The W3C PROV family of specifications for modelling provenance metadata

Paolo Missier
School of Computing Science
Newcastle University, UK
Paolo.Missier@ncl.ac.uk

Khalid Belhajjame
School of Computer Science
University of Manchester, UK
khalidb@cs.man.ac.uk

James Cheney
School of Informatics
University of Edinburgh, UK
jcheney@inf.ed.ac.uk

ABSTRACT

Provenance, a form of structured metadata designed to record the origin or source of information, can be instrumental in deciding whether information is to be trusted, how it can be integrated with other diverse information sources, and how to establish attribution of information to authors throughout its history. The PROV set of specifications, produced by the World Wide Web Consortium (W3C), is designed to promote the publication of provenance information on the Web, and offers a basis for interoperability across diverse provenance management systems. The PROV provenance model is deliberately generic and domain-agnostic, but extension mechanisms are available and can be exploited for modelling specific domains. This tutorial provides an account of these specifications. Starting from intuitive and informal examples that present idiomatic provenance patterns, it progressively introduces the relational model of provenance along with the constraints model for validation of provenance documents, and concludes with example applications that show the extension points in use.

Categories and Subject Descriptors

E [**Data**]: General; H.2.3 [**Database Management**]: Languages—*Data description languages*

General Terms

Design,Standardization

1. MODELLING PROVENANCE

The current definition of the term *provenance* by the W3C[1], with reference to data, is the following: "Provenance refers to the sources of information, such as entities and processes, involved in producing or delivering an artifact." This very broad definition borrows largely from the historical meaning of provenance, which according to the Oxford English Dictionary refers to "The fact of coming from some particular source or quarter; source, derivation", and initially applied primarily to historical objects and works of art[2].

Within the scope of Data and Information Management, such broad definition has been articulated in various forms, each with a precise technical meaning. Traditionally, there have been two main camps. Firstly, in the (relational) database context, research on *database provenance* is concerned with formally characterizing and computing the provenance of a query result, that is, of the information required to answer specific questions on how, and why, a certain data item has come to be part of the result of a query. This terminology was first proposed by Buneman, Khanna and Tan [3] and the essential research in this area is summarized in [5]. Secondly, *process provenance* has come to denote a set of data dependencies that account for the generation of a piece of data as a result of a sequence of process transformations. This latter definition has been successfully applied to data transformation and analysis pipelines, primarily for computational science. In this setting, provenance is a particular data model designed, essentially, to represent executions of processes encoded using workflow or scripting languages.

The two approaches, for database and process provenance respectively, are still perceived as fairly distinct (the former is also known as *fine-grained* provenance, as it describes tuple-level transformations. This is in contrast to the latter, termed *coarse-grained* because of the black-box nature of the composing elements of workflows). Regarding the former, a few implementations of database extensions for provenance management are available [7, 1, 13]. At the same time the increasing popularity, amongst scientists, of workflows as a high-level programming paradigm, has ensured that provenance recording, storage, and query architectures are now available for a number of scientific workflow management systems, which are essentially implementations of various types of dataflow models.

For further reading on the state of the art in provenance research and practice, a recommended recent resource is the report of the March 2012 Dasgtuhl seminar on *Principles of Provenance* [6]. This comprehensive report includes contributions on both types of provenance, both of theoretical and more practical and applied nature.

2. TUTORIAL FOCUS AND SCOPE

Against this backdrop, this tutorial is focused on a new data model for provenance, simply called **PROV**, which is

