W3C Working Group Note

PROV Model Primer

W3C Working Group Note 30 April 2013

This version:
> http://www.w3.org/TR/2013/NOTE-prov-primer-20130430/
Latest published version:
> http://www.w3.org/TR/prov-primer/
Previous version:
> http://www.w3.org/TR/2013/WD-prov-primer-20130312/

Editors:
> Yolanda Gil, Information Sciences Institute, University of Southern California, US
> Simon Miles, King's College London, UK
Contributors:
> Khalid Belhajjame, University of Manchester
> Helena Deus, Digital Enterprise Research Institute (DERI), NUI Galway
> Daniel Garijo, Ontology Engineering Group, Universidad Politécnica de Madrid, Spain
> Graham Klyne, University of Oxford
> Paolo Missier, Newcastle University
> Stian Soiland-Reyes, University of Manchester
> Stephan Zednik, Rensselaer Polytechnic Institute

Abstract

This document provides an intuitive introduction and guide to the PROV Data Model for provenance interchange on the web. PROV defines a core data model for provenance for building representations of the entities, people and processes involved in producing a piece of data or thing in the world. This primer explains the fundamental PROV concepts and provides examples of its use. The primer is intended as a starting point for those wishing to create or use PROV data.

The PROV Document Overview describes the overall state of PROV, and should be read before other PROV documents.

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at http://www.w3.org/TR/.

PROV Family of Documents

This document is part of the PROV family of documents, a set of documents defining various aspects that are necessary to achieve the vision of inter-operable interchange of provenance information in heterogeneous environments such as the Web. These documents are listed below. Please consult the [PROV-OVERVIEW] for a guide to reading these documents.

- PROV-OVERVIEW (Note), an overview of the PROV family of documents [PROV-OVERVIEW];
- PROV-PRIMER (Note), a primer for the PROV data model (this document);
- PROV-O (Recommendation), the PROV ontology, an OWL2 ontology allowing the mapping of the PROV data model to RDF [PROV-O];
- PROV-DM (Recommendation), the PROV data model for provenance [PROV-DM];
- PROV-N (Recommendation), a notation for provenance aimed at human consumption [PROV-N];
- PROV-CONSTRAINTS (Recommendation), a set of constraints applying to the PROV data model [PROV-CONSTRAINTS];
- PROV-XML (Note), an XML schema for the PROV data model [PROV-XML];
- PROV-AQ (Note), mechanisms for accessing and querying provenance [PROV-AQ];
- PROV-DICTIONARY (Note) introduces a specific type of collection, consisting of key-entity pairs [PROV-DICTIONARY];
- PROV-DC (Note) provides a mapping between PROV-O and Dublin Core Terms [PROV-DC];
- PROV-SEM (Note), a declarative specification in terms of first-order logic of the PROV data model [PROV-SEM];
- PROV-LINKS (Note) introduces a mechanism to link across bundles [PROV-LINKS].

Implementations Encouraged

The Provenance Working Group encourages implementation of the material defined in this document. Although work on this document by the Provenance Working Group is complete, errors may be recorded in the errata or and these may be addressed in future revisions.

Please Send Comments

This document was published by the Provenance Working Group as a Working Group Note. If you wish to make comments regarding this document, please send them to public-prov-comments@w3.org (subscribe, archives). All comments are welcome.

Publication as a Working Group Note does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

This document was produced by a group operating under the 5 February 2004 W3C Patent Policy. W3C maintains a public list of any patent disclosures made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains Essential Claim(s) must disclose the information in accordance with section 6 of the W3C Patent Policy.

Table of Contents

1. Introduction

This primer document provides an accessible introduction to the PROV data model for provenance interchange on the Web. The *provenance* of digital objects represents their origins. PROV is a specification to express provenance records, which contain *descriptions* of the entities and activities involved in producing and delivering or otherwise influencing a given object. Provenance can be used for many purposes, such as understanding how data was collected so it can be meaningfully used, determining ownership and rights over an object, making judgements about information to determine whether to trust it, verifying that the process and steps used to obtain a result complies with given requirements, and reproducing how something was generated.

As a specification for provenance, PROV accommodates all those different uses of provenance. Different people may have different perspectives on provenance, and as a result different types of information might be captured in provenance records.

- One perspective might focus on *agent-centered provenance*, that is, what people or organizations were involved in generating or manipulating the information in question. For example, in the provenance of a picture in a news article we might capture the photographer who took it, the person that edited it, and the newspaper that published it.
- A second perspective might focus on *object-centered provenance*, by tracing the origins of portions of a document to other documents. An example is having a web page that was assembled from content from a news article, quotes of interviews with experts, and a chart that plots data from a government agency.
- A third perspective one might take is on *process-centered provenance*, capturing the actions and steps taken to generate the information in question. For example, a chart may have been generated by invoking a service to retrieve data from a database, then extracting certain statistics from the data using some statistics package, and finally processing these results with a graphing tool.

Provenance records are metadata. There are other kinds of metadata that are not provenance. For example, the size of an image is metadata of that image but it is not provenance. For general background on provenance, a comprehensive overview of requirements, use cases, prior research, and proposed vocabularies for provenance are available from the Final Report of the W3C Provenance Incubator Group [PROVENANCE-XG]. That document contains three general scenarios that may help identify the provenance aspects of planned applications and help plan the design of a provenance system.

This primer document aims to ease the adoption of the PROV specifications by providing:

- A high-level explanation of how PROV models provenance, in Section 2. A detailed description of all the concepts and relations in the PROV Data Model is provided in [PROV-DM].
- A simple self-contained example that illustrates how to produce and use PROV assertions, in Section 3. The example includes

snippets in RDF using the PROV ontology [PROV-O], in a notation designed for human consumption [PROV-N], and in PROV's XML format [PROV-XML]. The example shows how to combine PROV with other popular vocabularies such as FOAF [FOAF] and Dublin Core [DCTERMS].

The document ends with a summary of major capabilities and features of PROV.

2. Intuitive overview of PROV

This section provides an explanation of the main concepts in PROV. As with the rest of this document, it should be treated as a starting point for understanding the model. The PROV data model document [PROV-DM] provides precise definitions and constraints [PROV-CONSTRAINTS] to be followed.

