PROVENANCE & e-Science

Ammar Benabadelkader
Provenance Taskforce
Budapest, 24 March 2014
Outlines

- Background
- What is Provenance?
- Provenance for e-Science
- PROV Concepts
- A walkthrough PROV
  - Using an example
- PROV usage and Applications
- Work Progress
- Discussion
Wikipedia: Provenance, from the French provenir, "to come from", means the origin, or the source of something, or the history of the ownership or location of an object.

La Bella Principessa, a recent rediscovery said to be by Leonardo da Vinci, whose provenance is still the subject of research and controversy.

A note dated November 12, 1883 explaining the provenance of a painting sold to Baroness Villani, who ships it painting to the US to sell it

(http://www.3pipe.net/2012/02/search-for-truth-and-clarity.html)
• Provenance is 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*.

• Provenance plays many *roles*, it applies to many different kind of *information*, and it is intended for different *uses*.

It is metadata which can be viewed differently from one application to another.
Provenance for e-science

• Why do the scientists take provenance into account?
  – to **understand** how data and results were generated,
  – to establish **credibility** and **trust** in their publications,
  – to **verify** data for proves,
  – to **analyze** and **correlate** results of related experiments,
  – to **debug**, **rectify** or **improve** their methodology, ...
Provenance for e-science

Reliability & quality
- Trust the source or the process that lead to the object
- Trust at one point in time and during the entire (processing) life

Justification & Audit
- Accurate records of the sources and methods according to those published.

Ownership & security
- As objects migrate, so must their provenance

Reusability & reproducibility
- Possibility for others to repeat and validate the experiment
- Only possible under similar conditions

Change & evolution
- Changes in underlying data may lead to invalid annotations

Versioning
- As objects version, so must their provenance
PROV enables to **represent** and **interchange** provenance information using widely available formats such as RDF and XML.

- The PROV defines a model, corresponding serializations, and other supporting definitions to enable the inter-operable interchange of provenance information in heterogeneous environments such as the Web.

- PROV is a succession of OPM (Open Provenance Model)
The Organization of PROV
# PROV Family of Documents

**Note**  
PROV-PRIMER is the entry point to PROV offering an introduction to the provenance data model. This is where you should start and for many may be the only document needed.

**Rec**  
PROV-O defines a light-weight OWL2 ontology for the provenance data model. This is intended for the Linked Data and Semantic Web community.

**Note**  
PROV-XML defines an XML schema for the provenance data model. This is intended for developers who need a native XML serialization of the PROV data model.

**Rec**  
PROV-DM defines a conceptual data model for provenance including UML diagrams. PROV-O, PROV-XML and PROV-N are serializations of this conceptual model.

**Rec**  
PROV-N defines a human-readable notation for the provenance model. This is used to provide examples within the conceptual model as well as used in the definition of PROV-CONSTRAINTS.

**Rec**  
PROV-CONSTRAINTS defines a set of constraints on the PROV data model that specifies a notion of valid provenance. It is specifically aimed at the implementors of validators.

**Note**  
PROV-AQ defines how to use Web-based mechanisms to locate and retrieve provenance information.

**Note**  
PROV-DC defines a mapping between Dublin Core and PROV-O.

**Note**  
PROV-DICTIONARY defines constructs for expressing the provenance of dictionary style data structures.

**Note**  
PROV-SEM defines a declarative specification in terms of first-order logic of the PROV data model.

**Note**  
PROV-LINKS defines extensions to PROV to enable linking provenance information across bundles of provenance descriptions.
Entities, activities and agents are represented as **nodes**, with oval, rectangular, and octagonal shapes, respectively.

