P

Provenance Standards

Paolo Missier
School of Computing Science, Newcastle University, Newcastle upon Tyne, UK

Synonyms

PROV

Definition

PROV, the Provenance standard, is a family of specifications released in 2013 by the Provenance Working Group, as a contribution to the Semantic Web suite of technologies at the World Wide Web Consortium. The specifications define a data model along with a number of serializations for representing aspects of provenance. The term provenance, as understood in these specifications, refers to *information about entities*, *activities*, *and people involved in producing a piece of data or thing*, *which can be used to form assessments about its quality*, *reliability*, *or trustworthiness* (PROV-Overview [1]). The specifications include a combination of W3C *Recommendation* and *Note* documents. Recommendation documents include:

(i) The main PROV data model specification (PROV-DM [2]), with an associated set of constraints and inference rules (PROV-CONSTRAINTS [3])
(ii) An OWL ontology that allows a mapping of the data model to RDF (PROV-O [4])
(iii) A notation for PROV with a relational-like syntax, aimed at human consumption (PROV-N [5])

All other documents are Notes. These include PROV-XML, which defines a XSD schema for XML serialization (http://www.w3.org/TR/prov-xml/); PROV-AQ, the Provenance Access and Query document (http://www.w3.org/TR/prov-aq/), which defines a Web-compliant mechanism to associate a dataset to its provenance; PROV-DICTIONARY (http://www.w3.org/TR/prov-dictionary/), for expressing the provenance of data collections defined as sets of key-entity pairs; and PROV-DC (http://www.w3.org/TR/prov-dc/), which provides a mapping between PROV-O and Dublin Core terms.

Historical Background

The idea of a community-grown data model for describing the provenance of data originated around 2006, when consensus began to emerge on the benefits of having a uniform representation for "data provenance, process documentation, data derivation, and data annotation", as stated in [6]. The first Provenance Challenge [7] was then launched, to test the hypothesis that heterogeneous systems (mostly in the

L. Liu, M.T. Özsu (eds.), *Encyclopedia of Database Systems*,
DOI 10.1007/978-1-4899-7993-3_80749-1

e-science/cyberinfrastructure space), each individually capable of producing provenance data by observing the execution of data-intensive processes, could successfully exchange such provenance observations with each other, without loss of information. The Open Provenance Model (OPM) [6] was proposed as a common data model for the experiment. Other Provenance Challenges followed, to further test the ability of the OPM to support interoperable provenance.

Central to the OPM is the notion of *causal relationships*, or dependencies, involving *artifacts* (e.g., data items), *processes*, and *agents*. Using the OPM, one can assert that an artifact A was produced or consumed by a process P, e.g., "the cake C was produced by a baking process B, which used eggs E and flour F." Here C, E, and F are artifacts, and B is a process. One can also assert a derivation dependency between two artifacts, A1 and A2, without mentioning any mediating process, i.e., "A2 was derived from A1." Agents, including humans, software systems, etc., can be mentioned in OPM as process controllers, i.e., "the baking was controlled by Bob (the cook)."

OPM statements attempt to explain the existence of artifacts. Since such statements may reflect an incomplete view of the world, obtained from a specific perspective, the OPM adopts an open world assumption, whereby dependencies are interpreted as correct but possibly incomplete knowledge: "A2 was derived from A1" asserts a certain derivation, but does not exclude that other, possibly unknown artifacts, in addition to A1, may have contributed to explaining the existence of A2. Other features of the OPM, including built-in rules for inference of new provenance facts, are described in detail in [6].

In September, 2009, the W3C Provenance Incubator Group was created. Its mission, as stated in the charter (http://www.w3.org/2005/Incubator/prov/charter), was to "provide a state-of-the art understanding and develop a roadmap in the area of provenance for Semantic Web technologies, development, and possible standardization." W3C Incubator Groups produce recommendations on whether a standardization effort is worth undertaking. Led by Yolanda Gil at USC/ISI, the group produced its final report in December 2010 (http://http://www.w3.org/2005/Incubator/prov/XGR-prov-20101214). The report highlighted the importance of provenance for multiple application domains, outlined typical scenarios that would benefit from a rich provenance description, and summarized the state of the art from the literature, as well as in the Web technology available to support tools that exploit a future standard provenance model. As a result, the W3C Provenance Working Group was created in 2011, chaired by Luc Moreau (University of Southampton) and Paul Groth (VU University Amsterdam). The group released its final recommendations for PROV in June 2013.

