Policy-Aware Pipes - Accountability in Mashup Service of Linked Data

Fuming Shih

MIT CSAIL
Decentralized Information Group
In Practice ...

![Diagram of BBC and LOD Sets]

- **BBC**: backstage.bbc.co.uk
  - SPARQL Endpoint
  - RSS Feed

- **Company A**
  - LOD Sets
  - RSS
  - SPARQL

- **Organization B**
  - LOD Sets
  - API
  - SPARQL

- **Service/Application**
  - RSS
  - SPARQL Endpoint
  - APIs
Scenario(1)

• Joe is a pop-music fan and a Web expert
• He creates a site called "music-meme.org"
  – Up-to-date articles and news about pop music
  – Publishes RSS feed
  – Music notes: mashup information and reviews
  – Joe reads policies stated on BBC’s website
    • http://welcomebackstage.com/about/faqs/#canimakemoney
Joe’s Checklist of BBC’s Policy

- You must not: Charge users for accessing your work that contains or uses BBC Content
  - Sell applications that use or incorporate BBC Content
  - Re-edit or re-contextualize BBC Content in any way that is illegal, is or is likely to bring the BBC into disrepute or is otherwise inappropriate

- However, you own the Intellectual Property rights in and to your application and nothing prevents you being able to sell or commercially license your application and make money from it.
Scenario(2)

- Bob is a software engineer of website Alpha Music Box, an online music store.
  - Charges users for downloading songs
  - Bob implements music review by obtaining feeds from music-meme.org
  - Eventually, albums with better reviews → better sales!

  • Bob’s assumes everything is free for reuse from music-meme.org
  • Use of music reviews in a commercial mashup possibly infringes BBC’s policy
Observed Problems

- Policy is not machine readable from the source
- Policy should be propagated as well as data in the mashup
- More than just access control - information accountability needed
  - assertions of intended usage on data (e.g. Creative Commons)
  - help people who tend to comply with the policy
  - increase trust
Requirements of Policy Support for Information Accountability

Data Publisher

Mashup Developer

Mashup Consumer

Policy language to assert policies

Policy checking and policy awareness

Provenance information for trust
Pipes

• Rapid prototyping of web content
  – Concepts of the workflow system
  – Graphic editor + built-in operators/filters
  – Easy for casual users to reuse web content
    • Fetch data from multiple sources
    • Apply built-in filters and integrate data
    • Output as web pages or another data stream
  – Deri’s Semantic Pipes ([Linked data] ➔ Policy Aware Pipes ([policy+]))
Deri’s Semantic Pipes
Editors & Debugging

<table>
<thead>
<tr>
<th>log</th>
<th>type</th>
<th>timestamp</th>
<th>message</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&quot;POLICY&quot;</td>
<td>2009-07-13T00:00:32-04:00</td>
<td>[Alert] Profile of <a href="http://foole.csail.mit.edu/wsprofieB#profile">http://foole.csail.mit.edu/wsprofieB#profile</a> is conflicted with policy: <a href="http://foole.csail.mit.edu/policy/policyA#wspolicy">http://foole.csail.mit.edu/policy/policyA#wspolicy</a></td>
</tr>
<tr>
<td></td>
<td>&quot;POLICY&quot;</td>
<td>2009-07-13T00:00:32-04:00</td>
<td>[Alert] Rejeding profile with author: <a href="http://www.w3.org/People/djweitzner/foaf#DJW">http://www.w3.org/People/djweitzner/foaf#DJW</a></td>
</tr>
<tr>
<td></td>
<td>&quot;POLICY&quot;</td>
<td>2009-07-13T00:00:32-04:00</td>
<td>[Error] Some merged web service is conflicted with the policy of channel: <a href="http://meerkat.oreillynet.com/?_fi=rss1.0">http://meerkat.oreillynet.com/?_fi=rss1.0</a></td>
</tr>
<tr>
<td></td>
<td>&quot;POLICY&quot;</td>
<td>2009-07-13T00:00:32-04:00</td>
<td>[Alert] Profile of <a href="http://foole.csail.mit.edu/wsprofieA#profile">http://foole.csail.mit.edu/wsprofieA#profile</a> is conflicted with policy: <a href="http://foole.csail.mit.edu/policy/policyB#wspolicy">http://foole.csail.mit.edu/policy/policyB#wspolicy</a></td>
</tr>
<tr>
<td></td>
<td>&quot;POLICY&quot;</td>
<td>2009-07-13T00:00:32-04:00</td>
<td>[Alert] Rejeding profile with subject: news</td>
</tr>
<tr>
<td></td>
<td>&quot;POLICY&quot;</td>
<td>2009-07-13T00:00:32-04:00</td>
<td>[Error] Some merged web service is conflicted with the policy of channel: <a href="http://johnson.freenet.net/?_fi=rss1.0">http://johnson.freenet.net/?_fi=rss1.0</a></td>
</tr>
</tbody>
</table>

