Practical Data Provenance in Distributed Environment or: implementing Linked Data Broker using Microservices Architecture

Joonas Kesäniemi, Stefan Negru, João da Silva
SWIB 2017
Hamburg
ATTX project

• 8/2016-4/2018
• Developing software component for building semantic data brokers
• Main features
  • “Easy” & scalable deployment
  • Flexible & linked data
  • Full & usable provenance
• Funded by the Ministry of Education and Culture
• Executed by the Helsinki University Library
• http://attx-project.github.io
Data brokering and ATTX

Owners and maintainers of published (open) data

Data sources

Internal data

Redistributed data

ATTX components

Users of redistributed data
ATTX deliverables

COMPONENTS
- WORKFLOW
- GRAPH MANAGER
- PROVENANCE
- PROCESSING
- DISTRIBUTION
- MESSAGE BROKER

DEPLOYMENT ENVIRONMENTS
- SINGLE HOST
  - DOCKER COMPOSE
  - DOCKER SWARM
- OPEN STACK CLOUD
  - DOCKER SWARM
- KONTENA CLOUD
  - KONTENA

PROTOTYPES
- OPEN ACCESS DASHBOARD
  - UNIVERSITY OF JYVÄSKYLÄ
  - HANKEN
- METADATA MAPPING AND VALIDATION
  - CSC / METAX
- RESEARCH DATASET METADATA BROKER
  - UNIVERSITY OF HELSINKI
ATTX core components

- WorkflowManager – UnifiedViews & custom provenance API
- GraphManager
  - Manages the state of the internal graph store
- MessageBroker – RabbitMQ
- Indexing
- Distribution
  - In JSON format using ElasticSearch
- Transformation to RDF
  - RML processor to transform from CSV, JSON and XML
- Transformation from RDF to JSON
  - JSON-LD Framing
- Provenance
“Provenance is a record that describes the people, institutions, entities, and activities involved in producing, influencing, or delivering a piece of data or a thing. In particular, the provenance of information is crucial in deciding whether information is to be trusted, how it should be integrated with other diverse information sources, and how to give credit to its originators when reusing it. In an open and inclusive environment such as the Web, where users find information that is often contradictory or questionable, provenance can help those users to make trust judgements.”

Emphasis mine

Prov-O - You know, for Provenance

Adapted from https://www.w3.org/TR/prov-o/
ATTX provenance model

https://attx-project.github.io/attx-onto/

prov:used / prov:generated
ATTX pipelines

**PIPELINES**

**Ingest (Extract)**
- Download external data

**Process (Transform)**
- Select source datasets
- Create new dataset
- Store new dataset

**Publish (Load)**
- Select source datasets
- Transform to published format
- Publish dataset

**STEPS**

- Extract
- Transform
- Load

**Internal graph store**

Data API
Example case
Connecting publications to files

• CRIS system is the source for publication metadata
  • ID = pub1
  • DOI = doi1
  • Title = “Simple example”

• Digital repository is the source for file metadata
  • ID = file1
  • DOI = doi1
  • Download link = link1
  • File type = “Publisher’s PDF”

• Data broker’s internal data
  - extpub:doi1
  - repo:file1
  - Missing from the input data. Needs to be generated.
Example case – Pipelines in UnifiedViews (UV)

<table>
<thead>
<tr>
<th>Actions</th>
<th>Name</th>
<th>Last run time</th>
<th>Last execution time</th>
<th>Last status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Harvest CRIS publications</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Harvest repository files</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Infer files from parallel pubs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Publish dataset</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example case – Ingestion pipeline (UV)

Transformation from JSON to RDF
Example case – Processing pipeline (UV)

Graph selection using GraphManager

Creating new RDF data

Input graphs

<table>
<thead>
<tr>
<th>Label</th>
<th>Check</th>
</tr>
</thead>
<tbody>
<tr>
<td>JYU Publications RDF</td>
<td>✓</td>
</tr>
<tr>
<td>JYX Files RDF</td>
<td>✓</td>
</tr>
</tbody>
</table>

Graph management
Example case – Publishing pipeline (UV)

Graph selection using GraphManager

Transformation from RDF to JSON

Input graphs

<table>
<thead>
<tr>
<th>Label</th>
<th>Check</th>
</tr>
</thead>
<tbody>
<tr>
<td>JYU Publications RDF</td>
<td>✅</td>
</tr>
<tr>
<td>JYX Files RDF</td>
<td>✅</td>
</tr>
<tr>
<td>File links from parallel pubs</td>
<td>✅</td>
</tr>
</tbody>
</table>

Indexing service
Collecting provenance data

• Explicit messages
  • “I did this”

• “Fire-and-forget” type of operation
  • Message broker is responsible for getting message to the provenance service using message persistency and automatic retries

• Activities are connected through shared input/output entities

• Resulting provenance graph is generated from bits and pieces sent in by multiple components running in different containers and possibly on different nodes
Provenance messages

- **Workflow Management**: executedWorkflow
- **Provenance Service**: executedStep, replacedGraph, retrievedGraph
- **Graph Management**: replacedGraph
- **RML**: generatedRDF
- **Indexing**: replacedIndex
- **Framing**: generatedJson
Publishing provenance

• Provenance service is updating the ElasticSearch index with the up-to-date information automatically

• Provenance graphs are converted to JSON using JSON-LD framing

• Documents related a single provenance graph, i.e. provenance related to single workflow execution, is indexed under common document type
  • GET /prov/workflow1_activity1
Using provenance

• Provenance use case scenarios
  • How are the inputs and outputs of the pipelines related to one another?
  • Document was downloaded from an endpoint X, what are the data sources and transformations related to that endpoint?

• Provenance browser (PoC)
  • Workflow, step and service level information
  • Connections between pipelines
    • WF B used the data generated by WF A as a data source
Publish pipeline execution

Failed run – indexing part is missing

Successful run

Plan

- attx-e-selectDS
- attx-t-framing service
- attx-l-publish toapi
Connected datasets

Created using Prov-O-Viz
http://provoviz.org/
The TODO

• Provenance for incrementally harvested datasets
  • Datasets that have subsets

• Integrating Service Registry to the provenance data
  • More information about the component in a common manner

• Implicit provenance
  • Routing all the messages to the provenance service
  • Creating the request-response patterns based on provenance contexts
Thank you