The following diagram provides a high level overview of the structure of PROV records, limited to some key PROV concepts discussed in this document. Note that because PROV is meant to describe how things were created or delivered, PROV relations are named so they can be used in assertions about the past.
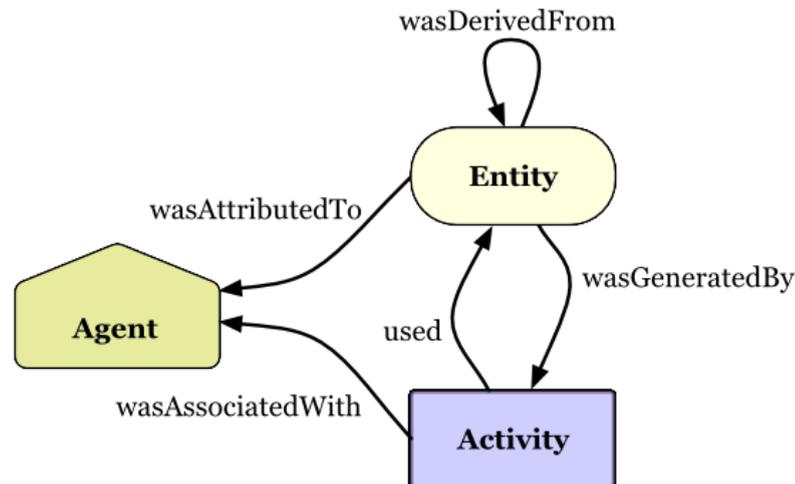
2.1 Entities

In PROV, physical, digital, conceptual, or other kinds of thing are called *entities*. Examples of such entities are a web page, a chart, and a spellchecker. Provenance records can describe the provenance of entities, and an entity's provenance may refer to many other entities. For example, a document D is an entity whose provenance refers to other entities such as a chart inserted into D, and the dataset that was used to create that chart. Entities may be described as having different attributes and be described from different perspectives. For example, document D as stored in my file system, the second version of document D, and D as an evolving document, are three distinct entities for which we may describe provenance.

2.2 Activities

Activities are how entities come into existence and how their attributes change to become new entities, often making use of previously existing entities to achieve this. They are dynamic aspects of the world, such as actions, processes, etc. For example, if the second version of document D was generated by a translation from the first version of the document in another language, then this translation is an activity.

2.3 Usage and Generation

Activities *generate* new entities. For example, writing a document brings the document into existence, while revising the document brings a new version into existence. Activities also make *use* of entities. For example, revising a document to fix spelling mistakes uses the original version of the document as well as a list of corrections. Generation does not always occur at the end of an activity, and an activity may generate entities part-way through. Likewise, usage does not always occur at the beginning of an activity.

2.4 Agents and Responsibility

An *agent* takes a role in an activity such that the agent can be assigned some degree of responsibility for the activity taking place. An agent can be a person, a piece of software, an inanimate object, an organization, or other entities that may be ascribed responsibility. When an agent has some responsibility for an activity, PROV says the agent was *associated* with the activity, where several agents may be associated with an activity and vice-versa. Consider a chart displaying some statistics regarding crime rates over time in a linear regression. To represent the provenance of that chart, we could state that the person who created the chart was an agent involved in its creation, and that the software used to create the chart was also an agent involved in that activity. An agent may be *acting on behalf* of others, e.g. an employee on behalf of their organization, and we can express such chains of responsibility in the provenance.

We can also describe that an entity is *attributed* to an agent to express the agent's responsibility for that entity, possibly along with other agents. This description can be understood as a shorthand for saying that the agent was responsible for the activity which generated the entity.

One may want to describe the provenance of an agent. For example, an organization responsible for the creation of a report may evolve over time as the report is written as some members leave and others join. To make provenance assertions about an agent in PROV, the agent must be declared explicitly both as an agent and as an entity.

2.5 Roles

A *role* is a description of the function or the part that an entity played in an activity. Roles specify the relationship between an entity and an activity, i.e. how the activity used or generated the entity. Roles also specify how agents are involved in an activity, qualifying their participation in the activity or specifying for what aspect of it each agent was responsible. For example, an agent may play the role of "editor" in an activity that uses one entity in the role of "document to be edited" and another in the role of "addition to be made to the document", to generate a further entity in the role of "edited document". Roles are application specific, so PROV does not define any particular roles.

2.6 Derivation and Revision

When one entity's existence, content, characteristics and so on are at least partly due to another entity, then we say that the former was *derived* from the latter. For example, one document may contain material copied from another, and a chart was derived from the data that it illustrates.

PROV allows some common, specialized kinds of derivation to be described. For example, a given entity, such as a document, may go through multiple *revisions* over time. Between revisions, one or more attributes of the entity may change. In PROV, the result of each revision is a new entity. PROV allows one to relate those entities by making a description that one was a revision of another. Another kind of derivation is to say that one entity, a quotation, *was quoted from* another entity, commonly a document.

2.7 Plans

Activities may follow pre-defined procedures, such as recipes, tutorials, instructions, or workflows. PROV refers to these, in general, as *plans*, and allows the description that a plan was followed, by agents, in executing an activity.

2.8 Time

Time is often a critical aspect of provenance. PROV allows the timing of significant events to be described, including when an entity was generated or used, or when an activity started and finished. For example, the model can be used to describe facts such as when a new version of a document was created (generation time), or when a document was edited (start and end of the editing activity).

2.9 Alternate Entities and Specialization

There is often more than one way to describe something in a record of provenance. Each perspective will be referred to by a separately identified entity, and PROV provides a mechanism for linking the different descriptions of the same thing together through the mechanism of *specialization*. One entity is a specialization of another entity if it shares the same fixed attributes, with the possible addition of further fixed attributes. This concept is best illustrated through a few use cases.

Entities can be mutable things. For example, a webpage is a single entity, W, despite being edited over time. Each version of the webpage is also an entity, W1, W2... To connect an individual version to the webpage in general, we say that the former is a specialization of the latter: W1 is a specialization of W, W2 is a specialization of W, and so on.

Two individuals may create provenance referring to the same thing from different perspectives. For example, the author of an article may attribute that article to themselves using PROV while, independently, a reader might quote a fact from that article elsewhere and document this in PROV. If the author later changes the fact, then from the reader's perspective there are now two versions of the article, and the reader had quoted from the version before the change. From the author's perspective, there is a single article, attributed to the author. If the author, the reader, or a third party, were to connect the two PROV records, that party would say that the article as referred to by the reader is a specialization of the same article as referred to by the author.

The above illustrates where we may want to connect entities by saying that they refer to the same thing, but at different levels of specialization. PROV also allows us to more generally draw a connection between two descriptions of the same thing, even if not at different levels of specialization, describing the entities as *alternates* of each other. For example, two versions of the webpage above, W1 and W2, are alternates of each other because they describe the same webpage.

As another example, if a file is copied from one directory to another to create a backup, we may say that the copies are alternate versions of the same, location-independent, file. Specifically, we may say that the file in the first directory, entity F1, is an alternate of the file in the second directory, entity F2. Note that it is the context (location) rather than content of the file that differs between the entities in this case.

3. Examples of Key Concepts in PROV

In the following sections, we show how PROV can be used to model provenance in a specific example scenario. Samples of PROV data are given. These samples use the namespace prefixes **prov**, denoting terms from the PROV ontology, and prefixes **exc**, **exn**, **exb**, **exg**, denoting terms specific to the example. We illustrate in these examples how PROV can be used in combination with other languages, such as FOAF [FOAF] and Dublin Core [DCTERMS] (with namespace prefix **foaf** and **dcterms** respectively).