[1] `http://www.w3.org/2005/Incubator/prov/wiki/What\_Is\_Provenance`

[2] `http://en.wikipedia.org/wiki/Provenance`

EDBT/ICDT '13 March 18 - 22 2013, Genoa, Italy

being standardised by the Provenance Working Group at the World Wide Web Consortium (W3C)[3]. The model is not tailored to database provenance or to any specific scientific application. Instead, it is meant to be generic and accommodate the provenance of data that is generated from a variety of diverse data sources, including human information processing. The group's main goal, as stated in its charter[4], is to promote interoperability amongst a diverse variety of provenance producers and consumers. This is accomplished by a suite of specifications, which encompass a conceptual model for provenance along with multiple encodings for interoperability, and with a formal semantics. At the core of the family is a data model [PROV-DM], which is essentially a relational model that captures the intrinsic elements of provenance, and that can be extended to accommodate the requirements of specific application domains. The model is expressed as an OWL ontology [PROV-O] but a human-readable relational syntax [PROV-N] and an XML encoding [PROV-XML] are also provided. Furthermore, a strong notion of *valid* provenance is defined by means of a system of constraints on the model [PROV-CONSTR] using first-order logic. A further document specifies mechanisms for accessing provenance documents on the Web [PROV-AQ].

2.1 Tutorial structure

The tutorial is structured in three parts. The first part offers an intuitive overview of the PROV conceptual model, using examples from an existing PROV Primer document as a starting point, and then delving into the technical details of the [PROV-DM] specification. The relational syntax [PROV-N] is used in the examples. A brief overview of the ontology and of the RDF representation of provenance documents are also provided.

The second part introduces the Constraints of the provenance model [PROV-CONSTR], and illustrates their rationale and usage by showing examples of valid and invalid provenance. The constraints are essentially first-order formulas similar to tuple-generating and equality-generating dependencies used in data exchange [11].

Finally, the third part presents applications and extensions to the model, which third parties are proposing in order to accommodate specific application requirements. These include for instance the description of structural elements of programs (workflows) that are responsible for data production, as well as an implementation of the model using a native graph database.

2.2 A PROV taster

As an example of the relations available in PROV, consider the following account of how a document was collaboratively edited and published by a group of co-authors, led by Alice and including Bob and Charlie[5]. Bob produced an initial `draft-v1` of the document, which includes references to two papers, `paper1` and `paper2`. Alice then typed some comments into document `draft-comments`, including the recommendation to also consider `paper3` in the next revision. Bob then used those comments to produce version `draft-v2` of the document. At this point Charlie, who like Bob works for Alice, published the document as Working Draft `WD1`, using the publication guidelines `pub-guide-v1` issued by the W3C. He, however, ignored version `pub-guide-v2` of those guidelines, which the W3C had issued as update before the publication process was completed.

Fig. 1 shows a graph representation of the provenance statements that describe this scenario (including additional details which are now discussed here). The nodes represent instances of the three types of PROV elements, namely **Entities** (the documents, the papers), **Activities** (drafting, commenting, ...) and **Agents** (Alice, Bob,...). Directed edges represent relations (**derivation**, **generation**, **usage**, **association**, ...) that hold amongst these elements[6]. Additionally, each of the elements may be annotated with attributes, both pre-defined (**type**, **role**) or user-defined (status, version,...). A complete account of PROV relations, expressed in a human-readable relational notation, is provided in the first part of the tutorial.

The second part elaborates on the notion of *valid* provenance statements, which is defined with reference to a set of constraints that a set of statements must satisfy. These are defined in [PROV-CONSTR]. For instance, a set of constraints defines the temporal interpretation of a set of provenance statements. Consider for example the two edges in the graph, representing that entity `draft-v1` *was generated by* activity `drafting`, and that `draft-v2` *was generated by* `editing`. In PROV, one instantaneous event is associated with each generation statement, in this example let these be `gen1`, `gen2`, respectively. Events are temporally ordered, and inferences may sometimes be made regarding the relative order of two events. In this example, these two events together with the additional statement found in the graph: `draft-v2` *was derived from* `draft-v1`, entails that `gen1` strictly precedes `gen2`. If an additional edge existed in the graph, stating that `draft-v1` *was derived from* `draft-v2`, one would also infer that `gen2` strictly precedes `gen1`, leading to an inconsistency.