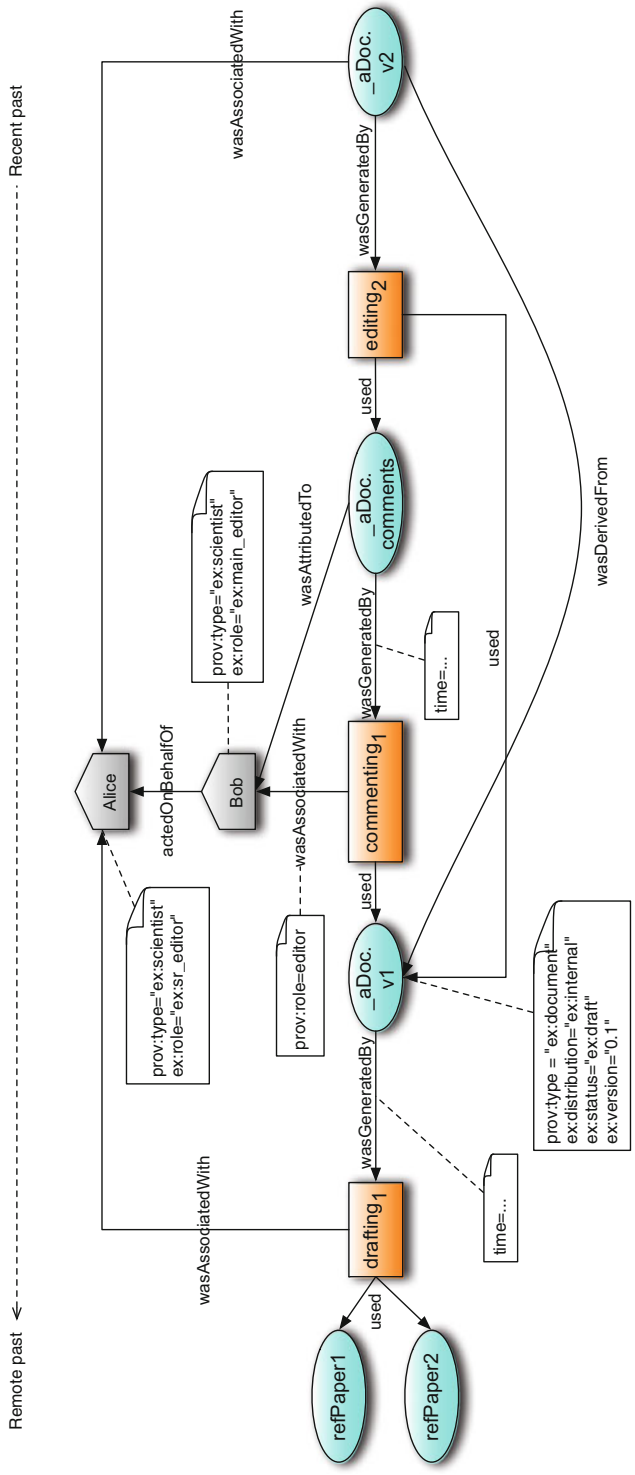
Scientific Fundamentals

While PROV builds upon the prior experience gained from the OPM, and therefore it echoes some of the notions presented above (see "Historical Background"), its design emerges from a more disciplined community effort, governed by standard W3C Working Group policy. PROV is the result of 2 years of work and incorporates the expectations of group members representing over 50 organizations from a broad range of application domains, each bringing different sets of requirements.

The brief account of PROV that follows cannot possibly cover all the features of the family of specifications. The reader is referred to the overview document [1], which provides the main entry point and a roadmap to the other documents, including the nonnormative Notes. What follows is a summary of the main principles that informed the design, following mainly the PROV-DM document [2] (please note, all sentences in italics below are quotes from that document).