The scenario describes a blogger exploring the provenance of an online newspaper article, including a chart produced from a government agency dataset. The provenance data comes from different sources: the blogger, the newspaper, the chart generator company and the government agency. The samples of provenance from each source use a different namespace prefix for identifiers that source has created: **exb**, **exn**, **exc**, and **exg** respectively.

The samples can be displayed in one or more of the following formats.

- [PROV-O] RDF triples, expressed using the [TURTLE] notation.
- [PROV-N] expressions.
- [PROV-XML] fragments.

Select the formats to display using the buttons below. Note that if all formats are hidden, the worked examples may not make sense!

| Hide Turtle Examples | | Hide PROV-N Examples | | Hide XML Examples |

3.1 Entities

An online newspaper publishes an article with a chart about crime statistics based on data (GovData) provided by a government portal. The article includes a chart based on the data, with data values composed (aggregated) by geographical regions.

A blogger, Betty, looking at the article, spots what she thinks to be an error in the chart. Betty retrieves a record of the provenance of the article, describing how it was created.

Betty finds the following descriptions of entities in the provenance.

Turtle Example *(hide all)*

```
exn:article      a prov:Entity ;
                 dcterms:title "Crime rises in cities" .
exg:dataset1     a prov:Entity .
exc:regionList   a prov:Entity .
exc:composition1 a prov:Entity .
exc:chart1       a prov:Entity .
```

PROV-N Example *(hide all)*

```
entity(exn:article, [dcterms:title="Crime rises in cities"])
entity(exg:dataset1)
entity(exc:regionList)
entity(exc:composition1)
entity(exc:chart1)
```

XML Example *(hide all)*

```
<prov:document>
  ...
  <prov:entity prov:id="exn:article">
    <dct:title>Crime rises in cities</dct:title>
  </prov:entity>
  <prov:entity prov:id="exg:dataset1"/>
  <prov:entity prov:id="exc:regionList1"/>
  <prov:entity prov:id="exc:composition1"/>
  <prov:entity prov:id="exc:chart1"/>
</prov:document>
```

These statements, in order, refer to the article (`exn:article`), an original data set (`exg:dataset1`), a list of regions (`exc:regionList`), data aggregated by region (`exc:composition1`), and a chart (`exc:chart1`), and state that each is an entity. Any entity may have attributes, such as the title of the article, expressed using `dcterms:title` above.

Notice the different namespace prefixes used: for the article it corresponds to the newspaper that published it (`exn`), and for the dataset it is the government namespace (`exg`). The `dcterms:title` namespace is taken from the Dublin Core vocabulary.

PROV data is commonly visualized for human consumption using [particular conventions](#), which we will introduce over the following sections. To start with, entities are denoted using ovals, as shown below.
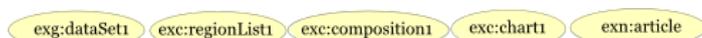
3.2 Activities

Further, the provenance describes that there was an activity (`exc:compile1`) denoting the compilation of the chart from the data set.

Turtle Example *(hide all)*

```
exc:compile1 a prov:Activity .
```

PROV-N Example *(hide all)*

```
activity(exc:compile1)
```

XML Example *(hide all)*

```
<prov:document>
  ...
  <prov:activity prov:id="exc:compile1"/>
</prov:document>
```

The provenance also includes reference to the more specific steps involved in this compilation, which are first composing the data by region (`exc:compose1`) and then generating the chart graphic (`exc:illustrate1`).

Turtle Example *(hide all)*

```
exc:compose1    a prov:Activity .
exc:illustrate1 a prov:Activity .
```

PROV-N Example *(hide all)*

```
activity(exc:compose1)
activity(exc:illustrate1)
```

XML Example *(hide all)*

```
<prov:document>
  ...
  <prov:activity prov:id="exc:compose1"/>
  <prov:activity prov:id="exc:illustrate1"/>
</prov:document>
```

In visualizations of the PROV data, activities are depicted as rectangles, as below.

exc:compose1 exc:illustrate1 exc:compile1

3.3 Usage and Generation

Concluding the basic description of what occurred, the provenance describes the key relations among the above entities and activities, i.e. the usage of an entity by an activity, or the generation of an entity by an activity.

For example, the descriptions below state that the composition activity (`exc:compose1`) used the original data set, that it used the list of regions, and that the composed data was generated by this activity.

Turtle Example *(hide all)*

```
exc:compose1      prov:used           exg:dataset1 ;
                  prov:used           exc:regionList1 .
exc:composition1  prov:wasGeneratedBy exc:compose1 .
```

PROV-N Example *(hide all)*

```
used(exc:compose1, exg:dataset1, -)
used(exc:compose1, exc:regionList1, -)
wasGeneratedBy(exc:composition1, exc:compose1, -)
```

Note that the - argument in the examples above denote unspecified optional information. See the [PROV-N] specification for the details of what arguments may be expressed in each PROV-N statement.

XML Example *(hide all)*

```
<prov:document>
  ...
  <prov:used>
    <prov:activity prov:ref="exc:compose1"/>
    <prov:entity prov:ref="exg:dataset1"/>
  </prov:used>
  <prov:used>
    <prov:activity prov:ref="exc:compose1"/>
    <prov:entity prov:ref="exc:regionList1"/>
  </prov:used>
  <prov:wasGeneratedBy>
    <prov:entity prov:ref="exc:composition1"/>
    <prov:activity prov:ref="exc:compose1"/>
  </prov:wasGeneratedBy>
</prov:document>
```

Similarly, the chart graphic creation activity (`exc:illustrate1`) used the composed data, and the chart was generated by this activity.

Turtle Example *(hide all)*

```
exc:illustrate1 prov:used           exc:composition1 .
exc:chart1      prov:wasGeneratedBy exc:illustrate1 .
```

PROV-N Example *(hide all)*

```
used(exc:illustrate1, exc:composition1, -)
wasGeneratedBy(exc:chart1, exc:illustrate1, -)
```

XML Example *(hide all)*

```
<prov:document>
  ...
  <prov:used>
    <prov:activity prov:ref="exc:illustrate1"/>
    <prov:entity prov:ref="exc:composition1"/>
  </prov:used>
```
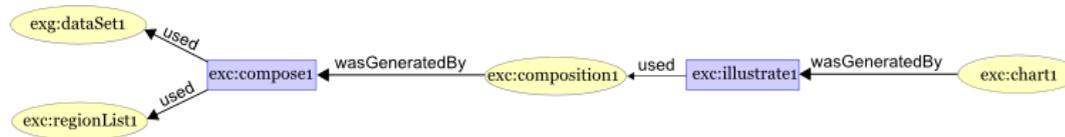
```
    <prov:wasGeneratedBy>
      <prov:entity prov:ref="exc:chart1"/>
      <prov:activity prov:ref="exc:illustrate1"/>
    </prov:wasGeneratedBy>
  </prov:document>
```

In visualizing the PROV data, usage and generation are connections between entities and activities. The arrows point from the future to the past.