Usage, Generation, Derivation, and Activity Association are represented as **directed edges**.
PROV Graph Layout Conventions

Arrangement

• Entities are laid out according to the ordering of their generation.
• Arrows point "back into the past"
**ENTITY**

**Entity**: a physical, digital, conceptual, or other kind of thing with some fixed aspects; entities may be real or imaginary.

```
Entity(id, [attr1=val1, ...])
Entity(Scan-101, [prov:type="T13D SAG", prov:format="DICOM"])
```

**PROV-XML**:
```
<prov:document>
  <prov:entity id="Scan-101">
    <prov:type>T13D SAG</prov:type>
    <prov:format>DICOM</prov:format>
  </prov:entity>
</prov:document>
```
**Activity**

**Activity**: Something that occurs over a period of time and acts upon or with entities.

```
Activity(id, startTime, endTime, [attr1=val1, ...])
Activity(Freesurfer, [prov:version= "5.0", prov:platform= "Lunix" ])
```

**PROV-XML**:

```
...  
<prov:activity id="Freesurfer">
  <prov:version>5.0</prov:version>
  <prov:platform>DICOM</prov:platform>
</prov:activity>
...
```
**AGENT**

**AGENT**: something that bears some form of responsibility for an activity taking place, for the existence of an entity, or for another agent's activity.

Agent(id, [attr1=val1, ...])

Agent(Simon, [prov:type= "Person", foaf:Name= "Simon", foaf:Age= "28" ])

Agent(KEBB, [prov:type= “Organization", foaf:Name= “AMC, Kebb"])

**PROV-O**:  

...  
Simon a prov:Agent ;  
a prov:Person ;  
foaf:Name "Simon"^^xsd:string ;  
foaf:Age "28"^^xsd:int .  
...
PROV is meant to describe how things were created or delivered, therefore, relations are named so they can be used in assertions about the past.
PROV is meant to describe how things were created or delivered, therefore, relations are named so they can be used in assertions about the past.
<table>
<thead>
<tr>
<th>Relation Name</th>
<th>influencee</th>
<th>influencer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Generation</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- wasGeneratedBy(id; e, a, t, attrs)</td>
<td>entity</td>
<td>activity</td>
</tr>
<tr>
<td><strong>Usage</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- used(id; a, e, t, attrs)</td>
<td>activity</td>
<td>entity</td>
</tr>
<tr>
<td><strong>Communication</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- wasInformedBy(id; a2, a1, attrs)</td>
<td>informed</td>
<td>informant</td>
</tr>
<tr>
<td><strong>Start</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- wasStartedBy(id; a2, e, a1, t, attrs)</td>
<td>activity</td>
<td>trigger</td>
</tr>
<tr>
<td><strong>End</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- wasEndedBy(id; a2, e, a1, t, attrs)</td>
<td>activity</td>
<td>trigger</td>
</tr>
<tr>
<td><strong>Invalidation</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- wasInvalidatedBy(id; e, a, t, attrs)</td>
<td>entity</td>
<td>activity</td>
</tr>
<tr>
<td><strong>Derivation</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- wasDerivedFrom(id; e2, e1, a, g2, u1, attrs)</td>
<td>generatedEntity</td>
<td>usedEntity</td>
</tr>
<tr>
<td><strong>Attribution</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- wasAttributedTo(id; e, ag, attrs)</td>
<td>entity</td>
<td>agent</td>
</tr>
<tr>
<td><strong>Association</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- wasAssociatedWith(id; a, ag, pl, attrs)</td>
<td>activity</td>
<td>agent</td>
</tr>
<tr>
<td><strong>Delegation</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- actedOnBehalfOf(id; ag2, ag1, a, attrs)</td>
<td>delegate</td>
<td>responsible</td>
</tr>
</tbody>
</table>
**A derivation:**

wasDerivedFrom(id; e2, e1, a, g2, u1, attrs)

- **id:** an OPTIONAL identifier for a derivation;
- **generatedEntity:** the identifier (e2) of the entity generated by the derivation;
- **usedEntity:** the identifier (e1) of the entity used by the derivation;
- **activity:** an OPTIONAL identifier (a) for the activity using and generating the above entities;
- **generation:** an OPTIONAL identifier (g2) for the generation involving the generated entity (e2) and activity (a);
- **usage:** an OPTIONAL identifier (u1) for the usage involving the used entity (e1) and activity (a);
- **attributes:** an OPTIONAL set (attrs) of attribute-value pairs representing additional information about this derivation.
PROV Constraints: Overview

