Measuring PROV Provenance on the Web of Data

Paul Groth[1] and Wouter Beek[2]

[1] *Elsevier Labs*
[2] *Vrije Universiteit Amsterdam*

May 11, 2016

1 Introduction

One of the motivations behind the original charter for the W3C Provenance Incubator group was the need for provenance information for Semantic Web and Linked Data applications. Thus, a question to ask, three years after the introduction of the W3C PROV family of documents, is what is the adoption of PROV by the Semantic Web community.

A proxy for this adoption is measuring how often PROV is used within Linked Data. In this work, we begin to do such a measurement. Our analytics are based on the LOD Laundromat (Beek et al. 2014). The LOD Laudromat crawls and cleans over 650 thousand linked data documents representing over 38 billion triples. LOD Laudromat has been used in the past to do large scale analysis of linked data (Rietveld et al. 2015).

Here, we focus on core statistics based around what PROV-DM refers to as core structures. We only look at directly asserted information about resources in the dataset (i.e. no inference was performed before calculating these statistics).

2 Statistics

We first look at how many times both the namespace is declared and how many resources are of a given core type.

- The PROV namespace occurs in 1159 documents.
- Number of Entites: 1214
- Number of Activities: 283
- Number of Agents: 641

We also looked at the number of PROV edges that were used with the various documents.

- Number of wasDerivedFrom edges: 23088
- Number of used edges: 0
- Number of wasGeneratedBy edges: 0

- Number of wasAssociatedWith edges: 0
- Number of wasAttributedTo edges: 0

We also note that PROV has been extended by 8 other ontologies as calculated by manual inspection of the extensions of the various core classes as listed in the appendix.

3 Conclusion

This initial analysis shows some uptake within the Semantic Web community. However, while PROV is widely referenced within the community's literature, it appears, that direct usage of the standard could be improved (at least within the dataset represented by the LOD Laudromat). It should be noted that our analysis is preliminary and there is a much room for further work. In particular, we aim to look at the indirect usage of PROV through usage by ontologies that extend it (e.g. The Provenance Vocabulary) or that map to it such as Dublin Core or PAV. Understanding such indirect usage will help us better understand the true state of provenance interoperability within Linked Data. Likewise, it would be interesting to perform network analysis to understand the role that PROV plays within the Linked Data network.

References

- Beek, W. & Rietveld, L & Bazoobandi, H.R. & Wielemaker, J. & Schlobach, S.: LOD Laundromat: A Uniform Way of Publishing Other People's Dirty Data. Proceedings of the International Semantic Web Conference (2014).
- Rietveld, L. & Beek, W. & Schlobach, S.: LOD Lab: Experiments at LOD Scale. Proceedings of the International Semantic Web Conference (2015).

