Tracking workflow execution with TavernaProv

Document id: https://github.com/stain/2016-provweek-tavernaprov/
Stian Soiland-Reyes, Pinar Alper, Carole Goble; eScience Lab, University of Manchester

Apache Taverna is a scientific workflow system for combining web services and local tools. Taverna records provenance of workflow runs, intermediate values and user interactions, both as an aid for debugging while designing the workflow, but also as a record for later reproducibility and comparison.

Taverna also records provenance of the evolution of the workflow definition (including a chain of `wasDerivedFrom` relations), attributions and annotations; for brevity we here focus on how Taverna's workflow run provenance extends PROV and is embedded with Research Objects.

Data bundle

A workflow run can be exported from Taverna as a workflow data bundle; a Research Object bundle in the form of a ZIP archive that contains the workflow definition (itself a Research Object), annotations, inputs, outputs and intermediate values, and a PROV-O trace of the workflow execution, showing every process execution within the workflow run, linking to the produced and consumed values using relative paths.

A workflow run can thus be downloaded as a single file from a Taverna Server, shared (myExperiment, SEEK), published (ROHub, Zenodo), imported in another Taverna Workbench, shown in the Databundle Viewer or modified with the DataBundle API.

Abstraction levels

PROV is a generic model for describing provenance. While this means there are generally many multiple ways to express the same history in PROV, a scientific workflow run with processors and data values naturally match the `Activity` and `Entity` relations `wasGeneratedBy` and `used`.

However, using PROV-O to describe the details of a Taverna execution meant a significant increase in verbosity. To simplify query and interoperability with PROV tools, we declare relations both with starting point terms and as qualified terms, e.g. to represent an `Activity` that used different values as different input parameters, we provide both a direct used, but also a qualifiedUsage to a Usage that specify hadRole and entity.

PROV deliberately does not mandate how to make the design decisions on what activities, entities and agents participate in a particular scenario, but for interoperability purposes this flexibility means that PROV is a kind of "XML for provenance" - a common language with a defined semantics, but which can be applied in many different ways.

One interoperability design question for representing computational workflow runs is how much of the workflow engine's internal logic and language should be explicit in the PROV representation. As we primarily wanted to convey what happened in the workflow at the same granularity as its definition, we tried to hide provenance that would be intrinsic to the Taverna Engine, e.g. implicit iteration is not shown as a separate `Activity`.

However keeping provenance only at the dataflow level (input/outputs of workflow processes) meant that TavernaProv could not easily represent "deeper" provenance such as the intermediate values of while-loops or intermittent failures that were automatically recovered by Taverna's retry mechanism, as we wanted to avoid unrolled workflow provenance.

Keeping the link between the workflow definition and execution is essential to understanding Taverna provenance, yet PROV doesn't describe the structure of a Plan. Taverna's workflow definition is in Linked Data using the SCUFL2 vocabulary, which includes many implementation details for the Taverna Engine, and so forming a meaningful query like *"What is the value made by calls to webservice X"* means understanding the whole conceptual model of Taverna workflow definitions.

Therefore Taverna's PROV export also includes an annotation with the
wfdesc abstraction of the workflow definition, embedding user annotations and higher-level information like web service location.

wfdesc deliberately leaves out execution details like iteration and parallelism controls as it primarily functions as a target for user-driven annotations about the workflow steps.

Correspondingly the provenance bundle from TavernaPROV includes a higher level wfprov abstraction of the workflow execution, with direct shortcuts like describedByProcess and describedByParameter to bypass the indirection of PROV qualified terms; simplifying queries like *"Which web service consumed value Y?"*.

The duality between wfdesc and wfprov is similar to the "future provenance" model of P-Plan and OPMW and its workflow templates , and similarly the the split between "prospective provenance" and "retrospective provenance" of the ProvONE Data Model for Scientific Workflow Provenance.

Identifiers and interoperability

A great advantage of using Linked Data was that we could use the same identifiers in all three formats. One challenge was that Taverna workflows are often run within a desktop user interface or on the command line, and with privacy concerns we didn't have the luxury of a server to mint URIs; we already learnt our lesson with LSIDs.

Taverna therefore generate UUID-based structured `http://` URIs within our namespaces, e.g. `http://ns.taverna.org.uk/2011/run/d5ee659e-e11e-43a5-bc0a-58d93674e5e2/process/1e027057-2aeb-47f7-97dc-03e19e9772be/` and `http://ns.taverna.org.uk/2010/workflowBundle/2f0e94ef-b5c4-455d-aeab-1e9611f46b8b/workflow/HelloWorld/processor/hello/`

The scufl2-info web-service provide a minimal JSON-LD wfprov/wfdesc representation identifying the URI as a provenance or workflow item, but (by design) not having access to the data bundle it can't say anything more.

We found that our UUID-based URIs don't play too well with PROV Toolbox and alternative PROV formats like PROV-N and PROV-XML, as every URI ending in `/` is registered as a separate namespace in order to form valid QNames. A suggested improvement for TavernaProv is to generate provly identifiers, while remaining compliant with the 10 simple rules for identifiers.

