VisTrails Provenance Traces for Benchmarking

Fernando Chirigati
Polytechnic Institute of NYU
fchirigati@nyu.edu

David Koop
Polytechnic Institute of NYU
dkoop@poly.edu

Juliana Freire
Polytechnic Institute of NYU
juliana.freire@nyu.edu

Cláudio Silva
Polytechnic Institute of NYU
csilva@nyu.edu

1. THE VISTRAILS SYSTEM

VisTrails (`http://www.vistrails.org`) is an open-source provenance management and scientific workflow system designed to support the scientific discovery process [2, 5]. It provides support for data analysis and visualization, together with a user-centered design. The system combines and substantially extends useful features of visualization and scientific workflow systems, enabling users to create complex workflows that encompass important steps of scientific discovery, from data gathering and manipulation, to complex analyses and visualizations, all integrated in a single system.

A key feature of VisTrails is its *comprehensive provenance infrastructure* that maintains detailed history information about the steps followed and data derived in the course of an exploratory task [4]—VisTrails maintains provenance of data products, of the workflows that derive these products, and of their executions. The system keeps track of *prospective provenance*, *retrospective provenance*, and *workflow evolution* [3], which help users to reason about the results, to follow chains of reasoning backward and forward, and to navigate through workflow versions in an intuitive way, using a history tree. Users do not lose any results, even when undoing changes.

2. THE VISTRAILS PROVENANCE SDK

The change-based provenance used in the VisTrails workflow system has also been generalized to support any application. VisTrails, Inc., has developed the Provenace Software Development Kit (ProvSDK) to capture the evolution of all results for an application [7]. Each application defines methods for serializing and deserializing actions, but the SDK takes care of metadata, version dependencies, and interfaces for search, playback, and inspection.

3. PROVENANCE TRACES

The provided traces are from the scientific and information visualization domain, and they encompass the three types of provenance captured by VisTrails: prospective, retrospective and workflow evolution. We include workflows that, for instance, read structured and unstructured grid data, extract an isosurface from a model and render surfaces and volumes. Additionally, we include some provenance traces generated in the VisTrails plugin for Autodesk Maya [1], which uses an early version of the VisTrails Provenance SDK [7] to transparently capture the provenance of the user's actions when building three-dimensional models.

4. PROVENANCE QUERIES

Below are some possible provenance queries that can be evaluated using our provided traces.

- What was the set of parameters used in module m?
- How many times was module m executed?
- How long did the execution of module m take?
- To which module m is module execution m' related?
- In which workflow version v was module m added?
- In which workflow version v was parameter p was set?
- To which workflow version v is module execution m' related?
- From which version v was verion v' derived?
- When did user u last modify version v?

5. FINAL CONSIDERATIONS

Applications. The ability to navigate through different versions and compare them, never losing previous results, is one of the key features of VisTrails and the ProvSDK, and the provenance traces contain information with respect to the workflow evolution. Applications interested in keeping track of all user's actions can directly benefit from the submitted provenance traces by looking at how VisTrails systematically stores the workflow versions.

Coverage of PROV. Some of the VisTrails native schema terms correspond to the PROV data model. In fact, there is a translation from the VisTrails schema to PROV, and VisTrails provides a serialization to XML (PROV-XML). Table 1 presents the coverage of PROV in VisTrails.

6. SUMMARY OF SUBMISSION

Table 2 presents a summary of the VisTrails provenance traces' submission.

Table 1: Coverage of PROV in VisTrails

PROV-O Term	Covered?
prov:Activity	Y
prov:Agent	Y
prov:Entity	Y
prov:actedOnBehalfOf	N
prov:endedAtTime	Y
prov:startedAtTime	Y
prov:used	Y
prov:wasAssociatedWith	Y
prov:wasAttributedTo	N
prov:wasDerivedFrom	N
prov:wasGeneratedBy	Y
prov:wasInformedBy	N

Table 2: Summary of submission

Data Format	XML
Data Model	VisTrails native schema
Size	5.2 MB
Tools	VisTrails system
Application Domain	Visualization
Submission Group	Refer to authors and affiliation
Contact	Refer to authors and affiliation
License	cc-by-nc-sa [6]

7. REFERENCES

[1] Maya. http://usa.autodesk.com/maya/.

[2] J. Freire, D. Koop, E. Santos, C. Scheidegger, C. Silva, and H. T. Vo. *The Architecture of Open Source Applications*, chapter VisTrails. Lulu.com, 2011.

[3] J. Freire, D. Koop, E. Santos, and C. T. Silva. Provenance for computational tasks: A survey. *Computing in Science and Eng.*, 10(3):11–21, May 2008.

[4] J. Freire, C. Silva, S. Callahan, E. Santos, C. Scheidegger, and H. Vo. Managing rapidly-evolving scientific workflows. In *International Provenance and Annotation Workshop (IPAW)*, LNCS 4145, pages 10–18. Springer Verlag, 2006.

[5] J. Freire and C. T. Silva. Making computations and publications reproducible with vistrails. *Computing in Science and Engineering*, 14(4):18–25, 2012.

[6] Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License. `http://creativecommons.org/licenses/by-nc-sa/3.0/`.

[7] VisTrails Provenance SDK. `http://www.vistrails.com/sdk.html`.