3.4 Agents and Responsibility

Digging deeper, Betty wants to know who compiled the chart. Betty sees that Derek was involved in both the composition and chart creation activities:

Turtle Example *(hide all)*

```
    exc:compose1    prov:wasAssociatedWith exc:derek .
    exc:illustrate1 prov:wasAssociatedWith exc:derek .
```

PROV-N Example *(hide all)*

```
    wasAssociatedWith(exc:compose1, exc:derek, -)
    wasAssociatedWith(exc:illustrate1, exc:derek, -)
```

XML Example *(hide all)*

```
  <prov:document>
    ...
    <prov:wasAssociatedWith>
      <prov:activity prov:ref="exc:compose1"/>
      <prov:agent prov:ref="exc:derek"/>
    </prov:wasAssociatedWith>
    <prov:wasAssociatedWith>
      <prov:activity prov:ref="exc:illustrate1"/>
      <prov:agent prov:ref="exc:derek"/>
    </prov:wasAssociatedWith>
  </prov:document>
```

The record for Derek provides the following description that Derek is an agent, specifically a person, followed by non-PROV information giving attributes of Derek.

Turtle Example *(hide all)*

```
    exc:derek a prov:Agent ;
            a prov:Person ;
            foaf:givenName "Derek"^^xsd:string ;
            foaf:mbox      <mailto:derek@example.org> .
```

PROV-N Example *(hide all)*

```
    agent(exc:derek,
        [prov:type='prov:Person', foaf:givenName="Derek",
         foaf:mbox="<mailto:derek@example.org>"])
```

XML Example *(hide all)*

```
  <prov:document>
    ...
    <prov:agent prov:id="exc:derek">
      <prov:type>prov:Person</prov:type>
      <foaf:givenName>Derek</foaf:givenName>
      <foaf:mbox>mailto:derek@example.org</foaf:mbox>
    </prov:agent>
  </prov:document>
```

Derek works as part of an organization, Chart Generators Inc, and so the provenance declares that he acts on their behalf. Note that the organization is itself an agent. The namespace prefix used by the organization is `exc`.

Turtle Example *(hide all)*

```
    exc:derek prov:actedOnBehalfOf exc:chartgen .
    exc:chartgen a prov:Agent ;
               a prov:Organization ;
               foaf:name "Chart Generators Inc" .
```

PROV-N Example *(hide all)*

```
agent(exc:chartgen,
      [prov:type='prov:Organization',
       foaf:name="Chart Generators Inc"])
actedOnBehalfOf(exc:derek, exc:chartgen)
```

XML Example *(hide all)*

```
<prov:document>
  ...
  <prov:agent prov:id="exc:chartgen">
    <prov:type>prov:Organization</prov:type>
    <foaf:name>Chart Generators Inc</foaf:name>
  </prov:agent>
  <prov:actedOnBehalfOf>
    <prov:delegate prov:ref="exc:derek"/>
    <prov:responsible prov:ref="exc:chartgen"/>
  </prov:actedOnBehalfOf>
</prov:document>
```

It would also be possible to express the more specific statement that Derek worked on the organization's behalf for a particular activity, rather than in general, and so may have acted on behalf of other organizations for other activities. See the PROV specifications for details on how to express activity-specific delegation.

Finally, there is an explicit statement in the provenance that the chart was attributed to Derek.

Turtle Example *(hide all)*

```
exc:chart1 prov:wasAttributedTo exc:derek .
```

PROV-N Example *(hide all)*

```
wasAttributedTo(exc:chart1, exc:derek)
```

XML Example *(hide all)*

```
<prov:document>
  ...
  <prov:wasAttributedTo>
    <prov:entity prov:ref="exc:chart1"/>
    <prov:agent prov:ref="exc:derek"/>
  </prov:wasAttributedTo>
</prov:document>
```

We can extend our graphical depiction to show the agents, association and attribution links.
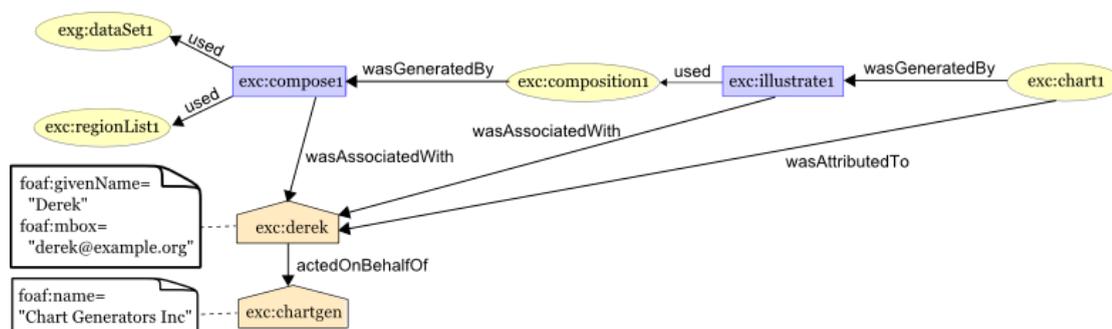
3.5 Roles

For Betty to understand where the error lies, she needs to have more detailed information on how entities have been used in and generated by activities. Betty has determined that `exc:compose1` used entities `exc:regionList1` and `exg:dataset1`, but she does not know what function these entities played in the processing. Betty also knows that `exc:derek` was associated with the activities, but she does not know if Derek was the analyst responsible for determining how the data should be composed.

The above information is described as roles in the provenance. The composition activity involved entities in four roles: the data to be composed (`exc:dataToCompose`), the regions to aggregate by (`exc:regionsToAggregateBy`), the resulting composed data (`exc:composedData`), and the analyst doing the composition (`exc:analyst`).

Turtle Example *(hide all)*

```
exc:dataToCompose        a prov:Role .
exc:regionsToAggregateBy a prov:Role .
exc:composedData         a prov:Role .
exc:analyst              a prov:Role .
```

Examples in the sections above show descriptions of the simple facts that the composition activity used, generated and was enacted by entities/agents. For example, the usage of the data set by the compose activity is expressed as follows.

Turtle Example *(hide all)*

```
exc:compose1 prov:used exg:dataset1 .
```

The provenance can contain more details of exactly how these entities and agents were involved in the activity. To express this, PROV-O refers to *qualified usage*, *qualified generation*, etc., which are descriptions consisting of several statements about how usage, generation, etc. took place. For example, we may describe the plan followed by an agent in performing an activity, or the time at which an activity generated an entity, both illustrated later. Another example of qualified involvement is the role an entity played in an activity. The descriptions below state that the composition activity (exc:compose1) included the usage of the government data set (exg:dataset1) in the role of the data to be composed (exc:dataToCompose).

Turtle Example *(hide all)*

```
exc:compose1 prov:qualifiedUsage [
            a prov:Usage ;
            prov:entity  exg:dataset1 ;
            prov:hadRole exc:dataToCompose
      ] .
```

In PROV-N, the role is expressed as one of the list of attributes in the used expression, with the attribute name prov:role.