The scenario depicted in Fig. 1 will be used to illustrate those principles. The primer document [8] also provides a complete running example.

In this scenario, two coauthors, Alice and Bob, are responsible for editing a document. After Alice has edited a first version draft, Bob comments on the draft, and then Alice edits a second version, based upon the first draft and Bob's comments.



Provenance Standards, Fig. 1 Document coediting scenario

P

Entities, Activities, and Agents

At the core of the PROV data model are the notions of entities, activities, and agents. *Provenance describes the use and production of entities by activities, which may be influenced in various ways by agents.*

Entities may represent data, but they are more generally defined as *physical*, *digital*, *conceptual*, *or other kind of thing with some fixed aspects.* Entities may be real or even imaginary. Examples of entities are a particular version of a document, the output produced by some algorithm, a record in a database, a car at a particular stage of its lifetime, etc. In practice, anything that may have a provenance is an entity.

Importantly, the "fixed aspects" mentioned informally above refer to the characteristics that are relevant to describing the provenance of the entity. For instance, a document as in Fig. 1 may be characterized by a filename, version number, and current content. Some of these properties may persist over time (e.g., the filename), while others may change. A document entity, *_aDoc.v1*, is a document with values specified for each of those properties. When any of those values change, a different document entity, for instance, *_aDoc.v2*, is defined, with a different provenance. In our example, this is the document with updated content and a new version number. The editing activity *editing2* accounts for the change relative to *_aDoc.v1*.

Unlike entities, activities such as *drafting1*, *commenting1*, and *editing2* have a duration: *An activity is something that occurs over a period of time and acts upon or with entities*; *it may include consuming*, *processing*, *transforming*, [...,] *using*, *or generating entities*. Typically, activities may *use* existing entities (*editing2* used entity *_aDoc.comments*) or *generate* new ones (*editing2* generated *_aDoc.v2*).

Agents, on the other hand, *bear some form of responsibility for an activity taking place*, *for the existence of an entity*, *or for another agent's activity*. Agents may be humans, as in the case of *Alice* and *Bob* in the example, or, for instance, software systems. Note that one may want to describe the provenance of an agent – for instance, to help explain their behavior vis a vis carrying out an activity (for instance, what knowledge did *Alice* have during her *drafting1* activity?) Therefore, in PROV, agents are viewed as a particular type of entity.

All entities, activities, and agents are given a unique ID, drawn from a specific namespace, which is valid within a given scope, i.e., a *provenance document*. Furthermore, they can be annotated with sets of properties, i.e., of name-value pairs. PROV reserves certain properties, using the PROV namespace. Thus, for instance, the following statement in PROV-N notation:

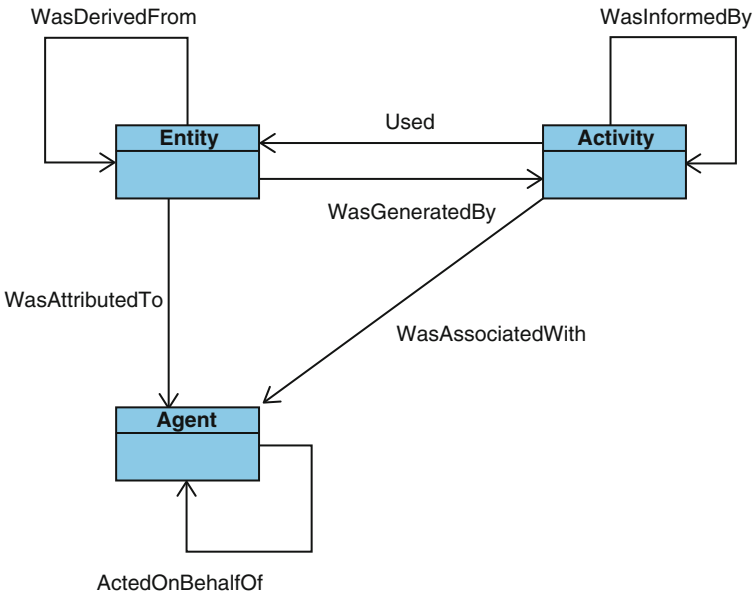
```
entity(ex:_aDoc.v1; [prov:type
= "ex:document", ex:distribution
="ex:internal",...])
```

defines a new entity with unique name *_aDoc.v1* in a custom namespace denoted by prefix *ex* and annotated with two properties, one of which is the standard *prov:type* property (but with a value in the *ex* namespace).