The third part of the tutorial presents an overview of emerging applications that use the PROV model, in some cases by extending it, for capturing provenance traces. These are briefly described next.

3. PROV EXTENSIONS

We will also present a number of emerging applications that use the PROV model for capturing provenance traces.

3.1 PROV and Dictionaries

PROV Dictionary[7] extends PROV to provide the means for tracking the provenance of collections of entities, as well as that of their members. Dictionaries are logical structure consisting of key-entity pairs, and act as a generic indexing mechanism, a.k.a. maps in the literature, to represent commonly used data structures, e.g., relational tables and ordered lists. To track their provenance, PROV dictionary provides relationships for asserting the membership to a dictionary and for recording the history of members insertion and removal to and from a dictionary.

[3] `http://www.w3.org/2011/prov/wiki/`
[4] `http://www.w3.org/2011/01/prov-wg-charter`
[5] Adapted from [14].

[6] Relations are generally n-ary. Here the edges connect the the main two elements of a relation tuple.
[7] `http://www.w3.org/TR/prov-dictionary`

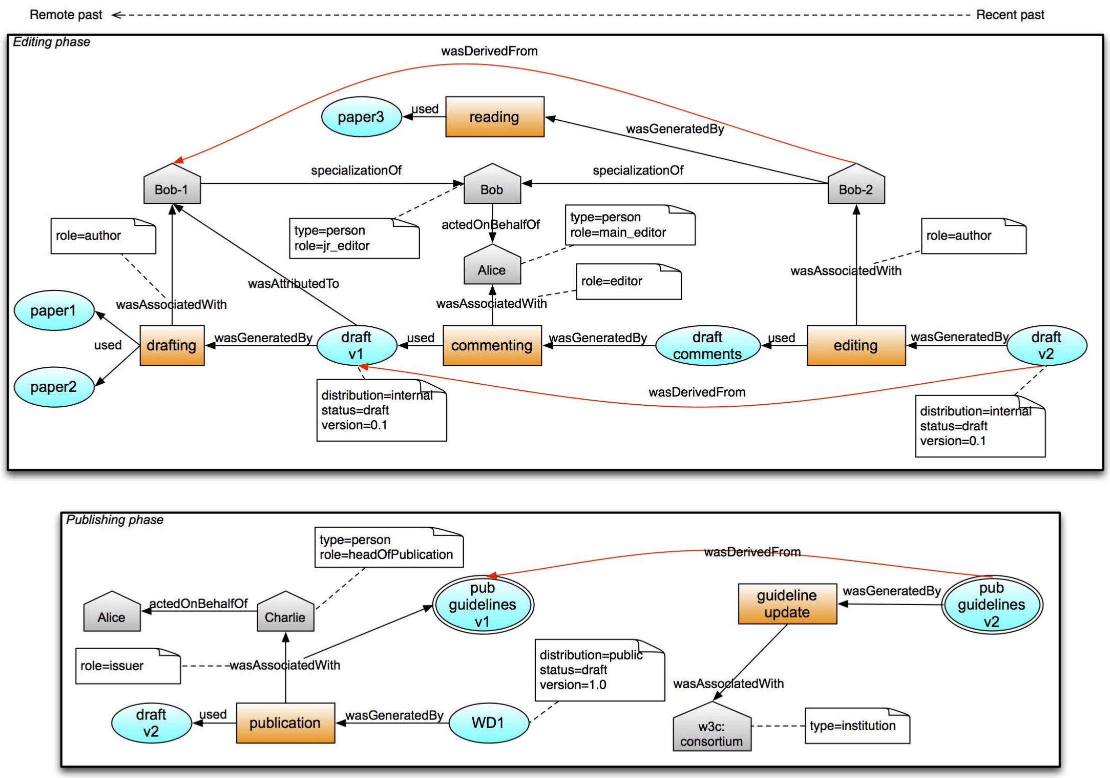

Figure 1: Graph representation of a set of PROV provenance statements

3.2 PROV and Scientific Workflows

Scientific workflows have gained considerable momentum as a mechanism for specifying and automating the execution of scientific experiments [9], as suggested by the number of scientific fields that adopted workflows, including, bioinformatics, cheminformatics and astronomy, to cite a few. In bioinformatics, for example, they are used for rapid implementation of in-silico experiments. Using workflows, an experiment is modeled as a network of analysis operations, connected together by data links describing how the operations are to be composed, so that the outputs of some operations are fed into the inputs of others.

