On Data Provenance in Group-centric Secure Collaboration

Oct. 17, 2011
CollaborateCom

Jaehong Park, Dang Nguyen and Ravi Sandhu
Institute for Cyber Security
University of Texas at San Antonio
Group-centric Collaboration
Group Collaboration Operations

• Administrative operations
  — Establish/disband groups, join/leave/substitute users, add/remove object versions to/from a group, import/merge object versions from a group to an org

• Usage operations
  — Read/update/create object versions
Towards Assured Data Provenance

- **Data Provenance Security & Trustworthiness**
  - e.g. Provenance Access/Usage Control

- **Provenance Data**
  - Some Assurance

- **Data Provenance Model/System**
  - GC Provenance
  - Some Utilities

- **Operations for Provenance**
  - Group Collaboration

**Target Domain**
Data Provenance

• **Utilities of data provenance**
  – Pedigree, Usage tracking, Versioning capability
  – Trustworthiness, Accountability, Compliance

  – Depend on the kinds of provenance data that are captured
Capturing Provenance Data

- Capturing a complete provenance data for all operations is neither feasible nor necessary
  - Some can be captured only by user’s manual declaration (i.e., user intention) while user’s memory is limited and cannot identify all the source information (i.e., citations in scientific research article).
  - Not all operation information provide additional provenance utilities
- For proper discussion, we need a specific application domain where a set of operations can be specified and expressed
Data Provenance Requirements

- **Identifying** operations for provenance data
- **Capturing** operations as provenance data in a provenance model
- **Provenance data expression**
- **Provenance data querying**
- **Provenance data analysis**

- **Data Provenance Assurance**
  - Access/usage Control, trustworthiness, integrity, accountability, etc.
Data Object Versioning

• One object can have multiple versions
• Each version can have a multiple identical copies
• The versions of an object form a rooted tree structure, relating a parent version to its immediate children versions
• Each copy is considered as a separate object.
Open Provenance Model (OPM) Notations

• 3 Nodes
  – Artifact (ellipse)
  – Process (Rectangle)
  – Agent (Octagon)

• 5 Causality dependency edges (not dataflow)
OPM includes...

- A unique identifier for each node
  - To distinguish nodes of the same type

- Accounts
  - Multiple abstracted views of provenance graph by utilizing indirect (dashed) edges

- OPM Profile
  - Includes domain specific subtypes of edges that are defined for additional semantics
  - Includes role-specific (solid) edges
Establish/Disband operations

a) Establish operation using orgs' admin

b) Disband operation
Join/Leave Operations

a) Join/Leave operation on group

b) Join/Leave operation w/ attribute update
Add/Remove Operations

a) Add operation

b) Remove operation
Substitute/Import Operations

a) Substitute operation

b) Import operation
Merge Operation

• Similar to “import”
  – A version is copied from cg to org

• Different from “import”
  – The initial version of the merged version in cg was added from the org while the initial version of imported version is newly created in cg
  – The merged version becomes a new version of the original version in org
Read/Update/Create Operations

- **Read**
  - a) Read operation
  - u(toRead)
  - u(sourceEntity)

- **Create**
  - u(targetEntity)
  - wasCreatedIn

- **Update**
  - u(toUpdate)
  - wasNewVersionOf
  - WasUpdatedIn
OPM in RDF Expression

- Using RDF (Resource Description Framework) data representation to express provenance data
- RDF supports a directed graph

```xml
<opm:process><opm:used><opm:artifact>
<opm:artifact><opm:wasGeneratedBy><opm:process>
<opm:process><opm:wasControlledBy><opm:agent>
<opm:process><opm:wasTriggeredBy><opm:process>
<opm:artifact><opm:wasDerivedFrom><opm:artifact>
```
OPM Profile for Group Collaboration Operations (subtypes of “wasDerivedFrom”)

<gcp:artifact><gcp:wasCopyOf><gcp:artifact>
<gcp:artifact><gcp:wasNewVersionOf><gcp:artifact>
<gcp:artifact><gcp:HadAdmin><gcp:artifact>
<gcp:artifact><gcp:HadJoinedCgMember><gcp:artifact>
<gcp:artifact><gcp:HadLeftCgMember><gcp:artifact>
<gcp:artifact><gcp:HadRemovedAdmin><gcp:artifact>
<gcp:artifact><gcp:HadAddedAdmin><gcp:artifact>
<gcp:artifact><gcp:wasCreatedIn><gcp:artifact>
<gcp:artifact><gcp:wasUpdatedIn><gcp:artifact>
Roles for “Used” Edges

