A Framework for Policies over Provenance

Tyrone Cadenhead, Murat Kantarcioglu
and Bhavani Thuraisingham

The University of Texas at Dallas

TaPP 2011 : 3rd USENIX Workshop on the Theory and Practice of Provenance
Heraklion, Crete, Greece
Jun 20, 2011 - Jun 21, 2011

What is Provenance?

- Provenance records the history of a document
  - Takes the form of directed graph
  - Captures the causality among documents
- Provenance is the lineage or pedigree of a resource
- Provenance determines the trustworthiness of shared information
  - Used for conducting their day-to-day with high quality information
- Metadata about the origin and history of a piece of item
  - annotations about data items
  - account of the history affecting data items
- A data item can be electronic or physical

- Provenance is essential
  - ***In healthcare***: tracks the activities of healthcare professionals, regulatory compliance
  - ***In E-science***: replicates experiments and verify the steps and the results
  - ***In business***: provides an audit trail, which can be used for accountability
  - ***In intelligence***: verifies the sources of information
  - ***In courts***: provides trace and evidence
  - ***Data quality***: estimates data reliability and trustworthiness

Unified Framework

- Provides intermediary policy languages that specify policies over a provenance graph
- Translates policies into graph operations over a provenance graph
  - Make use of regular expression queries
- Evaluates different policy sets over a provenance graph
  - View their outcomes graphically
- Compare the words described by regular expression queries
  - Determine equivalence and subsumption of policies
- Write more compact policies
  - Eliminate redundancies and inefficiencies
- Interface accepts a high level policy
  - translate into the required format for our graph rewriting system
  - abstract the details of the framework from a user

High Level Policy Language (Access Control)

```
<policy ID="1" >
  <target> <subject>anyuser</subject>
      <record>Report3</record>
    <restriction>
     Report3 [WasGeneratedBy] process    AND   process [WasTriggeredBy]/country
    </restriction>
   <scope>non-transferable</scope>
 </target>
 <condition>purpose == research</condition>
  <effect>Permit</effect>
</policy>
```

- **subject** : name of a user or any collection of users
- **record** : name of a resource
- **restriction**: refines the applicability of subject or record
- **scope** : indicate whether target applies only to record or its ancestry
- **condition**: describes conditions access is permitted.
- **effect** : if policy is positive or negative authorization

- **Current Drawbacks:**
  - Typically defined for systems with single data items.
  - The number of resources in a provenance graph is exponential in the number of single resources.
  - To identify all these resources, we need to iterate all of them
  - Lead to administration burdens, when done manually
- **No support the provenance directed graph structure**
  - The relationships between the single data items is what sets a provenance access policy apart from the existing access control policies)

High Level Policy Language (Redaction)

```
<policy ID="2" >
  <lhs> start=Report3
      chain=[WasGeneratedBy] process AND
      process [Used] report AND
      report [WasGeneratedBy] process.
  </lhs>
  <rhs> start=Report3
      chain=[WasGeneratedBy] process AND
      process [WasTriggeredBy] _:A1.
  </rhs>
  <condition><application>null</application>
    <attribute>null</attribute>
  </condition>
  <embedding>
    <pre>null</pre>
    <post>(ProcessJ,Used, Report3)</post>
  </embedding>
</policy>
```

- **lhs** element describes the left hand side of a rule.
- **rhs** element describes the right hand side of a rule.
- **starting** entity. Each path in the lhs and rhs begins at a starting point.
- **condition** element has two optional sub elements,
  - the **application** defines the conditions that must hold for rule application to proceed
  - the **attribute** element describes the annotations in *LHS*
- **embedding** element has two optional sub elements,
  - **pre** describes how *LHS* is connected to the provenance graph
  - **post** describes how *RHS* is connected to the provenance graph

- What is Redaction
  - Process that protects sensitive information by removing or circumventing it
  - A process that focuses on sharing information

- What is a Redaction policy?
  - Rules that govern how to completely or partially remove sensitive attributes of the information being shared
- Commercially available redaction tools
  - block out (or delete) the sensitive parts of documents
  - available as text and images

Why we need graphs?