PROV-N Example *(hide all)*

```
used(exc:compose1, exg:dataset1, -, [prov:role='exc:dataToCompose'])
```

XML Example *(hide all)*

```
<prov:document>
  ...
  <prov:used>
    <prov:activity prov:ref="exc:compose1"/>
    <prov:entity prov:ref="exg:dataset1"/>
    <prov:role>exc:dataToCompose</prov:role>
  </prov:used>
</prov:document>
```

This can then be distinguished from the same activity's usage of the list of regions because the roles played are different.

Turtle Example *(hide all)*

```
exc:compose1 prov:qualifiedUsage [
            a prov:Usage ;
            prov:entity  exc:regionList1 ;
            prov:hadRole exc:regionsToAggregateBy
      ] .
```

PROV-N Example *(hide all)*

```
used(exc:compose1, exc:regionList1, -, [prov:role='exc:regionsToAggregateBy'])
```

XML Example *(hide all)*

```
<prov:document>
  ...
  <prov:used>
    <prov:activity prov:ref="exc:compose1"/>
    <prov:entity prov:ref="exc:regionList1"/>
    <prov:role>exc:regionsToAggregateBy</prov:role>
  </prov:used>
</prov:document>
```

Similarly, the provenance includes descriptions that the same activity was enacted in a particular way by Derek, so it indicates that he had the role of exc:analyst, and that the entity exc:composition1 took the role of the composed data in what the activity generated:

Turtle Example *(hide all)*

```
exc:compose1 prov:qualifiedAssociation [
            a  prov:Association ;
            prov:agent    exc:derek ;
            prov:hadRole  exc:analyst
] .
exc:composition1 prov:qualifiedGeneration [
            a prov:Generation ;
            prov:activity  exc:compose1 ;
            prov:hadRole   exc:composedData
] .
```

PROV-N Example *(hide all)*

```
wasAssociatedWith(exc:compose1, exc:derek, -, [prov:role='exc:analyst'])
wasGeneratedBy(exc:composition1, exc:compose1, -, [prov:role='exc:composedData'])
```

XML Example *(hide all)*

```
<prov:document>
  ...
  <prov:wasAssociatedWith>
    <prov:activity prov:ref="exc:compose1"/>
    <prov:agent prov:ref="exc:derek"/>
    <prov:role>exc:analyst</prov:role>
  </prov:wasAssociatedWith>
  <prov:wasGeneratedBy>
    <prov:entity prov:ref="exc:composition1"/>
    <prov:activity prov:ref="exc:compose1"/>
    <prov:role>exc:composedData</prov:role>
  </prov:wasGeneratedBy>
</prov:document>
```
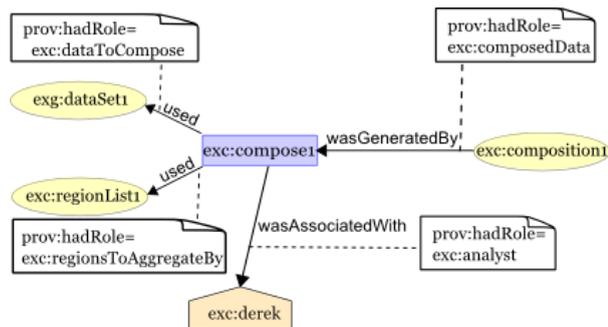
Depicting the above visually, we have the following.

3.6 Derivation and Revision

After looking at the detail of the compilation activity, there appears to be nothing wrong, so Betty concludes the error is in the government dataset. She looks at the dataset `exg:dataset1`, and sees that it is missing data from one of the zipcodes in the area. She contacts the government agency, and a new version of GovData is created, declared to be the next revision of the data. The provenance of this new dataset, `exg:dataset2`, states that it is a revision of the old data set, `exg:dataset1`.

Turtle Example *(hide all)*

```
exg:dataset2 a prov:Entity ;
            prov:wasRevisionOf exg:dataset1 .
```

PROV-N Example *(hide all)*

```
entity(exg:dataset2)
wasDerivedFrom(exg:dataset2, exg:dataset1, [prov:type='prov:Revision'])
```

XML Example *(hide all)*

```
<prov:document>
  ...
  <entity prov:id="dataSet2"/>
  <prov:wasDerivedFrom>
    <prov:generatedEntity prov:ref="exg:dataset2"/>
    <prov:usedEntity prov:ref="exg:dataset1"/>
    <prov:type>prov:Revision</prov:type>
  </prov:wasDerivedFrom>
</prov:document>
```

Derek notices that there is a new dataset available and creates a new chart based on the revised data, using another compilation activity. Betty checks the article again at a later point, and wants to know if it is based on the old or new GovData. She sees a new description stating that the new chart is derived from the new dataset (the same relation could be expressed between the old chart and old dataset).

Turtle Example *(hide all)*

```
exc:chart2 a prov:Entity ;
            prov:wasDerivedFrom exg:dataset2 .
```

PROV-N Example *(hide all)*

```
wasDerivedFrom(exc:chart2, exg:dataset2)
```

XML Example *(hide all)*

```
<prov:document>
  ...
  <prov:wasDerivedFrom>
```

```
      <prov:generatedEntity prov:ref="exc:chart2"/>
      <prov:usedEntity prov:ref="exg:dataset2"/>
   </prov:wasDerivedFrom>
</prov:document>
```

and that the new chart is a revision of the original one:

Turtle Example *(hide all)*

```
exc:chart2 a prov:Entity ;
        prov:wasRevisionOf exc:chart1 .
```

PROV-N Example *(hide all)*

```
entity(exc:chart2)
wasDerivedFrom(exc:chart2, exc:chart1, [prov:type='prov:Revision'])
```

XML Example *(hide all)*

```
<prov:document>
  ...
  <entity prov:id="chart2"/>
  <prov:wasDerivedFrom>
    <prov:generatedEntity prov:ref="exc:chart2"/>
    <prov:usedEntity prov:ref="exc:chart1"/>
    <prov:type>prov:Revision</prov:type>
  </prov:wasDerivedFrom>
</prov:document>
```

Derivation and revision are connections between entities, and so depicted with arrows in our visualization.
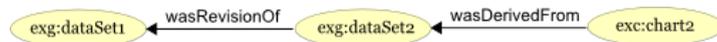
3.7 Plans

Betty then wishes to know whether the new data set correctly addresses the error that existed before. The provenance of the new dataset, exg:dataset2, describes not only who performed the corrections, Edith, but also what instructions she followed in doing so (in PROV terms, the plan). First, the correction activity (exg:correct1), the person who corrected it, Edith (exg:edith), and the correction instructions (exg:instructions) are described.

