The Open Provenance Model
Towards inter-operability of Provenance Systems

Luc Moreau, University of Southampton
Juliana Freire, University of Utah,
Joe Futrelle, NCSA,
Robert E. McGrath, NCSA
Jim Myers, NCSA,
Patrick Paulson, PNNL
Overview

- Provenance Challenge 1
- Provenance Challenge 2
- OPM (non-)Requirements
- OPM Overview
- Formal Definition
- What next?
Provenance Challenge 1

- Idea came after IPAW’06 standardisation discussion
- Set up to be *informative* rather than *competitive*
- Aims to provide a forum for the community to understand the capabilities of different provenance systems and the expressiveness of their provenance representations
1. Find the process that led to Atlas X Graphic / everything that caused Atlas X Graphic to be as it is.

2. Find the process that led to Atlas X Graphic, excluding everything prior to the averaging of images with softmean.

3. Find the Stage 3, 4 and 5 details of the process that led to Atlas X Graphic.

4. Find all invocations of procedure align_warp using a twelfth order nonlinear 1365 parameter model that ran on a Monday.
Participating Teams

REDUX, MSR
Karma, Indiana U.
myGrid, U. of Manchester
Gridprovenance, Cardiff U.
Zoom, U. of Pennsylvania
DAKS, UC Davis
SDG, PNNL
UChicago, U. of Chicago
USC/ISI, ISI

MINDSWAP, U. of Maryland
JP, CESNET
VisTrails, U. of Utah
ES3, UCSB
RWS, UC Davis and SDSC
PASS, Harvard
NcsaD2k and NcsaCi, NCSA
PASOA, U. of Southampton
Representations
Challenge 1 Outcomes

- Challenge 1 Provenance questions and expected answers not precise enough
- Difficult to validate if results returned are correct or even comparable
- Challenge 2 aimed at establishing inter-operability of systems, by exchanging provenance information
Provenance Challenge 2

System 1

System 2

System 3
Participating Teams

<table>
<thead>
<tr>
<th>MyGrid U. of Manchester</th>
<th>PASOA, U. of Southampton</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDG, PNNL</td>
<td>MINDSWAP, U. of Maryland</td>
</tr>
<tr>
<td>Karma, Indiana U.</td>
<td>Lineage for JOpera, ETH Zurich</td>
</tr>
<tr>
<td>OntoGrid, OntoGrid project</td>
<td>CESNET, CESNET</td>
</tr>
<tr>
<td>VisTrails, U. of Utah</td>
<td>ES3, UCSB</td>
</tr>
<tr>
<td>NCSA, NCSA</td>
<td>PASS, Harvard</td>
</tr>
<tr>
<td>ISI with PASOA, ISI</td>
<td></td>
</tr>
</tbody>
</table>

The Open Provenance Model – p.10/38
Outcomes

- Differences between “process provenance” and “data provenance” easily bridged
- Integrating two or three systems’ provenance data meant interpreting where an identifier produced by one system referred to the same entity as another identifier produced by a different system.
- Provenance must, at least, contain a causality graph, i.e. the process that occurred, the derivation of data etc.
- It must be an annotated causality graph, in order to capture the details and not just the structure of the provenance.
OPM Requirements

- To allow provenance information to be exchanged between systems, by means of a compatibility layer based on a shared provenance model.
- To allow developers to build and share tools that operate on such provenance model.
- To define the model in a precise, technology-agnostic manner.
- To support a digital representation of provenance for any “thing”, whether produced by computer systems or not.
OPM Non-Requirements

- OPM does not specify the internal representations that systems have to adopt to store and manipulate provenance internally.
- OPM does not define a computer-parsable syntax for this model.
- OPM does not specify protocols to store such provenance information in provenance repositories.
- OPM does not specify protocols to query provenance repositories.
**Entities**

**Artifact**  Immutable piece of state, which may have a physical embodiment in an physical object, or a digital representation in a computer system.

**Process**  Action or series of actions performed on or caused by artifacts, and resulting in new artifacts.

**Agent**  Contextual entity acting as a catalyst of a process, enabling, facilitating, controlling, affecting its execution.
Edges

- A used(R) P
- P wasGeneratedBy(R) A
- Ag wasControlledBy(R) P

A1 wasTriggeredBy P1
A2 wasDerivedFrom P2
Pegasus Example

Pegasus/Condor Dagman

wasControlledBy(enactor)

FITS DataSet

used(inputSet)

Degree

used(size)

Produce Sky Mosaic

wasGeneratedBy(out)

Mosaic
An Account

(2,6)

add1ToAll

(3,7)
Combining Accounts

The diagram represents a process involving accounts, with nodes labeled as follows:

- (2,6)
- Split
- Add1ToAll
- (3,7)
- Cons

Connections show the flow of operations:

- (2,6) to Split
- Split to Add1ToAll
- Add1ToAll to (3,7)
- (3,7) to Cons
- Cons to 6
- 6 to 2
- 2 to +1
- +1 to 3
- +1 to 3
- 3 to 7
- 7 to +1
- +1 to 6

This diagram illustrates operations and links between accounts, possibly indicating a process of combining accounts or performing operations on them.
Distinguishing Accounts

The Open Provenance Model – p.21/38
1. *Artifacts* are identified by unique identifiers. Two artifacts are equal if they have the same identifier. Artifacts can optionally belong to accounts. (Likewise, for processes and agents)

2. *Edges* are identified by their source, destination, and role. The source and destination consist of identifiers for artifacts, processes, or agents. Two edges are equal if they have the same source, the same destination, and the same role. Edges can also optionally belong to accounts.
Definitions 2

1. Roles are mandatory in edges *used*, *wasGeneratedBy* and *was Controlled By*. The meaning of a role is defined by the semantics of the process they relate to. Within the context of a process, roles are *not* required to be different.

2. If an edge belongs to an account, then its source and destination also belong to this account.
Definitions 3

1. An *OPM graph* is a set of artifacts, processes, agents, edges, and accounts. (May be cyclic!)

2. The set of OPM graphs is closed under the intersection and union operations. OPM graphs may be disconnected. The empty set is an OPM graph. A singleton containing an artifact, a process or an agent is an OPM graph.

3. A view of an OPM graph according to *one* account, referred to as *account view*, is the set of elements whose effective account membership (for artifacts, processes, and
1. While cycles can be expressed in the syntax of OPM, a *legal account view* is defined as an acyclic account view, which contains at most one `wasGeneratedBy` edge per artifact. This ensures that within one account, an OPM graph captures proper causal dependencies, and that a single explanation of the origin of an artifact is given.

2. Hence, a *legal OPM graph* is one for which all account views are legal.

3. Legal account views are OPM graphs. The union of two legal account views is an OPM graph (it is not a legal view since it may contain cycles).
1. Two account views can be declared to be alternate to express the fact that represent different descriptions of an execution.

2. A declaration that two views are alternate is legal if the views have some artifact, process or agent in common.

3. A provenance graph is a legal OPM graph where alternate views are legal.

4. Edges can optionally be annotated with time information.

5. A provenance graph does not need to contain time annotations.
**ProcessId** : \textit{primitive set} (Process Identifiers)

**ArtifactId** : \textit{primitive set} (Artifact Identifiers)

**AgentId** : \textit{primitive set} (Agent Identifiers)

**Role** : \textit{primitive set} (Roles)

**Name** : \textit{primitive set} (Names)

**Account** : \textit{primitive set} (Accounts)
Formal Definition (2)

\[
\begin{align*}
\text{ProcessAttributes} & = \text{Name} \times \ldots \\
\text{ArtifactAttributes} & = \text{Name} \times \ldots \\
\text{AgentAttributes} & = \text{Name} \times \ldots \\
\text{Process} & = \text{ProcessAttributes} \times \mathbb{P}(\text{Account}) \\
\text{Artifact} & = \text{ArtifactAttributes} \times \mathbb{P}(\text{Account}) \\
\text{Agent} & = \text{AgentAttributes} \times \mathbb{P}(\text{Account}) \\
\text{ProcessMapping} & = \text{ProcessId} \rightarrow \text{Process} \\
\text{ArtifactMapping} & = \text{ArtifactId} \rightarrow \text{Artifact} \\
\text{AgentMapping} & = \text{AgentId} \rightarrow \text{Agent}
\end{align*}
\]
Formal Definition (3)

\[
\begin{align*}
\text{Used} & = \text{ProcessId} \times \text{Role} \times \text{ArtifactId} \times \mathbb{P}(\text{Account}) \\
\text{WasGeneratedBy} & = \text{ArtifactId} \times \text{Role} \times \text{ProcessId} \times \mathbb{P}(\text{Account}) \\
\text{WasTriggeredBy} & = \text{ProcessId} \times \text{ProcessId} \times \mathbb{P}(\text{Account}) \\
\text{WasDerivedFrom} & = \text{ArtifactId} \times \text{ArtifactId} \times \mathbb{P}(\text{Account}) \\
\text{WasControlledBy} & = \text{ProcessId} \times \text{AgentId} \times \mathbb{P}(\text{Account}) \\
\text{Alternate} & = \text{Account} \times \text{Account}
\end{align*}
\]
Formal Definition (4)

\[ OPM_{\text{Graph}} = \text{ArtifactMapping} \times \text{ProcessMapping} \times \text{AgentMapping} \times \mathbb{P}(\text{Used}) \times \mathbb{P}(\text{WasGeneratedBy}) \times \mathbb{P}(\text{WasTriggeredBy}) \times \mathbb{P}(\text{WasDerivedFrom}) \times \mathbb{P}(\text{WasControlledBy}) \times \mathbb{P}(\text{Alternate}) \]
Example (1)