Typing Constraints

wasAssociatedWith(id; a,ag,pl,attrs)

<table>
<thead>
<tr>
<th>a</th>
<th>'activity'</th>
</tr>
</thead>
<tbody>
<tr>
<td>ag</td>
<td>'agent'</td>
</tr>
<tr>
<td>pl</td>
<td>'entity'</td>
</tr>
</tbody>
</table>

**Constraint 29 (unique-endTime)**

IF activity(a2,_t1,t2,_attrs) and wasEndedBy(_end; a2,_e,_a1,t,_attrs1), THEN t2 = t

**Constraint 56 (membership-empty-collection)**

IF hadMember(c,e) and 'prov:EmptyCollection' ∈ typeOf(c) THEN INVALID.
Provenance@work

• **Three-fold Process:**

1. Implementing the **core structures** of the provenance information (**PROV-DM/PROV-CONSTRAINTS**) and associated generic **interfaces**

2. Provenance **Data Collector**

3. Implementing provenance **Data Usage/exploitation** tools: sharing, query, retrieval, automated on-demand materialized views, etc.
**PROV Core Implementation: Done**

**PROV-man**: a set of methods and interfaces to create and manipulate provenance information, including representation into XML, Graph, RDF, etc.

Implemented using:

- **SQL database**
  - Allows for remote and distributed access
  - Enforces data integrity (PROV-Constraints)

- **Hibernate**
  - Mapping of domain object to relational database
  - DBMS independent implementation

- **Java**: Portability and platform independent
PROV: Data Collection

• **What** kind of information/data to collect?
  • **Quality**: what kind of data to collect
  • **Quantity**: to what depth we should collect?

• **How** to collect the data
  • **Manual**: hard and error prone, due to data complexity
  • **Automatic**: time efficient and cost-effective

**AIM:**
Implement a data collector for WSPGADE/gUse environment
PROV: Extended Usage

- Provenance plays many *roles*, it applies to many different kind of *information*, and it is intended for different *uses*
- It is metadata which can be *viewed differently* from one application to another

**Extended usage:**
Provenance information can be used in combination with application specific data to perform some extended usage of provenance:
- E.g. reporting, visualization, analyses, web semantics, etc.
**PROV: other applications**

- **Safety/Security**: 
- **Privacy**: data not to be distributed
- **Accountability**: if something went wrong, who is accountable for?
- **Sharing of data**: 
  - incidental finding
  - regulations
  - level of confidence of the finding
- **Informing about the results of the research**: 
- **European regulation / right to be forgotten**: 
- **A solution to black-box software in e-science**
Useful Links

- **PROV-Primer**: [http://www.w3.org/TR/2013/NOTE-prov-primer-20130430/](http://www.w3.org/TR/2013/NOTE-prov-primer-20130430/)
- **PROV-DM**: [http://www.w3.org/TR/2013/REC-prov-dm-20130430/](http://www.w3.org/TR/2013/REC-prov-dm-20130430/)
- **PROV-O**: [http://www.w3.org/TR/2013/REC-prov-o-20130430/](http://www.w3.org/TR/2013/REC-prov-o-20130430/)
- **PROV-SEM**: [http://www.w3.org/TR/2013/NOTE-prov-sem-20130430/](http://www.w3.org/TR/2013/NOTE-prov-sem-20130430/)
- **Semantic Web**: [http://www.w3.org/standards/semanticweb/](http://www.w3.org/standards/semanticweb/)
- **World Wide Web Consortium (W3C)**: [http://www.w3.org](http://www.w3.org)
- **Resource Description Framework (RDF)**: [http://www.w3.org/TR/rdf-mt/](http://www.w3.org/TR/rdf-mt/)
Discussion / Questions