Turtle Example *(hide all)*

```
exg:correct1     a prov:Activity .
exg:edith        a prov:Agent, prov:Person .
exg:instructions a prov:Plan .
```

PROV-N Example *(hide all)*

```
activity(exg:correct1)
agent(exg:edith, [prov:type='prov:Person'])
entity(exg:instructions)
```

XML Example *(hide all)*

```
<prov:document>
  ...
  <prov:activity prov:id="exg:correct1"/>
  <prov:agent prov:id="exg:edith">
    <prov:type>prov:Person</prov:type>
  </prov:agent>
  <prov:entity prov:id="exg:instructions">
    <prov:type>prov:Plan</prov:type>
  </prov:entity>
</prov:document>
```

The connection between them is expressed in PROV-O using a qualified association giving details of how Edith was associated with the correction activity, including that she followed the above correction instructions.

Turtle Example *(hide all)*

```
exg:correct1 prov:qualifiedAssociation [
        a Association ;
        prov:agent   exg:edith ;
        prov:hadPlan exg:instructions
] .
exg:dataset2 prov:wasGeneratedBy exg:correct1 .
```

In PROV-N, the plan is an optional parameter to wasAssociatedWith descriptions.

PROV-N Example *(hide all)*

```
wasAssociatedWith(exg:correct1, exg:edith, exg:instructions)
wasGeneratedBy(exg:dataset2, exg:correct1, -)
```

In PROV-XML, the plan is an optional child of the wasAssociatedWith element.

XML Example *(hide all)*

```xml
<prov:document>
  ...
  <prov:wasAssociatedWith>
    <prov:activity prov:ref="exg:correct1"/>
    <prov:agent prov:ref="exg:edith"/>
    <prov:plan prov:ref="exg:instructions"/>
  </prov:wasAssociatedWith>
  <prov:wasGeneratedBy>
    <prov:entity prov:ref="exg:dataset2"/>
    <prov:activity prov:ref="exg:correct1"/>
  </prov:wasGeneratedBy>
</prov:document>
```

Plans are additional information about the connection from an activity to an agent, and so, in our visualization, connect to the link between them.
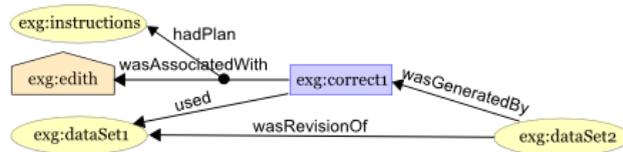
3.8 Time

The government agency that produced GovData is concerned to know how long the incorrect chart was in circulation before the corrected chart was created. That is, they wish to compare the times at which the original and the corrected charts were generated. The snippet below shows that the second chart was generated roughly a month after the first.

Turtle Example *(hide all)*

```
exc:chart1 prov:generatedAtTime "2012-03-02T10:30:00"^^xsd:dateTime .
exc:chart2 prov:generatedAtTime "2012-04-01T15:21:00"^^xsd:dateTime .
```

PROV-N Example *(hide all)*

```
wasGeneratedBy(exc:chart1, exc:compile1,  2012-03-02T10:30:00)
wasGeneratedBy(exc:chart2, exc:compile2, 2012-04-01T15:21:00)
```

XML Example *(hide all)*

```xml
<prov:document>
  ...
  <prov:wasGeneratedBy>
    <prov:entity prov:ref="exc:chart1"/>
    <prov:time>2012-03-02T10:30:00</prov:time>
  </prov:wasGeneratedBy>
  <prov:wasGeneratedBy>
    <prov:entity prov:ref="exc:chart2"/>
    <prov:time>2012-04-01T15:21:00</prov:time>
  </prov:wasGeneratedBy>
</prov:document>
```

To ensure their procedures are efficient, the agency also wishes to know how long the corrections took once the error was discovered. That is, they wish to know the start and end times of the correction activity (exg:correct1). These details are expressed as follows, showing that the corrections took a little over a day.

Turtle Example *(hide all)*

```
exg:correct1 prov:startedAtTime "2012-03-31T09:21:00"^^xsd:dateTime ;
             prov:endedAtTime   "2012-04-01T15:21:00"^^xsd:dateTime .
```

PROV-N Example *(hide all)*

```
activity(exg:correct1, 2012-03-31T09:21:00, 2012-04-01T15:21:00)
```

XML Example *(hide all)*
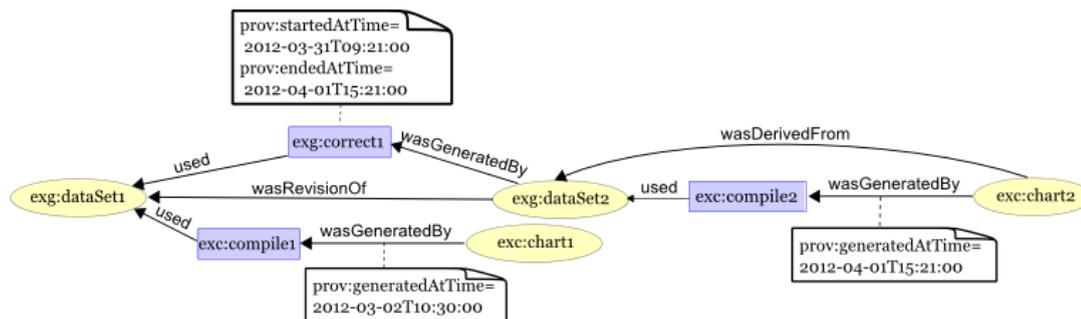
```xml
<prov:document>
  ...
  <prov:activity prov:id="exg:correct1">
    <prov:startTime>2012-03-31T09:21:00</prov:startTime>
    <prov:endTime>2012-04-01T15:21:00</prov:endTime>
  </prov:activity>
```

```
                  </prov:document>
```

Time is visualized as additional information regarding activities or the links between activities and entities or agents.

3.9 Alternate Entities and Specialization

Before noticing anything wrong with the government data, Betty had already posted a blog entry about the article. The blog entry had its own published provenance. In particular, it contains some text copied from the article, and the provenance states that this text (exb:quoteInBlogEntry-20130326) is quoted from the article. Note the namespace prefix (exb) used for the blog.

Turtle Example *(hide all)*

```
exb:quoteInBlogEntry-20130326 a prov:Entity ;
                              prov:value "Smaller cities have more crime than larger ones" ;
                              prov:wasQuotedFrom exn:article .
```

PROV-N Example *(hide all)*

```
entity(exb:quoteInBlogEntry-20130326, prov:value="Smaller cities have more crime than larger ones")
wasDerivedFrom(exb:quoteInBlogEntry-20130326, exn:article, [prov:type='prov:Quotation'])
```

XML Example *(hide all)*

```
<prov:document>
  ...
  <prov:entity prov:id="exb:BlogEntry-20130326">
    <prov:value>Smaller cities have more crime than larger ones</prov:value>
  </prov:entity>
  <prov:wasDerivedFrom>
    <prov:generatedEntity prov:ref="exb:quoteInBlogEntry-20130326"/>
    <prov:usedEntity prov:ref="exn:article"/>
    <prov:type>prov:Quotation</prov:type>
  </prov:wasDerivedFrom>
</prov:document>
```

The newspaper, from past experience, anticipated that there could be revisions to the article, and so created identifiers for both the article in general (exn:article) as a URI that got redirected to the first version of the article (exn:articleV1), allowing both to be referred to as entities in provenance data. In the provenance records, the newspaper describes the connection between the two: that the first version of the article is a specialization of the article in general.