- **Graphs are a very natural representation of data in many application domains**
  - Precedence networks, path hierarchy, family tree and concept hierarchy
- Provenance has a directed structure
  - Captures history, captures causal relationships
- Open provenance Model (OPM) describes provenance as a directed acyclic graph (DAG)
- Policy operations conceptualized as graph operations over provenance
- Provenance can be realized as a directed graph in order to visualize the causal relationships among entities
  - A "happens before" B is well captured in a directed labeled graph

Graph Models:

- **Resource Description Framework (RDF)**
  - A graph data model
  - A Semantic Web technology
  - RDF is a W3C Recommendation for representing data on the web
  - Expresses metadata or descriptions about any resources on the web
- A RDF triple is an ordered set (s p o)
  - the subject, predicate and object, respectively
- A predicate makes an assertion about the subject
- A set of RDF triples constitute a RDF graph
  - Represents the knowledge about a system

- **Open Provenance Model (OPM)**
  - Abstract model
  - Provenance as a directed acyclic graph that captures causal relationships
- OPM graph can be further enriched annotations
  - About time, location and other relevant contextual information
- The OPM model identifies three categories of entities
  - Artifacts, Processes and Agents
- Abstract vocabulary describe relationships between the entities

- *RDF Triples:*

<opm:Process> <opm:WasControlledBy> <opm:Agent>
<opm:Process> <opm:Used> <opm:Artifact>
<opm:Artifact> <opm:WasDerivedFrom> <opm:Artifact>
<opm:Artifact> <opm:WasGeneratedBy> <opm:Process>
<opm:Process> <opm:WasTriggeredBy> <opm:Process>

Provenance Graph: A Definition

- Provenance graph is a restricted RDF graph
  - ***Directed edges*** indicating that an event happened before another event
  - ***Causal dependencies*** between the node entities
    - Edges start at a node called ***the effect*** and points to another node called ***the cause*** of the event
  - ***Acyclic***, indicating that history is non-cyclic and immutable

- RDF graph (set of RDF triples)
- A RDF triple (s, p, o)
  - represented graphically as

$$s \xrightarrow{p} o$$

  - s is causally dependent on o
  - s as the effect and o as the cause of s

- V ={WasControlledBy, Used, WasDerivedFrom, WasGeneratedBy, WasTriggeredBy}
  - **Path (<s_1> (P) <o_n>)**
  - Define P over V using regular expressions
  - (x, [p]*, y) and (x,[p]+, y)
  - (x [WasGeneratedBy] / [WasControlledBy] y)