As well as promoting rapid specification and automatic execution of scientific experiments, workflow systems can be instrumented in a straightforward manner to capture provenance information about the experiment execution. Typically, the provenance traces generated capture information about the data products used and generated by the steps that compose the workflows, and the data and temporal dependencies between those steps. Provenance traces of such a form have numerous applications. For example, they can used for workflow debugging, to detect the workflow steps that are responsible for workflow failure and to identify the steps that were affected as a result, and for checking the reproducibility of workflow results, by comparing the data products generated by the executions of the same workflow over time.

As mentioned earlier, PROV is a generic model for provenance. We will discuss how the PROV constructs were adapted and extended to capture the provenance traces of scientific workflows. We will discuss the PROV extensions adopted by the scientific workflow system Taverna [15], and D-PROV [8], the provenance model specified by the DataONE scientific workflow and provenance working group.

3.3 PROV and Executable Documents

While electronic papers have played and continue to play a primordial role in the dissemination of research results, researchers now recognize that they are by no means sufficient to communicate and share research results. The hypothesis investigated during the research, the experiment designed to assess the validity of the hypothesis, the process (workflow) used to run the experiment, the datasets used and the results produced by the experiment, and the conclusions drawn by the scientist, are all elements that may be needed to understand, assess the claim, or be able to re-use the results of previous research investigations.

A handful of research projects are currently investigating executable papers. For example, the Wf4Ever project is using research objects [2] as an abstraction for communicating, sharing and reusing research results. Central to the notion of research object is the provenance of the elements that compose the research object and the research object as a whole. PROV is used as the underlying model to capture the provenance of the research object elements, and to trace the evolution of a research object over time.

DEEP (Documents with Embeded Execution and Provenance) [16] is an executable document platform that is targeted towards readers. It provides readers with access to both static and dynamically generated contents, by combining document presentation with a computational back-end. In particular, DEEP allows readers to interactively explore the material assembled within the document, and trigger the creation of new resources. The actions of readers and their consequences are captured using a provenance model that extends PROV. Collected provenance traces are used to improve readers experience. In particular, provenance traces are used to cache computation results. Moreover, they can be used by users to check the reproducibility of the results reported on in the document. For example, the reader is able to trigger the execution of a computation using inputs that are different from those that are in the document, and compare the results obtained to those reported on by the authors of the document.

3.4 PROV and Smart Cities

Smart cities have emerged as a new concept in the last 5 years, underlining the importance of citizens (as a social capital) in ensuring the competitiveness of cities. Smart cities target a variety of issues, e.g., mobility, governance, economy, environment, and rely on citizens participation and contribution. We will present in the tutorial, two smart cities projects, namely UrbanMatch [4] and CollabMap [10], that use and extend PROV.

The aim of UrbanMatch is to interlink urban-related datasets by exploiting the physical presence of people in the environment. Specifically, citizens are asked to rate links that associate point of interests in an urban environment to datasets containing images depicting those point of interests. To record provenance information about individuals and their contributions, UrbanMatch uses the Human Computation Ontology[8], which extends the PROV model.

CollabMap is another example of a smart cities application, which solicits citizen contributions to identify evacuation routes in residential areas. Citizens are asked to identify the outline of a building in a map, draw evacuation routes between buildings, as well as verify the contributions of other citizens. To assist in the verification of collected data, CollabMap records provenance information that logs citizens' actions. When exported for external use, such provenance information is expressed in PROV.

