Collaborative provenance for workflow-driven science and engineering

Altıntaş, İ.

Citation for published version (APA):
Addressing Interoperability in Collaborative Provenance

“Ordinary language carries with it conditions of meaning which it is easy to recognize by classifying the contexts in which the expression is employed in a meaningful manner.”

– Paul Ricoeur

Throughout the previous chapters of this thesis, we focused on collaborative provenance where scientific collaboration involves data and workflow sharing between separate groups and data products of scientific workflow runs are published and then used by other researchers as inputs to their workflows. This requires linking of not only workflow provenance, but also the data artifacts that flow through these systems using descriptive metadata. For the purposes of the explained data and querying model, we assumed that data is linked through a global identification scheme and data identifiers are shared between different workflow executions and their provenance. In the presence of multiple workflow systems executing on a set of data repositories and collecting provenance information in their local provenance models, there are significant challenges that need to be addressed for analysis of collaborative provenance. This chapter discusses our initial approaches as a step towards solving this problem.

9.1 Interoperability of Scientific Workflows and Their Provenance

Scientific workflow interoperability has been an evolving discussion in the context of scientific workflow systems. These studies are motivated by a number of scenarios where complex data artifacts that lead to a scientific discovery may result from processes that involve multiple scientific workflow systems and middleware tools. The workflow system interoperability discussion have been explored in multiple aspects including linking design execution environments with other model of computation (MoC) (Mandal et al. 2007) and the workflow language (Elmroth et al. 2010) at workflow and sub-workflow levels.
A relatively newer aspect of the workflow interoperability discussion is achieving interoperability through workflow-related data based on the provenance information that is collected for workflow design and execution. For such a scenario to be possible, integration of provenance models (Ludäscher et al. 2008) coming out of different systems through an overarching standard is required (Davidson and Freire 2008). Provenance Challenges\(^1\) brought together provenance models from multiple systems facilitating discussions for the development of Open Provenance Model (OPM) (Moreau et al. 2010).

Interoperability through provenance and data related to scientific workflows also requires an ability to link multiple workflow runs where the data generated by one workflow run can be used as an input by another workflow run (Missier et al. 2010a). A connection between workflow provenance and the data provenance at the data repository and identification level needs to be achieved (Altintas et al. 2010c) similar to the CAMERA usecase explained in Chapter 7. In (Altintas et al. 2010c), we describe how this is achieved in using a mapping tool between workflow related data artifacts and CAMERA’s semantic-aware database. (Missier et al. 2010a) extended this approach, focusing on a model for provenance sharing and a set of interoperability problems that emerge from the heterogeneity of workflow systems, data formats, and provenance models. Through an implementation of our model that we developed in the context of the Data Observation Network for Earth (DataONE\(^2\)) project and that can stitch together provenance records for runs from different Kepler and Taverna workflow runs, we provided a prototypical framework for seamless cross-system, collaborative provenance management and can be easily extended to include other systems. This approach is one of the first examples to workflow interoperability not only through often elusive workflow standards but through shared provenance information from public repositories through a set of operators.

In addition, in order to achieve provenance interoperability, (Ellqvist et al. 2009) described a mediator-based architecture for integrating provenance information from multiple sources through two key components: (i) a global mediated schema that is general and capable of representing provenance information represented in different model, and (ii) a query API. Through a usecase study, the authors described a new system-independent query API that is general and able to express complex queries over integrated provenance information from different sources.

### 9.2 Interoperability Scenarios based on Provenance Challenges

As an example to provenance interoperability between executions of scientific workflows implemented in different systems, we created a scenario, derived from the Third Provenance Challenge (PC3) along some prototypical collaborative queries.

---

\(^1\)Provenance Challenges website: http://twiki.ipaw.info/bin/view/Challenge/

\(^2\)http://dataone.org
9.2 Interoperability Scenarios Based on Provenance Challenges

9.2.1 PC3 Usecase

The workflows selected for PC3 are part of an image-processing pipeline in the Pan-STARRS\textsuperscript{3} project. A next generation panoramic telescope surveys the sky looking for asteroids or comets that may impact the Earth. The telescope may generate several Terabytes of data nightly, which must be reduced and stored into an object data management framework that is publicly accessible by astronomers. Based on this usecase, the main PC3 workflow ingests CSV files containing readings from the telescope into an SQL database and the plotting workflow creates histograms of the ingested data.