Provenance Graph *(in Intelligence domain)*

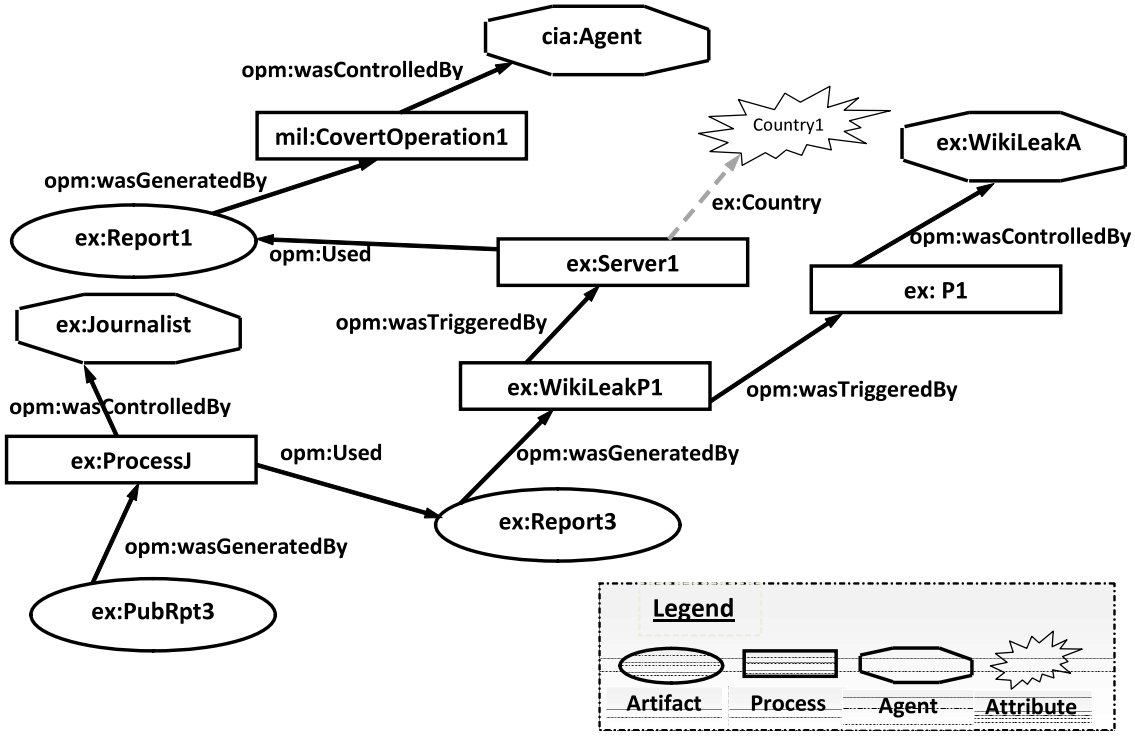
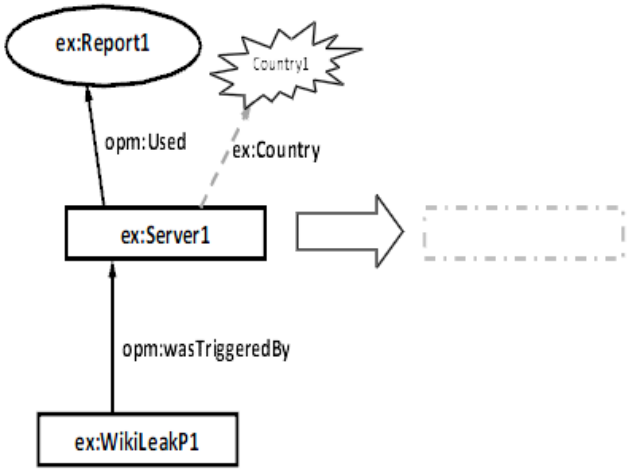


Figure 2

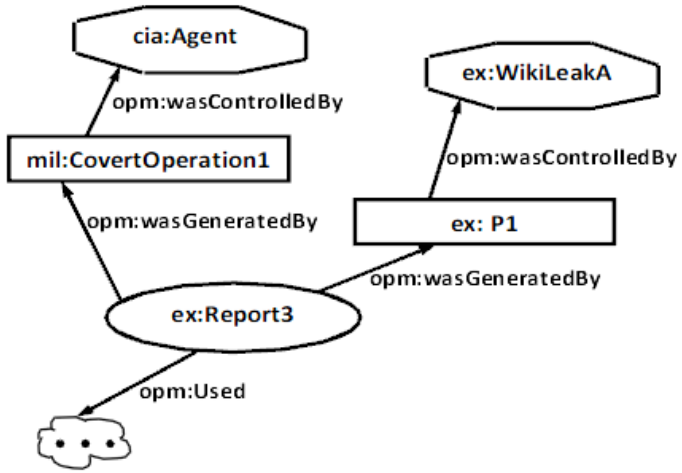
Graph Rewriting

- *A graph rewriting system is a three tuple, (Gℓ , q, P)*
  - *Gℓ is a labeled directed graph*
  - *q is a request on Gℓ that returns a subgraph Gq*
  - *P is a policy set*
- *For every policy p = (r, e) in P , r = (se, re) is a production rule*
  - *where se is a starting entity and re is a regular expression string; and e is an embedding instruction*
- *A production rule, r : L → R where L is a subgraph of Gq and R is a graph*
  - *We also refer to L as the left hand side (LHS) of the rule and R as the right hand side (RHS) of the rule*
- *During a rule manipulation, L is replaced by R and we embed R into Gq − L*
- *Embedding Information, e:*
  - *This specifies how to connect R to Gq − L*
  - *Gives special post-processing instructions for graph nodes and edges on the RHS of a graph production rule*

