Provenance: Information for Shared Understanding

M. David Allen
June 2012
Government Mandates

- Net-Centric Data Strategy mandate:
  “Is the source, accuracy and currency of the data asset available to users?”*

- Users need data provenance (i.e., where the information came from and the processes that acted upon that data) to understand and assess trustworthiness of information from many sources

*Net-Centric Data Strategy TechGuide on Goal 3.5, “Enable Data to be Trusted”
https://metadata.dod.mil/mdr/ns/ces/techguide/net_centric_data_strategy_ncds_goals.html#Trustworthy
Motivation

Rapid, agile information sharing

Capabilities composed on demand

Where’d they get those readings?

Why did we make that decision?

Who says? Should I act on this?
Provenance Basics

- “Family Tree” of relationships
  - Ovals = data,
  - Rectangles = processes
  - Show how data is used and reused

- Basic metadata
  - Timestamp
  - Owner
  - Name/Description

- Can also include annotations
  - E.g. quality information

- Does not contain the actual data object
Provenance is a Record of What Happened

The difference: Number of Air Surveillance Reports used.
The PLUS System
Requirements

- “Open World” collection in distributed, heterogeneous environments
  - Enables use of provenance in real systems
- Flexible annotation management over provenance
  - Enables analysis applications, e.g. taint analysis
- Attribute-based access controls
  - Enables sharing provenance across users with different privilege levels
- Security techniques
  - Enable maximally informative provenance information while protecting sensitive nodes and edges
Provenance Capture
Provenance Capture Must be Minimally Invasive to the User and Underlying Applications

- Application is not significantly slowed down by provenance collection
- Application developer time used to encode provenance calls is minimized
- Application developers can be provenance unaware and legacy applications can continue with normal operations
- Useful provenance information is obtained
Architectural Options for Provenance Capture

■ Insertion into a System Backbone
  – Strategy: Listen in to whatever is happening, and log silently as it happens
  – Requires a small number of points of provenance capture: ESBs are ideal, since they act as central “routers”

■ Owning the System
  – Strategy: Constrain all data and processes to a single system, and force the system to track provenance

■ Single Application Modification
  – Strategy: Each application calls provenance API to log whatever it thinks is important.
  – But, unrealistic for legacy applications

■ Legacy System Capture
  – Strategy: Write a transparent wrapper service. Make sure all orchestrations call the wrapper service with enough information for the wrapper to invoke the real thing.
  – Strategy: Scrape logs
Minimally Invasive Provenance Capture
Implementation

Java Reflection

- Ability in Java to dynamically analyze an object in the virtual machine at runtime
- Call a getter method to obtain information about the object
Once a backbone is provenance-enabled, any application that runs over that backbone is automatically provenance-enabled.
Another Example: Ozone Widget Framework (OWF)

- Used within the Intelligence Community as a tool for providing application and data resources to analysts
- Messaging Framework that allows applications (widgets) to be run within a browser
- Widgets can be built and released by any organization and found within a marketplace

- Widgets themselves shouldn’t be provenance aware
- The Framework should be provenance aware
Another Example: Web Proxy

- Used by organizations to funnel HTTP requests
- Modified so that we log all outgoing requests and responses

- Allows inspection of:
  - Results from web services
  - Parameters sent to web services
  - Who uses the results
  - Parts of the network that supply information
Architectural Options for Provenance Capture

■ Insertion into a System Backbone
  – Strategy: Listen in to whatever is happening, and log silently as it happens
  – Requires a small number of points of provenance capture: ESBs are ideal, since they act as central “routers”

■ Owning the System
  – Strategy: Constrain all data and processes to a single system, and force the system to track provenance

■ Single Application Modification
  – Strategy: Each application calls provenance API to log whatever it thinks is important.
  – But, unrealistic for legacy applications

■ Legacy System Capture
  – Strategy: Write a transparent wrapper service. Make sure all orchestrations call the wrapper service with enough information for the wrapper to invoke the real thing.
  – Strategy: Scrape logs
Owing the System

- Requirement to Use this Method: All data and processes occur within the bounds of the “system”
  - E.g. Workflow management system
  - E.g. A single machine in which the OS is controlled
Architectural Options for Provenance Capture

- Insertion into a System Backbone
  - Strategy: Listen in to whatever is happening, and log silently as it happens
  - Requires a small number of points of provenance capture: ESBs are ideal, since they act as central “routers”

- Owning the System
  - Strategy: Constrain all data and processes to a single system, and force the system to track provenance

- Single Application Modification
  - Strategy: Each application calls provenance API to log whatever it thinks is important.
  - But, unrealistic for legacy applications

- Legacy System Capture
  - Strategy: Write a transparent wrapper service. Make sure all orchestrations call the wrapper service with enough information for the wrapper to invoke the real thing.
  - Strategy: Scrape logs
Sometimes Single App Modification Is the Only Way

**Events**

- Great to “fill the holes” in provenance for a distributed system
- Worse development time
- Development time penalty for every new application
Architectural Options for Provenance Capture

- **Insertion into a System Backbone**
  - Strategy: Listen in to whatever is happening, and log silently as it happens
  - Requires a small number of points of provenance capture: ESBs are ideal, since they act as central “routers”

- **Owning the System**
  - Strategy: Constrain all data and processes to a single system, and force the system to track provenance

- **Single Application Modification**
  - Strategy: Each application calls provenance API to log whatever it thinks is important.
  - But, unrealistic for legacy applications

- **Legacy System Capture**
  - Strategy: Write a transparent wrapper service. Make sure all orchestrations call the wrapper service with enough information for the wrapper to invoke the real thing.
  - Strategy: Scrape logs
No modifications of the legacy application can be allowed.