4. REFERENCES

[1] P. Agrawal, O. Benjelloun, et al. Trio: a system for data, uncertainty, and lineage. In *Proceedings of the 32nd international conference on Very large data bases*, VLDB '06, pages 1151–1154. VLDB Endowment, 2006.

[2] K. Belhajjame et al. Workflow-centric research objects: First class citizens in scholarly discourse. In *Proceedings of Sepublica 2012*, pages 1–12, Hersonissos, 2012.

[3] P. Buneman, S. Khanna, and W. C. Tan. Why and Where: A Characterization of Data Provenance. In *ICDT*, pages 316–330, 2001.

[4] I. Celino, S. Contessa, M. Corubolo, et al. Linking smart cities datasets with human computation - the case of UrbanMatch. In P. Cudré-Mauroux et al., editors, *ISWC*, volume 7650 of *Lecture Notes in Computer Science*, pages 34–49. Springer, 2012.

[5] J. Cheney, L. Chiticariu, and W.-C. Tan. Provenance in Databases: Why, How, and Where. *Foundations and Trends in Databases*, 1:379–474, 2009.

[6] J. Cheney, A. Finkelstein, B. Ludaescher, and S. Vansummeren. Principles of Provenance (Dagstuhl Seminar 12091). *Dagstuhl Reports*, 2(2):84–113, 2012.

[7] L. Chiticariu, W.-C. Tan, and G. Vijayvargiya. DBNotes: a post-it system for relational databases based on provenance. In *SIGMOD*, 2005.

[8] V. Cuevas-Vicenttin, S. Dey, and B. Ludaescher. Modeling and querying scientific workflow provenance in the D-OPM. In *WORKS*. ACM, 2012.

[9] E. Deelman, D. Gannon, M. S. Shields, and I. Taylor. Workflows and e-science: An overview of workflow system features and capabilities. *Future Generation Comp. Syst.*, 25(5):528–540, 2009.

[10] M. Ebden, T. D. Huynh, L. Moreau, et al. Network analysis on provenance graphs from a crowdsourcing application. In Groth and Frew [12], pages 168–182.

[11] R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data exchange: semantics and query answering. *Theor. Comput. Sci.*, 336(1):89–124, 2005.

[12] P. T. Groth and J. Frew, editors. *4th International Provenance and Annotation Workshop, IPAW 2012, Santa Barbara, CA, USA, June 19-21, 2012*, volume 7525 of *Lecture Notes in Computer Science*. Springer, 2012.

[13] G. Karvounarakis, Z. G. Ives, and V. Tannen. Querying data provenance. In *SIGMOD Conference*, 2010.

[14] P. Missier and K. Belhajjame. A PROV encoding for provenance analysis using deductive rules. In *Procs. IPAW'12*, Santa Barbara, California, 2012. Springer-Verlag, Lecture Notes in Computer Science.

[15] P. Missier, S. Soiland-Reyes, S. Owen, et al. Taverna, reloaded. In *Procs. SSDBM 2010*, volume 6187 of *Lecture Notes in Computer Science*, pages 471–481, Heidelberg, Germany, 2010. Springer.

[16] H. Yang, D. T. Michaelides, C. Charlton, et al. Deep: A provenance-aware executable document system. In Groth and Frew [12], pages 24–38.

Online references:

PROV-DM: `http://www.w3.org/TR/prov-dm/`

PROV-O: `http://www.w3.org/TR/prov-o/`

PROV-N: `http://www.w3.org/TR/prov-n/`

PROV-XML: `http://www.w3.org/TR/prov-xml/`

PROV-CONSTR: `http://www.w3.org/TR/prov-constraints/`

PROV-AQ: `http://www.w3.org/TR/prov-aq/`

PROV-Dictionary `http://www.w3.org/TR/prov-dictionary/`

[8] `http://swa.cefriel.it/ontologies/hc.html`