To build a collaborative workflow environment, we assume that we executed the fragments of these PC3 workflows in three different workflow systems as shown in Figure 9.1. In this scenario, Taverna (\textit{Oinn et al.} 2006) performs the initialization and pre-loading checks, WS-VLAM (Korkhov et al. 2007) loads the CSV files into the database and updates the column counts, and Kepler (Ludäscher et al. 2006) creates the histograms. We chose this division of the PC3 workflows to evenly and logically divide the tasks among the workflow engines.

An example history of observables and actions within this usecase is shown in Table 9.1. In this scenario, all three workflow engines use the same database when executing their subset of the PC3 workflows. In the pre-load tasks, Taverna verifies the contents of the input CSV files and creates the tables in the database. Next, WS-VLAM reads the contents of the CSV files into these tables, and verifies the row counts and data values. Finally, Kepler produces histograms from these data. For example, the second row refers to run\textsubscript{1} performed by \textit{u2}.

\textsuperscript{3}http://pan-starrs.ifa.hawaii.edu
Table 9.1: The publish and run observables in interoperable PC3 scenario. The contents of the table shall be read as follows: e.g., the second row refers to run₁ performed by u₂ using \( w_f\text{preload} \) published by u₁. In run₁, u₂ used \( d_{J062941} \) as an input and the run produced \( d_{J062941-1} \) as its output.

<table>
<thead>
<tr>
<th>u₁</th>
<th>Published</th>
<th>( w_f\text{preload} )</th>
<th>u₂</th>
<th>Performed</th>
<th>Run₁</th>
<th>Used</th>
<th>( w_f\text{preload} )</th>
<th>Used</th>
<th>( d_{J062941} )</th>
<th>Produced</th>
<th>( d_{J062941-1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>u₁</td>
<td>Published</td>
<td>( w_f\text{preload} )</td>
<td>u₂</td>
<td>Performed</td>
<td>Run₁</td>
<td>Used</td>
<td>( w_f\text{preload} )</td>
<td>Used</td>
<td>( d_{J062941} )</td>
<td>Produced</td>
<td>( d_{J062941-1} )</td>
</tr>
<tr>
<td>u₁</td>
<td>Performed</td>
<td>Run₂</td>
<td>Used</td>
<td>( w_f\text{preload} )</td>
<td>Used</td>
<td>( d_{J062942} )</td>
<td>Produced</td>
<td>( d_{J062942-2} )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u₄</td>
<td>Published</td>
<td>( w_f\text{preload} )</td>
<td>u₂</td>
<td>Performed</td>
<td>Run₁</td>
<td>Used</td>
<td>( w_f\text{visualize} )</td>
<td>Used</td>
<td>( d_{J062941} )</td>
<td>Produced</td>
<td>( d_{J062941-1} )</td>
</tr>
<tr>
<td>u₅</td>
<td>Published</td>
<td>( w_f\text{visualize} )</td>
<td>u₂</td>
<td>Performed</td>
<td>Run₁</td>
<td>Used</td>
<td>( w_f\text{visualize} )</td>
<td>Used</td>
<td>( d_{J062941} )</td>
<td>Produced</td>
<td>( d_{J062941-2} )</td>
</tr>
<tr>
<td>u₃</td>
<td>Published</td>
<td>( d_{histogram} )</td>
<td>u₅</td>
<td>Published</td>
<td>( d_{histogram} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

using \( w_f\text{preload} \) published by u₁. In run₁, u₂ used \( d_{J062941} \) as an input and the run produced \( d_{J062941-1} \) as its output.

Collaborative PC3 Queries

Once the information such as the one in Table 9.1 is captured, it becomes possible to answer queries involving provenance data coming out of multiple workflow systems using the collaborative provenance model. The example collaborative provenance queries below show how our QLP extensions can be used to express such queries without writing complex SQL queries.