Core Provenance Relationships

The provenance of entities is expressed by means of a small set of core concepts, namely, generation, usage, derivation, communication, attribution, association, and delegation. In [2], these are defined independently of any formalism and are then manifested as relationships in a UML data model specification, reproduced in Fig. 2. The PROV-N notation [5] is recommended as a syntactic rendering of relationship instances and will be used in these examples.

A key principle is that entities have a lifetime, which begins with generation, defined as *the completion of production of a new entity by an activity*. *The entity did not exist before generation and becomes available for usage after this generation*. Associated with generation is a *generation event*, which can be thought of as a point in time. (However, PROV avoids explicit notions of time, which are difficult to manage when provenance is recorded by multiple distributed and autonomous systems, each possibly using a different clock.) Note that the production of an entity, for instance, a file produced incrementally by a program execution, may extend over time. In this case, the generation event marks the completion of the



Provenance Standards, Fig. 2 UML diagram for core PROV entities and relationships (From [PROV-DM])

production. Symmetrically, usage may also extend over time (i.e., the reading of the file). Thus, usage is defined as *the beginning of utilizing an entity by an activity*. Note how these definitions are suitable to model typical producer/consumer patterns in data processing.

Here are examples of generation and usage in the document editing scenario:

```
used(drafting1, refPaper1),
used(drafting1, refPaper2),
wasGeneratedBy(_aDoc.v2, drafting1)
```

as well as

```
used(editing2, _aDoc.comments)
wasGeneratedBy(_aDoc.v2, editing2)
```

In PROV, all relationships may optionally be given an explicit, unique ID, just like entities etc. Using IDs, the example above could also have been written as:

```
used(ex:u1; editing2, _aDoc.comments)
wasGeneratedBy(ex:g1; _aDoc.v2,
editing2)
```

where *ex*:*u1* and *ex:g1* are the new IDs. This design principle is useful when introducing derivations.

A derivation is a data dependency between two entities, e1 and e2, where e2 (the derived entity) is the result of some transformation that occurred to e1. The nature of such transformation may be implicit, for example:

```
wasDerivedFrom(_aDoc.v2, _aDoc.v1)
```

However, it may also be expressed in terms of a mediating activity *a* that "explains" the derivation in terms of usage of *e1* and generation of *e2*. More specifically, in abstract one could have the following statements, involving the IDs for generation and usage relationships:

```
used(u; a,e1)
wasGeneratedBy(g,e2,a)
wasDerivedFrom(e2, e1, a, g, u)
```

This is an example of a binary relationship, derivation, which admits additional arguments. Such optional arguments are common in PROV.

Similar to derivation, communication relates two activities, *a1* and *a2*, such that *a2* is dependent upon *a1*, *by way of some unspecified entity that is generated by a1 and used by a2*.

Constraints and Inferences

The previous example suggests, intuitively, possible connections among some of the relationships, e.g., derivation, usage, and generation. Such connections are indeed formalized, as part of a comprehensive normative PROV-CONSTRAINTS document [3], which specifies *definitions of some provenance statements*

in terms of others, *inferences over PROV instances that applications may employ*, and *a class of valid PROV instances by specifying constraints that valid PROV instances must satisfy*. In this context, the term *valid* refers to *a consistent history of objects and interactions to which logical reasoning can be safely applied*. As an example, returning to the derivation/usage/generation statements, consider the following inference rule, from [3#inf 11]:

```
IF wasDerivedFrom(_id; e2,e1,a,gen2,
use1,_attrs),
THEN there exists _t1 and _t2 such that
used(use1; a,e1,_t1,[])
and
wasGeneratedBy(gen2; e2,a,_t2,[]).
```

Here *_t1* and *_t2* indicate timestamps, a detail that can be overlooked at this stage. Informally the rule states that if derivation of *e2* from *e1* involves an activity *a*, then *a* must be involved in the usage of *e1* and the generation of *e2*.