Turtle Example *(hide all)*

```
exn:articleV1 a prov:Entity ;
             prov:specializationOf exn:article .
```

PROV-N Example *(hide all)*

```
entity(exn:articleV1)
specializationOf(exn:articleV1, exn:article)
```

XML Example *(hide all)*

```
<prov:document>
  ...
  <prov:entity prov:id="exn:articleV1"/>
  <prov:specializationOf>
    <prov:specificEntity prov:ref="exn:articleV1"/>
    <prov:generalEntity prov:ref="exn:article"/>
  </prov:specializationOf>
</prov:document>
```

Later, after the data set is corrected and the new chart generated, a new version of the article is created, exn:articleV2, with its own URI where the article is redirected to. To ensure that those consulting the provenance of exn:articleV2 understand that it is connected with the provenance of exn:article and exn:articleV1, the newspaper describes how these entities are related.

Turtle Example *(hide all)*

```
exn:articleV2 prov:specializationOf exn:article .
exn:articleV2 prov:alternateOf      exn:articleV1
```

PROV-N Example *(hide all)*

```
specializationOf(exn:articleV2, exn:article)
alternateOf(exn:articleV2, exn:articleV1)
```
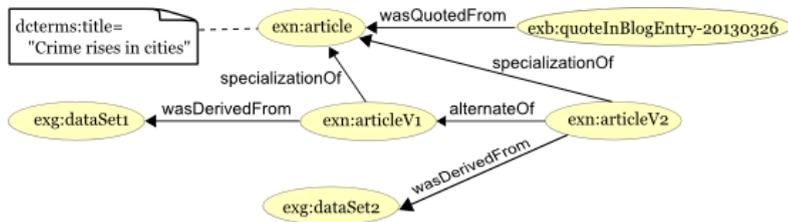
XML Example *(hide all)*

```xml
<prov:document>
  ...
  <prov:specializationOf>
    <prov:specificEntity prov:ref="exn:articleV2"/>
    <prov:generalEntity prov:ref="exn:article"/>
  </prov:specializationOf>
  <prov:alternateOf>
    <prov:alternate1 prov:ref="exn:articleV1"/>
    <prov:alternate2 prov:ref="exn:articleV2"/>
  </prov:alternateOf>
</prov:document>
```

Note that above we could have also stated that `exn:articleV2` was a revision of `exn:articleV1`, as we did between `exc:chart2` and `exc:chart1`, which would describe more concretely how the alternate entities are related. Specialization and alternate relations connect entities, and so are visualized as links between them.

3.10 The Complete Example

The full set of examples shown above are available in PROV-O Turtle format or PROV-N format or PROV-XML format. We visualize the whole example below.
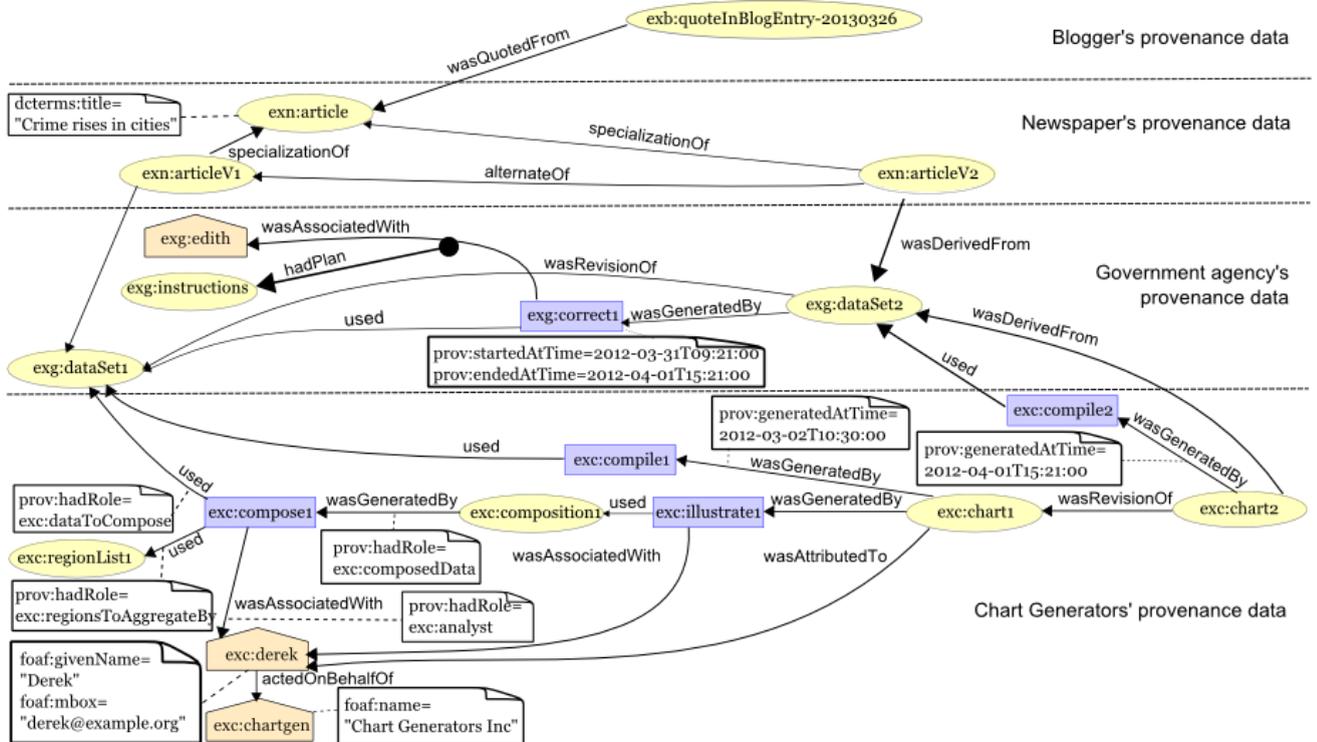
4. Summary

The examples in the previous section highlight the following important capabilities of PROV:

1. **Representing diverse entities involved in provenance**. The examples illustrate that provenance records can contain different kinds of information:
   - The newspaper took an object-centered view on their provenance statements, stating what objects were derived from others but not stating what processes were used or what entities were involved.
   - Betty the blogger took a very agent-centered view on provenance. She chose to associate her blog post with herself, the

        government agency that published the data, and the newspaper that published the article.
- ○ Derek took a process-centered perspective and combined it with object-centered and agent-centered provenance information. The processes followed to derive the charts from the government datasets were clearly. All intermediate and final documents were stated. His involvement in the various activities was clearly recorded, and the resulting documents were attributed to him.