\[
\begin{align*}
\text{ProcessID} & = \{P_1, P_2, P_3, P_4, P_5\} \\
\text{ArtifactID} & = \{a_1, a_2, a_3, a_4, a_5, a_6\} \\
\text{Account} & = \{G, O\} \\
\text{pm} & \subseteq \text{ProcessMapping} \\
\{ P_1 \rightarrow \langle\langle \text{add1ToAll, ...} \rangle, \{G\}, P_4 \rightarrow \langle\langle +1, ... \rangle, \{O\} \rangle \\
P_2 \rightarrow \langle\langle \text{split, ...} \rangle, \{O\} \rangle, P_5 \rightarrow \langle\langle \text{cons, ...} \rangle, \{O\} \rangle \\
P_3 \rightarrow \langle\langle +1, ... \rangle, \{O\} \rangle \}
\end{align*}
\]

\[
\begin{align*}
\text{am} & \subseteq \text{ArtifactMapping} \\
\{ a_1 \rightarrow \langle\langle (2, 6), ... \rangle, \{G, O\} \rangle, a_2 \rightarrow \langle\langle (3, 7), ... \rangle, \{G, O\} \rangle, \\
a_3 \rightarrow \langle\langle 2, ... \rangle, \{O\} \rangle, a_4 \rightarrow \langle\langle 6, ... \rangle, \{O\} \rangle, \\
a_5 \rightarrow \langle\langle 3, ... \rangle, \{O\} \rangle, a_6 \rightarrow \langle\langle 7, ... \rangle, \{O\} \rangle \}
\end{align*}
\]
Example (2)

\[
\begin{align*}
\text{u} & \subseteq \text{Used} = \\
& \{ \langle P_1, \text{in}, a_1, \{G\} \rangle, \langle P_2, \text{pair}, a_1, \{O\} \rangle, \\
& \quad \langle P_3, \text{in}, a_3, \{O\} \rangle, \langle P_4, \text{in}, a_4, \{O\} \rangle, \\
& \quad \langle P_5, \text{left}, a_5, \{O\} \rangle, \langle P_5, \text{right}, a_6, \{O\} \rangle \}
\end{align*}
\]

\[
\begin{align*}
\text{g} & \subseteq \text{WasGeneratedBy} = \\
& \{ \langle a_2, \text{out}, P_1, \{G\} \rangle, \langle a_3, \text{left}, P_2, \{O\} \rangle, \\
& \quad \langle a_4, \text{right}, P_2, \{O\} \rangle, \langle a_5, \text{out}, P_3, \{O\} \rangle, \\
& \quad \langle a_6, \text{out}, P_4, \{O\} \rangle, \langle a_2, \text{pair}, P_5, \{O\} \rangle \}
\end{align*}
\]

\[
\begin{align*}
\text{a} & \subseteq \text{Alternate} = \\
& \{ \text{alternate}(O, G) \}
\end{align*}
\]
One Step Inferences

A1
used(R1)
Acc1

P1
wasGeneratedBy(R2)
Acc2

A2
wasTriggeredBy

P2
used(R3)
Acc3

wasDerivedFrom
\[
\langle p_2, r_3, a_2, acc_3 \rangle \in Used \\
\land \ \langle a_2, r_2, p_1, acc_2 \rangle \in WasGeneratedBy \\
\langle p_2, p_1, acc_2 \cup acc_3 \rangle \in WasTriggeredBy
\]

\[
\langle a_2, a_1, acc \rangle \in WasDerivedFrom \\
\exists p_1, r_1, r_2, acc_1, acc_2, \\
\langle a_2, r_2, p_1, acc_2 \rangle \in WasGeneratedBy \\
\land \ \langle p_1, r_1, a_1, acc_1 \rangle \in Used \\
\land \ acc_1 \cup acc_2 = acc
\]
Transitive Closure

\[ a_2 = a_1 \]
\[ \lor \exists a_3, \langle a_2, a_3, acc_2 \rangle \in WasDerivedFrom \]
\[ \land \langle a_3, a_1, acc_1 \rangle \in WasDerivedFrom^* \]
\[ \land acc = acc_1 \cup acc_2 \]
\[ \langle a_2, a_1, acc \rangle \in WasDerivedFrom^* \]
Time Annotations

1. Are not required to explain causality
2. Time precedence does not imply causal dependence
3. Causal dependence implies time ordering (assuming a same clock)
t1 < t2 (artifact must exist before the start of the process it causes)
t1 < t3 (artifact must exist before being used)
t2 < t3 (process must have started before using artifacts)
t4 < t5 (process generates artifacts before it ends)
t4 < t6 (artifact must exist before being used)
no constraint between t3 and t4
no constraint between t5 and t6
What Next?

- Release OPM specification publicly for comment
- Provenance Challenge 3
- Inferences
- Streams
- Relationship with models of causality
- Inter-operability: recording, querying, etc
- Benchmarking, Performance