Policy URI

Policy compliance justification
## Scenario Walkthrough

- **Ontologies to model domain knowledge**
  - Policy ontology
  - Web service ontology

```xml
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix p: <http://foolme.csail.mit.edu/ns/ws/ite#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .

p:wspolicy rdf:type rdfs:Class;
  rdfs:label "Web service policy"^^xsd:string .

p:policy rdf:type rdfs:Class;
  rdfs:label "Policy"^^xsd:string .

p:creator rdf:type rdfs:Class;
  rdfs:label "Creator"^^xsd:string ;
  rdfs:domain rdfs:Resource .

p:subject rdf:type rdfs:Class;
  rdfs:label "Subject"^^xsd:string .

p:person rdf:type rdfs:Class;
  rdfs:label "Person"^^xsd:string .

p:resource rdf:type rdfs:Class;
  rdfs:label "Resource"^^xsd:string .

p:property rdf:type rdfs:Class;
  rdfs:label "Property"^^xsd:string .

p:relation rdf:type rdfs:Class;
  rdfs:label "Relation"^^xsd:string .

p:attribute rdf:type rdfs:Class;
  rdfs:label "Attribute"^^xsd:string .

p:domain rdf:type rdfs:Class;
  rdfs:label "Domain"^^xsd:string .

p:range rdf:type rdfs:Class;
  rdfs:label "Range"^^xsd:string .

p:category rdf:type rdfs:Class;
  rdfs:label "Category"^^xsd:string .

p:term rdf:type rdfs:Class;
  rdfs:label "Term"^^xsd:string .

p:property rdf:type rdfs:Class;
  rdfs:label "Property"^^xsd:string .

p:relation rdf:type rdfs:Class;
  rdfs:label "Relation"^^xsd:string .

p:attribute rdf:type rdfs:Class;
  rdfs:label "Attribute"^^xsd:string .

p:domain rdf:type rdfs:Class;
  rdfs:label "Domain"^^xsd:string .

p:range rdf:type rdfs:Class;
  rdfs:label "Range"^^xsd:string .

p:category rdf:type rdfs:Class;
  rdfs:label "Category"^^xsd:string .

p:term rdf:type rdfs:Class;
  rdfs:label "Term"^^xsd:string .
```

<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
xmlns="http://purl.org/rss/1.0/
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cc="http://creativecommons.org/ns#"
xmlns:ws="http://foolme.csail.mit.edu/ns/ws#"

<channel rdf:about="http://www.example.com/?_fl=rss1.0">
<title>Some Data Source Feed</title>
<link>http://www.example.com</link>
<description>A data source with policy and service profile</description>
<cc:license rdf:resource="http://www.creativecommons.org/licenses/by-nc-nd/3.0/us/" />
<cc:morePermissions rdf:resource="http://foolme.csail.mit.edu/policy/policyA#wspolicy" />
...
</channel>

<> dc:title "BBC Backstage Data Policy.
:wspolicy a p:wspolicy ;
 p:rejectSubject "commercial"^^xsd:string ;
 p:creator http://backstage.bbc.co.uk/;
Mashup Developer

• Joe uses PAP to construct his mashup
  – Check policy compliance
    • Use each source’s policy to check all other data sources’ profile
    • Use it’s own policy to check all data sources
  – Generate provenance data
  – Publish RSS feed of music-meme.org

< dc:title ”Music-meme Profile”.
p:profileA a ws:profile ;
  ws:subject ”community”^^xsd:string ;
  ws:author http://dig.csail.mit.edu/People/Joe#I;
Using PAP in Semantic Pipes

1. Uses Semantic Pipe’s operators
2. Adds policy-aware operator
3. Runs & Checks debugging pane
Provenance generation

```
  <rdf:type rdf:resource="http://fooleme.csail.mit.edu/ns/provenance#annotation"/>
  <provenance:timestamp>2009-07-10T13:10:25-04:00</timestamp>
  <provenance:sparql>
      ?artist foaf:name ?name.}"
  </sparql>
</rdf:Description>
```

PAP logs all the operations in the Semantic Pipes, saves logs as triples.
Mashup Consumer

• Bob uses PAP to build another mashup from Joe’s music-meme.org’s data feed

```
  <rdf:type rdf:resource="http://foolme.csail.mit.edu/ns/provenance#annotation"/>
  <provenance:output rdf:resource="http://music-meme.org/sparqleendpoint"/>
  <provenance:timestamp>2009-07-10T13:10:25-04:00</timestamp>
  <provenance:sparql>
    "SELECT ?rev ?name ?title WHERE {
      ?artist foaf:name ?name.}
    </sparql>
  </provenance:sparql>
</rdf:Description>
```
Conclusion

• PAP integrates policy into mashup development
  – Different policy support to different actors in the mashup cycle
• Introduce information accountability to mashup environment
  – Transparency of policy about data usage
  – Increase trust between mashups
Questions?