Wrappers

- Intercept calls to and from a legacy system
- Log all information coming in and going out to the provenance system
- Pass calls to legacy system and caller as if no provenance system involved

Log Scrapers

- Parsing log or monitoring files generated by tools
  - E.g. STrace
  - E.g. Pentaho’s Kettle
Evaluation and How to Choose Capture Points
Developer Hours to Add Provenance Reporting

- MULE ESB
- N-Tool
- X-Tool
- M-Tool

Bar chart showing developer hours for different tools and application types.
Breakdown of Time Needed

Given software, documentation, & access to a developer for questions. All integration was performed by a 3rd party developer.

- **M-Tool**: 49hr
  - Install and Test
  - Test Calls in 3rd Party Code
  - Debug 3rd Party Calls
  - Time for rest of code

- **N-Tool**: 4.25hrs
  - Install and Test
  - Test Calls in 3rd Party Code
  - Debug 3rd Party Calls

- **X-Tool**: 1.3hrs
  - Install and Test
  - Test Calls in 3rd Party Code
  - Debug 3rd Party Calls

- Written in C#.
  - Some time needed to convert PLUS to DLL

- More complicated MULE workflows than X-Tool

- Easily accomplished!
Choosing Capture Points

<table>
<thead>
<tr>
<th>Method</th>
<th>Developer Work (Single App)</th>
<th>Developer Work (Many Apps)</th>
<th>Average Coverage of User’s Actions</th>
<th>Depth of Detailed Provenance Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insertion in a Coordination Backbone</td>
<td>Medium</td>
<td>Low</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>Single Application Modification</td>
<td>Medium</td>
<td>High</td>
<td>Dependent upon usage of application</td>
<td>High</td>
</tr>
<tr>
<td>Legacy System Capture</td>
<td>Low</td>
<td>High</td>
<td>Dependent upon usage of legacy system</td>
<td>Dependent upon what is logged or passed</td>
</tr>
<tr>
<td>Owning the System</td>
<td>Medium</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
</tr>
</tbody>
</table>
Thanks

M. David Allen
The MITRE Corporation
Office: (804) 288-0355
dmallen@mitre.org
Applications of Provenance
The provenance graph is built automatically over time by watching user’s actions.
The system can show relationship information and metadata details.
The system provides ways to get information “at a glance”, e.g. what organizations own the data that was used.
Export Provenance: RDF, XML, JSON

Information in the system can be exported in any format, to be used by other systems.
Help the user assess data products for her specific use.
FitnessWidgets

Specify criteria to judge data for user or use
Annotate any node.

E.g. “This information is tainted and shouldn’t be used!”

Information can be propagated through graph.
Resource Discovery: RSS Publishing, Search, and Alerts

**IM-PLUS Lineage Feed**
Feed of the latest lineage objects reported to IM-PLUS

**Web Request**
Wednesday, April 07, 2010 7:29 PM
This object has an IM-PLUS type of data/string. The object refers to a web resource originally located at [http://money.cnn.com/2.html](http://money.cnn.com/2.html)

**Web Response**
Wednesday, April 07, 2010 7:28 PM
This object has an IM-PLUS type of data/string.

**Service on i.cdn.turner.com**
Wednesday, April 07, 2010 7:28 PM
This object has an IM-PLUS type of invocation/invocation.

**Web Request**
Wednesday, April 07, 2010 7:28 PM
This object has an IM-PLUS type of data/string. The object refers to a web resource originally located at [http://i.cdn.turner.com/money/element/ssl/css/4.0/data/cnn.search.css](http://i.cdn.turner.com/money/element/ssl/css/4.0/data/cnn.search.css)

**Web Response**
Wednesday, April 07, 2010 7:28 PM
This object has an IM-PLUS type of data/string.

**Service on i.cdn.turner.com**
Wednesday, April 07, 2010 7:28 PM
This object has an IM-PLUS type of invocation/invocation.

**Web Request**
Wednesday, April 07, 2010 7:28 PM
This object has an IM-PLUS type of data/string. The object refers to a web resource originally located at [http://i.cdn.turner.com/money/element/ssl/css/4.0/services/members.logon.html](http://i.cdn.turner.com/money/element/ssl/css/4.0/services/members.logon.html)

Inform the user of product usage
Publications

“Surrogate Parenthood: Protected and Informative Graphs” VLDB, 2011

“Provenance Needs Incentives for Everyone” TaPP, 2011

“Enabling Multi-organizational Provenance Exchange” TaPP, 2011

“PLUS: A Provenance Manager for Integrated Information” IRI, 2011

“Capturing Provenance in the Wild” 3rd International Provenance and Annotation Workshop (IPAW 2010)

“Towards Query Interoperability: PASSing PLUS” 2nd International Workshop on Theory and Practice of Provenance (TaPP 2010)

“Provenance-based Belief” 2nd International Workshop on Theory and Practice of Provenance (TaPP 2010)
Publications, continued

- “Understanding Provenance Black Boxes” *International Journal of Parallel and Distributed Databases*

- “Why Not?” *International Conference on the Management of Data (ACM SIGMOD)*, 2009; nominated for best paper award


- “Scalable Access Controls for provenance” *Theory and Practice of Provenance (TAPP ‘09)*, 2009

- “PLUS: Synthesizing privacy, provenance, uncertainty and security” *IEEE Int. Conf. on Data Engineering (ICDE) Workshops*, 2008

- “Provenance Capture and Use: A Practical Guide” MITRE Product available for public release