**Q1.** What data contributed to \( d_{histogram} \)?

\[
\text{DATA-DEP}( \ast \ldots d_{histogram} )
\]

**Q2.** If \( d_{J062942-2} \) is determined to be faulty, what other data products may be faulty based on \( d_{J062942-2} \)?

\[
\text{DATA-DEP}( d_{J062942-2} \ldots \ast )
\]

**Q3.** What runs contributed to the generation of \( d_{J062941-2} \)?

\[
\text{RUN-DEP}( \ast \ldots d_{J062941-2} )
\]

**Q4.** Which users contributed workflows that produced \( d_{histogram} \)?

\[
\text{COLLAB-DEP}( \ast \ldots d_{histogram} )
\]

9.2.2 PC1 Usecase

A similar example based on the First Provenance Challenge was implemented in the context of a DataONE student project called Data Tree of Life (DTOL). This scenario was published in (Moreau et al. 2010). The DTOL solves the problem of instance level bridging of provenance information that is a fundamental requirement for querying collaborative provenance.
Figure 9.2: DToL system architecture, implementing workflow run provenance interoperability between Kepler and Taverna: data is published by copying from local stores $S_T$, $S_K$ to the public store $S_P$, generating public identifiers in the process. Local provenance records are mapped to a common model and published, replacing local data references with global ones in the common provenance store, CPS.

Figure 9.2 shows the implemented prototype provenance sharing architecture, using the Kepler and Taverna scientific workflow systems as testbeds.

To test the hypothesis that provenance records for runs from different workflows and systems can be joined together, we have taken the First Provenance Challenge (Moreau et al. 2008) workflow, and split it into two parts. Both sub-workflows were encoded in Kepler and Taverna, and combined in a “crossing-over” manner in several ways. Overall, three different models of computation (MoCs) and corresponding designs were used, Taverna, “conventional” Kepler (Altintas et al. 2006a), and Kepler/COMAD (McPhillips et al. 2006), a novel MoC with its own provenance recorder to handle fine-grained dependencies within nested data collections (Bowers et al. 2007, Anand et al. 2009c).

In the implementation, we have used a combination of native provenance models, available from both workflow systems, as well as OPM provenance graphs derived from those models (Fig. 9.2). The former include all the read and write observable events, as well as the information about the workflow structure, while the latter contains instances of the data
dependency relation.

The Kepler and Taverna native provenance models differ in their data model, but can be mapped to the common provenance model with relatively little effort. The mapping tools obtain their source provenance information using an API that operates over a relational representation of the native provenance model, as well as over the OPM provenance graphs. In the case of Kepler and Taverna, the common model turns out to be less expressive than the native model, specifically with regards to the representation of nested lists, the only data structure used in Taverna (and one of many data structures supported by Kepler). This is not a problem, however, because the representation of the data structure (essentially a tree) is still maintained in the common data store.

The Kepler and Taverna storage models follow our conceptual model closely, in that all process data is stored in an internal, system-dependent data store. Provenance instances for workflow executions typically contain data references, but can also have embedded data values (for small data sizes).

![Architecture for answering collaborative queries](image)

Figure 9.3: Architecture for answering collaborative queries

### 9.3 QLP-based Interoperable Query Framework for Provenance

An important aspect of a provenance framework is being able to query the collected data, including collaborative provenance. This has been demonstrated by the simplicity of the QLP queries compared to the conventional queries in this thesis. Adoption of and extensions to the high-level Query Language for Provenance (QLP) with additional constructs allows
non-expert users to express collaborative provenance queries against this model easily and concisely.

For this purpose, we have designed an architecture that can be deployed on top of an existing system to execute both standard and collaborative queries expressed in QLP. Figure 9.3 shows the design of this end-to-end framework that can be plugged into any scientific infrastructure with the ability to publish data and workflows, to execute workflows using different workflow engines, to collect workflow provenance and to express and evaluate QLP queries.

In this architecture, workflows use a shared data space with common data identifiers. To generate data dependency views, using the QLP mapping to OPM, the QLP Querying Engine transforms users QLP queries into OPM queries. In addition, the same querying engine routes the mapped queries to distinct provenance stores using the developed SQL (RDBMS), XQuery (XML) and SPARQL (RDF) interfaces. This allows for execution of interoperable collaborative queries using a common data model similar to the one explained in the previous section. In addition, it allows for lower-level data and process dependencies (in one workflow run) using the provenance models and query engines provided by different workflow systems.

Summary

The interoperability of the provenance information collected by different scientific workflow systems is key to achieving a collaborative provenance framework since the ability to link workflow runs at large depends on it. The requirements and challenges for interoperable workflow provenance is summarized. Through interoperability scenarios based on Provenance Challenge workflows, prototype architectures towards solving interoperability during workflow stitching and provenance querying is presented. Next other future work plans and conclusions will be discussed.