The simple usage/generation provenance pattern shown earlier is helpful to illustrate *event ordering constraints*. Intuitively, if an activity *a* generates (resp. uses) entities *e2* (resp. *e1*), then it must be the case that the generation (resp. usage) event lies within the lifetime of *a*. The start and end events of an activity can indeed be expressed, i.e.:

```
wasStartedBy(start; a,_e1,_a1,_t1,
_attrs1)
wasEndedBy(end; a,_e2,_a2,_t2,_attrs2)
```

denote the start and end events, resp., for *a*.

Constraints 33 and 34 in [3] state that a pre-order relation must exist, whereby the start event (resp. end event) must precede (resp. not precede) any usage and generation event involving *a*. Formally:

```
IF wasStartedBy(start; a,_e1,_a1,_t1,
_attrs1) and used(use; a,_e2,_t2,
_attrs2) THEN start precedes use.
IF used(use; a,_e1,_t1,_attrs1) and
wasEndedBy(end; a,_e2,_a2,_t2,_attrs2)
THEN use precedes end.
IF wasStartedBy(start; a,_e1,_a1,_t1,
_attrs1) and wasGeneratedBy(gen; _e2,a,
_t2,_attrs2) THEN start precedes gen.
IF wasGeneratedBy(gen; _e,a,_t,_attrs)
and wasEndedBy(end; a,_e1,_a1,_t1,
_attrs1) THEN gen precedes end.
```

These exemplar rules and constraints are representative of a much larger collection. PROV-CONSTRAINTS include 21 inference rules and 55 constraints.

Actors and Their Relationships

The core elements of PROV also include three relationships involving agents, namely, attribution, association, and delegation. Their use is illustrated in the scenario of Fig. 1:

```
wasAttributedTo(_aDoc.comments, Bob)
```

denotes that entity *_aDoc.comments* is ascribed to Bob, without specifying any associated activity. One can also explicitly ascribe responsibility for an activity to an agent using the association relation, for example:

```
wasAssociatedWith(commenting1, Bob).
```

Inference rules are defined in [3], which formalize the relationship between attribution and association when they are both present.

Finally, one can specify chains of responsibility by means of the delegation relationship, as follows:

```
actedOnBehalfOf(Bob, Alice)
```

PROV-O

The PROV-O document [PROV-O] specifies the PROV data model as an OWL ontology. This makes it natural to express provenance statements, of the kind shown here using the PROV-N, as RDF triples. The PROV-PRIMER [8] document provides examples in both notations, as well as in PROV-XML (http://www.w3.org/TR/prov-xml/).

Extensibility

PROV is designed as *upper level* model, agnostic to any application domain. Two main mechanisms exist for extending the model. Firstly, one can introduce custom properties, as mentioned, as well as custom values for standard properties, such as *prov:type*. Secondly, one can extend the PROV-O ontology using the standard extension mechanisms available in the Semantic Web framework (e.g., subClass, subProperty).

Further Reading

A survey of foundations for Provenance on the Web, predating PROV, is available [9], as well as an introduction to PROV [10]. Research on Database Provenance has been developing alongside the more general model for provenance described here; however, it is not primarily within the scope of PROV. An account of Database Provenance can be found in [11]. Several tutorials on PROV are also available, including one that describes the data model, the constraints model, and a number of known implementations and extensions as of 2013 [12].

Key Applications

1. Attribution of user-authored Web pages (e.g., blogs) and social media content, trust in Web content.
2. Attribution of published science data, derivation history of scientific dataset that represents the outcome of experiments. In particular, workflow management systems have been early generators of provenance traces, which can be used to help validate published datasets in the eyes of potential new users. These include, among others, the VisTrails, Taverna, Kepler, Pegasus, and more. A recent analysis of workflow provenance can be found in [13].
3. Provenance, when it is sufficiently detailed, can be instrumental in some cases, to facilitate the reproducibility of scientific experiments [14–15].