Appendix: Code

```
In [95]: import requests
         nsr = requests.get("http://index.lodlaundromat.org/ns2d/", params={"uri":'
         total_prov_docs = nsr.json()["totalResults"]
```

```
In [87]: nsr = requests.get("http://index.lodlaundromat.org/ns2d/", params={"uri":'
```

```
In [88]: import io
         from rdflib.namespace import RDFS, RDF
         from rdflib.namespace import Namespace
         from rdflib import Graph
         from rdflib import URIRef
         PROV = Namespace('http://www.w3.org/ns/prov#')
```

```
In [91]: entitySubclasses = []
         activitySubclasses = []
         agentSubclasses = []
         totalNumberOfEntities = 0
         totalNumberOfActivities = 0
```

```
totalNumberOfAgents = 0
numWasDerivedFrom = 0
numUsed = 0
numWGB = 0
numWAW = 0
numWasAttributedTo = 0

for doc in nsr.json()["results"]:
    #print(doc)
    headers = {'Accept': 'text/turtle'}
    x = requests.get("http://ldf.lodlaundromat.org/" + doc, headers=header
    txt_res = x.text
    tmpGraph = Graph()
    tmpGraph.parse(io.StringIO(txt_res), format="turtle")
    #print(doc + " " + str(len(tmpGraph)))
    for entityClass in tmpGraph.subjects(RDFS.subClassOf, PROV.Entity):
        #print(entityClass)
        entitySubclasses.append(entityClass)
    for entity in tmpGraph.subjects(RDF.type, PROV.Entity):
        totalNumberOfEntities = totalNumberOfEntities + 1

    for activityClass in tmpGraph.subjects(RDFS.subClassOf, PROV.Activity)
        #print(activityClass)
        activitySubclasses.append(activityClass)

    for activity in tmpGraph.subjects(RDF.type, PROV.Activity):
        totalNumberOfActivities = totalNumberOfActivities + 1

    for agentClass in tmpGraph.subjects(RDFS.subClassOf, PROV.Agent):
        #print(agentClass)
        agentSubclasses.append(agentClass)

    for agent in tmpGraph.subjects(RDF.type, PROV.Agent):
        totalNumberOfAgents = totalNumberOfAgents + 1

    ##look at relations

    for s,p,o in tmpGraph.triples( (None, PROV.wasDerivedFrom, None )):
        numWasDerivedFrom = numWasDerivedFrom + 1

    for s,p,o in tmpGraph.triples( (None, PROV.used, None )):
        numUsed = numUsed + 1

    for s,p,o in tmpGraph.triples( (None, PROV.wasGeneratedBy, None )):
        numWGB = numWGB + 1

    for s,p,o in tmpGraph.triples( (None, PROV.wasAssociatedWith, None ))
        numWAW = numWAW + 1
```

```
            for s,p,o in tmpGraph.triples( (None, PROV.wasAttributedTo, None) ):
                numWasAttributedTo = numWasAttributedTo + 1
```

```
In [126]: from IPython.display import display, Markdown

          output = "### Statistics \n"
          output += "We first look at how many times both the namespace is declarec
          output += "* The PROV namespace occurs in " + str(total_prov_docs) + " dc
          output += "* Number of Entites: " + str(totalNumberOfEntities) + "\n"
          output += "* Number of Activities: " + str(totalNumberOfActivities) + "\r
          output += "* Number of Agents: " + str(totalNumberOfAgents) + "\n\n"

          output += "We also looked at the number of PROV edges that were used with
          output += "* Number of wasDerivedFrom edges: " + str(numWasDerivedFrom) +
          output += "* Number of used edges: " + str(numUsed) + "\n"
          output += "* Number of wasGeneratedBy edges: " + str(numWGB) + "\n"
          output += "* Number of wasAssociatedWith edges: " + str(numWAW) + "\n"
          output += "* Number of wasAttributedTo edges: " + str(numWasAttributedTo)

          display(Markdown(output))
```

Appendix: Classes that subclass a PROV core class

```
In [120]: print("Subclasses of Entity")
          for i in entitySubclasses:
              print(i)
          print("Subclasses of Activity")
          for i in activitySubclasses:
              print(i)
          print("Subclasses of Agent")
          for i in agentSubclasses:
              print(i)
```

```
Subclasses of Entity
http://www.gsi.dit.upm.es/ontologies/marl/ns#Opinion
http://purl.org/net/p-plan#Entity
http://www.w3.org/ns/prov#Plan
http://www.w3.org/ns/prov#Bundle
http://www.w3.org/ns/prov#Collection
http://www.opmw.org/ontology/WorkflowExecutionArtifact
http://purl.org/twc/vocab/vsr#Color
http://www.co-ode.org/ontologies/ont.owl#Graphic
```

```
http://purl.org/twc/vocab/vsr#Root
http://purl.org/twc/vocab/vsr#Color
http://www.co-ode.org/ontologies/ont.owl#Graphic
http://purl.org/twc/vocab/vsr#Root
http://purl.org/net/provenance/ns#DataItem
http://purl.org/net/provenance/ns#File
http://purl.org/net/provenance/ns#Immutable
http://purl.org/net/provenance/ns#File
http://purl.org/net/provenance/ns#Immutable
http://purl.org/net/provenance/ns#DataItem
Subclasses of Activity
http://www.gsi.dit.upm.es/ontologies/marl/ns#SentimentAnalysis
http://spitfire-project.eu/ontology/ns/Activity
http://www.w3.org/ns/org#ChangeEvent
http://purl.org/net/p-plan#Activity
http://www.opmw.org/ontology/WorkflowExecutionProcess
http://www.w3.org/ns/org#ChangeEvent
http://purl.org/net/provenance/ns#DataCreation
http://purl.org/net/provenance/ns#DataAccess
http://purl.org/net/provenance/ns#DataCreation
http://purl.org/net/provenance/ns#DataAccess
Subclasses of Agent
http://spitfire-project.eu/ontology/ns/Agent
http://purl.org/net/provenance/types#DataCreator
http://purl.org/net/provenance/ns#HumanAgent
http://purl.org/net/provenance/ns#HumanAgent
```