\[
\begin{align*}
\langle \text{gcp:process}\rangle & \langle \text{gcp:u(sourceEntity)}\rangle \langle \text{gcp:artifact} \rangle \\
\langle \text{gcp:process}\rangle & \langle \text{gcp:u(targetEntity)}\rangle \langle \text{gcp:artifact} \rangle \\
\langle \text{gcp:process}\rangle & \langle \text{gcp:u(adminGroup)}\rangle \langle \text{gcp:artifact} \rangle \\
\langle \text{gcp:process}\rangle & \langle \text{gcp:u(removedAdmin)}\rangle \langle \text{gcp:artifact} \rangle \\
\langle \text{gcp:process}\rangle & \langle \text{gcp:u(addedAdmin)}\rangle \langle \text{gcp:artifact} \rangle \\
\langle \text{gcp:process}\rangle & \langle \text{gcp:u(initialAdmin)}\rangle \langle \text{gcp:artifact} \rangle \\
\langle \text{gcp:process}\rangle & \langle \text{gcp:u(toJoin)}\rangle \langle \text{gcp:artifact} \rangle \\
\langle \text{gcp:process}\rangle & \langle \text{gcp:u(toLeave)}\rangle \langle \text{gcp:artifact} \rangle \\
\langle \text{gcp:process}\rangle & \langle \text{gcp:u(toAdd)}\rangle \langle \text{gcp:artifact} \rangle \\
\langle \text{gcp:process}\rangle & \langle \text{gcp:u(toRemove)}\rangle \langle \text{gcp:artifact} \rangle \\
\langle \text{gcp:process}\rangle & \langle \text{gcp:u(toImport)}\rangle \langle \text{gcp:artifact} \rangle \\
\langle \text{gcp:process}\rangle & \langle \text{gcp:u(toMergeTo)}\rangle \langle \text{gcp:artifact} \rangle \\
\langle \text{gcp:process}\rangle & \langle \text{gcp:u(toMergeFrom)}\rangle \langle \text{gcp:artifact} \rangle \\
\langle \text{gcp:process}\rangle & \langle \text{gcp:u(toRead)}\rangle \langle \text{gcp:artifact} \rangle \\
\langle \text{gcp:process}\rangle & \langle \text{gcp:u(toUpdate)}\rangle \langle \text{gcp:artifact} \rangle
\end{align*}
\]
Roles for “WasGeneratedBy” Edges

<gcp:artifact><gcp:g(toEstablish)><gcp:process>
<gcp:artifact><gcp:g(toJoin)><gcp:process>
<gcp:artifact><gcp:g(toLeave)><gcp:process>
<gcp:artifact><gcp:g(toAdd)><gcp:process>
<gcp:artifact><gcp:g(toSubstitute)><gcp:process>
<gcp:artifact><gcp:g(toImport)><gcp:process>
<gcp:artifact><gcp:g(toMerge)><gcp:process>
<gcp:artifact><gcp:g(toCreate)><gcp:process>
<gcp:artifact><gcp:g(toUpdate)><gcp:process>
SPARQL Query Expression

• Standard query language for RDF
• Can query by stating a consecutive path of specific triple types of subject, predicate, and object

```
SELECT ?ver
WHERE{
  gcp:cg1.o2v2 gcp:wasCopyOf ?obj.
  ?obj gcp:wasNewVersionOf ?ver.}
```
GLEEN-enabled SPARQL

• Gleen is a plugin for the ARQ query engine.
• ARQ is a query engine for Jena, a semantic web framework for Java which supports the SPARQL RDF query language
• Gleen onPath function supports regular expression-based recursive path patterns

subject gleen:OnPath (pathExpression object)
Provenance Data Example
Sample Query 1

- Identify the very initial version of cg1.o2v3 and whether it is created in the current group or added from an organization.
- The query will return "cg1.o2v1" and "add"

```
SELECT ?obj ?proc
WHERE{
  gcp:cg1.o2v3 gleen:OnPath(
    "[gcp:wasNewVersionOf]*" ?obj ).
  ?obj gleen:OnPath(
    [gcp:g(toCreate)] | [gcp:g(toAdd)] ?proc).
}
```
Sample Query (cont.)

- To verify users who may have influenced (update/create) an object content regardless of the fact that whether the influence is done on a version of the same object or a version of a copied object of the object.

```sparql
SELECT ?agent
WHERE{
  gcp:org1.o1v4 gleen:OnPath(
    "([ gcp:wasNewVersionOf ]|[gcp:wasCopyOf])*" ?obj).
  ?obj gleen:OnPath([gcp:g(toUpdate )]| [gcp:g(toCreate)] ?proc).
  ?proc gcp:wasControlledBy ?agent.}
```
Summary

• Identified/captured available or necessary operations as provenance data for group collaboration environment

• Expressed in RDF triples so it can be queried by utilizing a regular expression based path patterns in SPARQL query language

• Showed some utilities of data provenance in a group collaboration environment

• Provides an initial foundation for data provenance access control in group collaboration environment
• Questions and Comments?