Future Directions

Many implementations of PROV along with a variety of tools are currently being developed. An initial list, which however evolves rapidly, can be found in the PROV Implementation Report (http://www.w3.org/TR/prov-implementations/). Useful annual or biannual conferences to monitor include TAPP and IPAW.

URL to CODE

http://lucmoreau.github.io/ProvToolbox/

Cross-References

- Data Provenance
- Provenance in Scientific Databases
- Semantic Web Architecture

Recommended Reading

1. Groth P, Moreau L. PROV-Overview: an overview of the PROV Family of Documents [Internet]. 2012. Available from: http://www.w3.org/TR/prov-overview/
2. Moreau L, Missier P, Belhajjame K, B'Far R, Cheney J, Coppens S, et al. PROV-DM: the PROV Data Model [Internet]. In: Moreau L, Missier P, editors. 2012. Available from: http://www.w3.org/TR/prov-dm/
3. Cheney J, Missier P, Moreau L. Constraints of the provenance data model [Internet]. 2012. Available from: http://www.w3.org/TR/prov-constraints/
4. Lebo T, Sahoo S, McGuinness D, Belhajjame K, Cheney J, Corsar D, et al. PROV-O: the PROV ontology [Internet]. In: Lebo T, Sahoo S, McGuinness D, editors. 2012. Available from: http://www.w3.org/TR/prov-o/
5. Moreau L, Missier P, Cheney J, Soiland-Reyes S. PROV-N: the provenance notation [Internet]. In: Moreau L, Missier P, editors. 2012. Available from: http://www.w3.org/TR/prov-n/
6. Moreau L, Clifford B, Freire J, Futrelle J, Gil Y, Groth P, et al. The open provenance model – core specification (v1.1). Futur Gener Comput Syst Elsevier. 2011;7(21):743–56.
7. Moreau L, Ludäscher B, Altintas I, Barga RS. The first provenance challenge. Concurr Comput Pract Exp [Internet]. 2008;20:409–18. Available from: http://www3.interscience.wiley.com/journal/116837632/abstract
8. Gil Y, Miles S, Belhajjame K, Deus H, Garijo D, Klyne G, et al. PROV model primer [Internet]. In: Gil Y, Miles S, editors. 2012. Available from: http://www.w3.org/TR/prov-primer/
9. Moreau L. The foundations for provenance on the web. Found Trends Web Sci [Internet]. Citeseer; 2009 [cited 2011 Oct 18];131. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.155.784&rep=rep1&type=pdf

10. Moreau L, Groth P. Provenance: an introduction to PROV. Synth Lect Semant Web Theory Technol [Internet]. Morgan & Claypool Publishers; 2013 Sep 15 [cited 2014 Aug 25];3(4):10–129. Available from: http://www.morganclaypool.com/doi/abs/10.2200/S00528ED1V01Y201308WBE007
11. Cheney J, Chiticariu L, Tan W-C. Provenance in databases: why, how, and where. Found Trends{\textregistered} Databases. 2009;1:379–474.
12. Missier P, Belhajjame K, Cheney J. The W3C PROV family of specifications for modelling provenance metadata. In: Proceedings EDBT'13 (Tutorial) [Internet]. Genova: ACM; 2013. Available from: http://www.edbt.org/Proceedings/2013-Genova/papers/edbt/a80-missier.pdf
13. Bowers S. Scientific workflow, provenance, and data modeling challenges and approaches. J Data Semant [Internet]. 2012 Apr 11 [cited 2013 Jun 8];1(1):19–30. Available from: http://link.springer.com/10.1007/s13740-012-0004-y
14. Missier P, Woodman S, Hiden H, Watson P. Provenance and data differencing for workflow reproducibility analysis. Concurr Comput Pract Exp [Internet]. 2013;n/a–n/a. Available from: http://dx.doi.org/10.1002/cpe.3035
15. Peng R. Reproducible research in computational science. Science. 2011;334(6060):1226–127.