Valid Provenance Graph



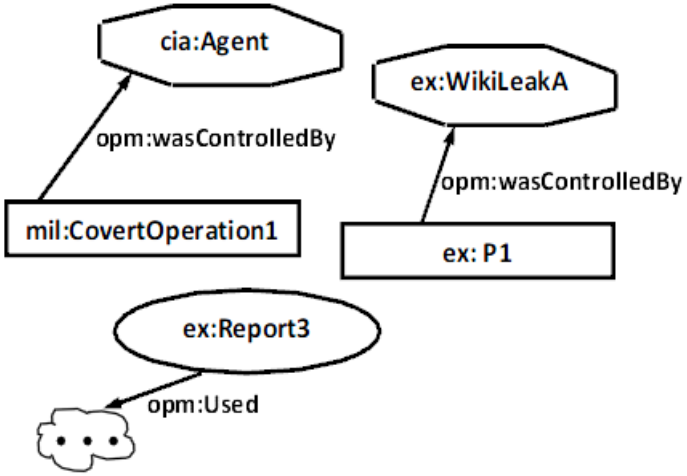
(a) Redaction Policy



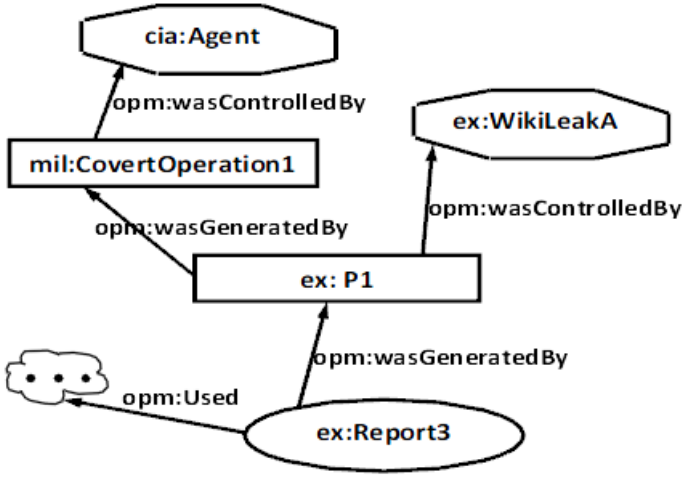
(b) Redacted Graph 1

- rule in Figure 5(a) that replaces a one subgraph with a null (or empty) graph.
- Figures 5(b) is the result of performing a transformation using the rule in Figure 5(a) and the following embedding instruction:
- <ex:Report3> <opm:WasGeneratedBy> <mil:CovertOperation1>
- <ex:Report3> <opm:WasGeneratedBy> <ex:P1>

Valid Provenance Graph



(c) Redacted Graph 2



(d) Redacted Graph 3

- Figures 5(c) is the result of performing a transformation using the rule in Figure 5(a) but with an empty embedding instruction.
- valid OPM graph, but the causal relationships are not preserved, for example there is a causal relationship between ex:Report3 and cia:Agent

- Figure 5(d) is the result of performing a transformation using the rule in Figure 5(a) and the following embedding instruction:
- <ex:P1> <opm:WasGeneratedBy> <mil:CovertOperation1>
- RDF triple <ex:P1> <opm:WasGeneratedBy> <mil:CovertOperation1> does not conform to the OPM nomenclature convention

Conclusions

- Solution not limited
  - no restriction on input provenance format
  - Any format XML, Relational or RDF
- Causal relationships
  - Easily visualized
  - Supports directed structure of provenance
- Propose a unified framework
  - Allows a domain user a choice of policies
  - Protecting and sharing provenance information
- Extends previous policy definitions
  - Support provenance
- Leverage over open technologies
  - RDF, SPARQL, OPM