the provenance also embed information on how it was generated.

Manifest listing all
resources in
bundle with media
types

META-INF/manifest.xml

```
<manifest:manifest  
  xmlns:manifest="urn:oasis:names:tc:opendocument:xmlns:manifest:1.0"  
  >  
  <manifest:file-entry manifest:media-type="application/vnd.taverna.scufl2.workflow-bundle"  
    manifest:full-path="/"/>  
  <manifest:file-entry manifest:media-type="application/rdf+xml"  
    manifest:full-path="WorkflowBundle.rdf"/>  
  <manifest:file-entry manifest:media-type="application/rdf+xml"  
    manifest:full-path="WorkflowHelloWorld.rdf"/>  
  <manifest:file-entry manifest:media-type="application/rdf+xml"  
    manifest:full-path="WorkflowServer.rdf"/>  
  <manifest:file-entry manifest:media-type="application/rdf+xml"  
    manifest:full-path="annotation/user_annotation.rdf"/>  
  <manifest:file-entry manifest:media-type="application/rdf+xml"  
    manifest:full-path="annotation/myExperiment-wf-765.rdf"/>  
  <manifest:file-entry manifest:media-type="image/png" manifest:full-path="diagram/workflow/HelloWorld.svg"/>  
</manifest:manifest>
```

workflow/HelloWorld.rdf

```
<rdf:RDF xmlns="http://ns.taverna.org.uk/2010/scufl2#">  
  <xsi:type>HelloWorld</xsi:type>  
  <owl:versionInfo>2010-08-17.08:57:31Z</owl:versionInfo>  
  <owl:imports>HelloWorld</owl:imports>  
  <owl:imports>WorkflowServer</owl:imports>  
  <owl:imports>Workflow</owl:imports>  
  <owl:imports>Workflow</owl:imports> <!-- plus each nested workflow -->  
  <owl:profile rdf:about="processor/HelloWorld/" />  
  <processors>  
    <processor rdf:about="processor/greeting/HelloWorld/" />  
    <activateConfiguration rdf:resource="configuration/HelloWorld/" />  
    <activity rdf:about="activity/HelloWorld/" />  
    <execute rdf:resource="execute/HelloWorld/" />  
    <nameHelloScript />  
    <inputActivityPort />  
    <inputActivityPort />  
    <outputActivityPort />  
    <outputActivityPort />  
  </processors>  
  <processes>  
    <process rdf:about="processor/greeting/HelloWorld/" />  
    <nameHello />  
    <rdf:resource="activity/HelloWorld/" />  
    <bindProcessor rdf:resource=".../HelloWorld/HelloWorld/processor/HelloWorld/" />  
    <bindProcessor rdf:resource=".../HelloWorld/HelloWorld/processor/HelloWorldConfig/" />  
    <bindProcessor rdf:resource=".../HelloWorld/HelloWorldConfig/" />  
    <optionPath rdf:resource="HelloWorldConfig/HelloWorld/HelloWorldConfig" />  
  </processes>  
  <configuration rdf:about="configuration/HelloWorld/" />  
  <rdf:resource="http://ns.taverna.org.uk/2010/activity/beanShellConfig/" />  
  <nameHello />  
  <activity rdf:about="activity/HelloWorldScript" />  
  <beanShellScript>HelloWorldHelloWorld />  
  <optionPath rdf:resource="HelloWorldConfig/HelloWorld/HelloWorldConfig" />  
  <configuration rdf:about="configuration/HelloWorldConfig/" />
```

Definition of
workflow structure.
Nested workflows
are separate
resources.

Every workflow part
has a URI, allowing
deep annotations

Relative references can also
be made absolute using the
sameBaseAs prefix, RDF clients
resolving such URIs at
<http://ns.taverna.org.uk/2010/workflowBundle/> can be
redirected to a generated RDF
resource stating what's
obvious from the URL.

"Cool URI"-style relative
references allow parsers to
'cheat' and pick out processor
"Hello" and port "greeting" -
but only if starts with the
keyword paths out/ in/
or processor/