2. **Stating partial or incomplete provenance**. The newspaper omitted information about their specific editorial processes in publishing articles. The chart generator said nothing about the software used, therefore omitting information about other agents involved in the processes.
3. **Integrating provenance-related information represented in other vocabularies**. The FOAF vocabulary was used for specifying details about Derek and his company. The Dublin Core vocabulary was used for specifying the article title.
4. **Combining distributed provenance records provided by different parties**. The provenance records described in this example came from a government source, a newspaper, a company, and a blogger. Using URIs and namespaces, a provenance record can draw from multiple sources on the Web.
5. **Describing the commonalities in the derivation of two entities**. The two versions of the charts were both derived through a similar compile activity.
6. **Relating versions of objects over time**. The second dataset was described as a revision of the first one.
7. **Stating the provenance of provenance assertions**. Betty could create a provenance record for her blog post, and then assert the provenance of that provenance record.
8. **Enabling alternative accounts for the provenance of an entity**. The blog post could have a provenance account from Betty as its author and another one from the newspaper that the blog quoted from.
9. **Supporting queries over provenance of different granularity**. The activity of composing the data from different regions was described in greater detail than the other steps to produce the chart. The provenance of the chart has much finer-grained detail than the provenance of the government dataset.
10. **Supporting reasoning and inference over provenance records**. The provenance assertions explicitly stated above can be used to infer additional assertions that are true according to the PROV model. For example, because Derek created the chart acting on behalf of Chart Generators, a PROV inference may be that the chart can be attributed to Chart Generators.

The examples also illustrate how PROV enables many uses of provenance, such as checking for possible errors, tracking down parties responsible for fixing them, repeating processes to re-generate datasets, relating versions of documents over time, and inspecting sources and attribution.

There are additional aspects of the PROV data model that we have not covered in this primer and that are important in many uses of provenance. These include the following.

1. **Provenance of provenance**: It is often useful to state the provenance of a provenance record. To enable this, PROV allows the grouping of provenance assertions as a *bundle*. A bundle is a type of entity, and therefore its provenance can be asserted in turn. For example, supposed that Betty wanted to express in the provenance of her blog entry that she personally checked her sources. She would want to state the provenance of her blog entry, and then state that she stands by that provenance. To do this, she would create a bundle that contains provenance assertions for her blog entry, and then she would state that she was the creator of that bundle.
2. **Collections of entities**: A *collection* in PROV is a type of entity that has other entities as members. For example, Derek could decide to create several types of charts using dataset2. He would describe all the charts as a collection, and then state the provenance of the collection to be a single step that generates all the charts.

Please see the [PROV-DM] specification for details on these and other concepts.

A. Acknowledgements

B. Changes Since Fourth Public Working Draft

- Corrected XML example for activities
- Clarified usage of adoptedOnBehalfOf relation
- Clarified wasQuotedFrom example
- Adapted PROV-XML examples to clarify they are not RDF/XML
- Clarified varying sources of example provenance data with different namespace prefixes, and in division of complete example provenance graph
- Used instance-specific identifiers for entities, activities and agents
- Added section summarising some additional aspects of PROV not covered in depth
- Added summary section
- Added proper acknowledgements
- Clarified the introduction regarding what is in each section.
- Added references for FOAF, Dublin Core and newer PROV documents.
- Typos fixed and minor clarifications throughout

C. References

C.1 Informative references

[DCTERMS]
Dublin Core Metadata Initiative. *DCMI Metadata Terms*. 14 June 2012. URL: http://dublincore.org/documents/2012/06/14/dcmi-terms/

[FOAF]
Dan Brickley, Libby Miller; eds. *FOAF Vocabulary Specification 0.98*. 9 August 2010. URL: http://xmlns.com/foaf/spec/20100809.html

[PROV-AQ]
Graham Klyne; Paul Groth; eds. *Provenance Access and Query*. 30 April 2013, W3C Note. URL: http://www.w3.org/TR/2013/NOTE-prov-aq-20130430/

[PROV-CONSTRAINTS]
James Cheney; Paolo Missier; Luc Moreau; eds. *Constraints of the PROV Data Model*. 30 April 2013, W3C Recommendation. URL: http://www.w3.org/TR/2013/REC-prov-constraints-20130430/

[PROV-DC]
Daniel Garijo; Kai Eckert; eds. *Dublin Core to PROV Mapping*. 30 April 2013, W3C Note. URL: http://www.w3.org/TR/2013/NOTE-prov-dc-20130430/

[PROV-DICTIONARY]
Tom De Nies; Sam Coppens; eds. *PROV Dictionary: Modeling Provenance for Dictionary Data Structures*. 30 April 2013, W3C Note. URL: http://www.w3.org/TR/2013/NOTE-prov-dictionary-20130430/

[PROV-DM]
Luc Moreau; Paolo Missier; eds. *PROV-DM: The PROV Data Model*. 30 April 2013, W3C Recommendation. URL: http://www.w3.org/TR/2013/REC-prov-dm-20130430/

[PROV-LINKS]
Luc Moreau; Timothy Lebo; eds. *Linking Across Provenance Bundles*. 30 April 2013, W3C Note. URL: http://www.w3.org/TR/2013/NOTE-prov-links-20130430/

[PROV-N]
Luc Moreau; Paolo Missier; eds. *PROV-N: The Provenance Notation*. 30 April 2013, W3C Recommendation. URL: http://www.w3.org/TR/2013/REC-prov-n-20130430/

[PROV-O]
Timothy Lebo; Satya Sahoo; Deborah McGuinness; eds. *PROV-O: The PROV Ontology*. 30 April 2013, W3C Recommendation. URL: http://www.w3.org/TR/2013/REC-prov-o-20130430/

[PROV-OVERVIEW]
Paul Groth; Luc Moreau; eds. *PROV-OVERVIEW: An Overview of the PROV Family of Documents*. 30 April 2013, W3C Note. URL: http://www.w3.org/TR/2013/NOTE-prov-overview-20130430/

[PROV-SEM]
James Cheney; ed. *Semantics of the PROV Data Model*. 30 April 2013, W3C Note. URL: http://www.w3.org/TR/2013/NOTE-prov-sem-20130430.

[PROV-XML]
Hook Hua; Curt Tilmes; Stephan Zednik; eds. *PROV-XML: The PROV XML Schema*. 30 April 2013, W3C Note. URL: http://www.w3.org/TR/2013/NOTE-prov-xml-20130430/

[PROVENANCE-XG]
Yolanda Gil, James Cheney, Paul Groth, Olaf Hartig, Simon Miles, Luc Moreau, Paulo Pinheiro da Silva. *Provenance XG Final Report*. 8 December 2010. URL: http://www.w3.org/2005/Incubator/prov/XGR-prov-20101214/

[TURTLE]
Eric Prud'hommeaux, Gavin Carothers; eds. *Turtle: Terse RDF Triple Language*. 9 August 2011. W3C Working Draft. URL: http://www.w3.org/TR/2011/WD-turtle-20110809/