Similarly, OWL reasoning is not generally applied by PROV-O consumers, so even though *wfprov* formally extends PROV in its ontology definitions, we needed to add explicitly the implied PROV-O statements in TavernaProv's Turtle output.

Common Workflow Language

Common Workflow Language has created a workflow language specification, a reference implementation cwltool, and a large community of workflow system developers who are adding CWL support across bioinformatics, including Apache Taverna and Galaxy. Unlike wfdesc, OPMW and P-Plan, CWL workflows are primarily intended to be executed, with a strong emphasis on the dataflow between command line tools packaged as Docker images.

CWL is specified using Schema Salad, which provides JSON-LD constructs in YAML. The CWL dataflow model is inspired by wfdesc and Apache Taverna and thus have similar execution semantics and provenance requirements. CWL is planning its provenance format based on PROV-O, wfprov and JSON-LD. As one of the CWL adopters, Apache Taverna will naturally also aim to support the CWL provenance model.

Future work

To face the verbosity issue, we are considering to split out wfprov statements to a different file; as a ZIP archive the Taverna Data Bundle can contain many provenance formats. Similarly splitting out the details using PROV-O Qualified Terms to a separate file is worth considering, this could also improve PROV visualization of workflow provenance. Having such separate PROV bundles would also make it easier for Taverna to support the ProvONE model as an additional format.

PROV Links could be added to Research Object Bundles to relate its data files to the then multiple workflow provenance traces that describe their generation and usage.

References

1. Katherine Wolstencroft, Robert Haines, Donal Fellows, Alan Williams, David Withers, Stuart Owen, Stian Soiland-Reyes, Ian Dunlop, Aleksandra Nenadic, Paul Fisher, Jiten Bhagat, Khalid Belhajjame, Finn Bacall, Alex Hardisty, Abraham Nieva de la Hidalga, Maria P. Balcazar Vargas, Shoaib Sufi, Carole Goble (2013): **The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the**

cloud. In: *Nucleic Acids Research*, **41**(W1): W557–W561. doi:10.1093/nar/gkt328

2. Paolo Missier, Satya Sahoo, Jun Zhao, Carole Goble, Amit Sheth. (2010): **Janus: from Workflows to Semantic Provenance and Linked Open Data** in *Provenance and Annotation of Data and Processes, Third International Provenance and Annotation Workshop, (IPAW'10)*, 15–16 Jun 2010. Springer, Berlin: 129–141. doi:10.1007/978-3-642-17819-1_16 [pdf]
3. Stian Soiland-Reyes, Matthew Gamble, Robert Haines (2014): **Research Object Bundle 1.0**. *researchobject.org Specification*. https://w3id.org/bundle/ 2014-11-05. doi:10.5281/zenodo.12586
4. Khalid Belhajjame, Jun Zhao, Daniel Garijo, Matthew Gamble, Kristina Hettne, Raul Palma, Eleni Mina, Oscar Corcho, José Manuel Gómez-Pérez, Sean Bechhofer, Graham Klyne, Carole Goble (2015): **Using a suite of ontologies for preserving workflow-centric research objects**, *Web Semantics: Science, Services and Agents on the World Wide Web*. doi:10.1016/j.websem.2015.01.003
5. Daniel Garijo, Yolanda Gil, Oscar Corcho (2014): **Towards workflow ecosystems through semantic and standard representations**. *Proceeding WORKS '14 Proceedings of the 9th Workshop on Workflows in Support of Large-Scale Science*. doi:10.1109/WORKS.2014.13 [pdf]
6. Víctor Cuevas-Vicenttín, Parisa Kianmajd, Bertram Ludäscher, Paolo Missier, Fernando Chirigati, Yaxing Wei, David Koop, Saumen Dey (2014): **The PBase Scientific Workflow Provenance Repository**. *International Journal of Digital Curation* **9**(2). doi:10.2218/ijdc.v9i2.332
7. Stian Soiland-Reyes, Alan R Williams (2016): **What exactly happened to LSID?** *myGrid developer blog*, 2016-02-26. http://dev.mygrid.org.uk/blog/2016/02/what-exactly-happened-to-lsid/ doi:10.5281/zenodo.46804
8. Julie A McMurry, Niklas Blomberg, Tony Burdett, Nathalie Conte, Michel Dumontier, Donal Fellows, Alejandra Gonzalez-Beltran, Philipp Gormanns, Janna Hastings, Melissa A Haendel, Henning Hermjakob, Jean-Karim Hériché, Jon C Ison, Rafael C Jimenez, Simon Jupp, Nick Juty, Camille Laibe, Nicolas Le Novère, James Malone, Maria Jesus Martin, Johanna R McEntyre, Chris Morris, Juha Muilu, Wolfgang Müller, Christopher J Mungall, Philippe Rocca-Serra, Susanna-Assunta Sansone, Murat Sariyar, Jacky L Snoep, Natalie J Stanford, Neil Swainston, Nicole L Washington, Alan R Williams, Katherine Wolstencroft, Carole Goble, Helen Parkinson (2015): **10 Simple rules for design, provision, and reuse of identifiers for web-based life science data**. *Zenodo*. Submitted to PLoS Computational Biology. doi:10.5281/zenodo.31765