Visualizing Provenance using Comics

Andreas Schreiber
German Aerospace Center (DLR)

Regina Struminski
University of Applied Science Düsseldorf
Introduction

Deutsches Zentrum
für Luft- und Raumfahrt
German Aerospace Center
Simulation and Software Technology, Cologne/Berlin
Head of Intelligent and Distributed Systems department
Institute of Data Science, Jena
Head of Secure Software Engineering group

Co-Founder
Data Scientist
Patient
Motivation – Use Cases

Quantified Self (n = 1 participant)

Medical Trials (n > 1 participants)
Motivation – Use Cases

Telemedicine

Medical experiments
Understand, how Quantified Self data has been produced, processed, stored, accessed, ...

Pictures from Breakout Session on Mapping Data Access (2014 QS Europe Conference, Amsterdam)
https://forum.quantifiedself.com/t/breakout-mapping-data-access/995
Example: Weight Tracking Workflow

Devices

- Withings weight scale

Applications

- Withings Health Mate App
- Weight Companion App
- Twitter App
- Python Script

Services

- Withings Cloud API
- Twitter API
- Zenobase Web App & API

User

- Step on
- Display
- Import
- Generate
- Export
- Request
- Notify
- Share
Questions related to Quantified Self Data and Activities

Data
- What data about the user were created during the activity X?
- What data about the user were automatically generated?
- What data about the user were derived from manual input?

Apps and Services
- Which activities support visualization of the users data?
- In which activities can the user input data?
- What processes are communicating data?

Access and Privacy
- What parties were involved in generating data X?
- What parties got access on data X?
- Can other parties see user’s data X?
Provenance Model for Quantified Self

Sub models for basic Activities

• Input
• Sensing
• Export
• Request
• Aggregate
• Visualize

The activities generate or change data that is associated or attributed to Agents

• Users
• Software
• Organizations


<prefix userdata "http://software.dlr.de/qs/userdata/"/>
<prefix qs "http://software.dlr.de/qs/"/></prefix>
<prefix graphic "http://software.dlr.de/qs/graphic/"/></prefix>
<prefix app "http://software.dlr.de/qs/app/"/></prefix>
<prefix user "http://software.dlr.de/qs/user/"/></prefix>
<prefix device "http://software.dlr.de/qs/device/"/></prefix>
<prefix method "http://www.java.com"/></prefix>

wasGeneratedBy(qs:graphic/diagram, method:visualize, 2016-12-01T16:06:22+00:00, [prov:role="displaying"]
activity(method:visualize, 2016-12-01T16:06:21+00:00, 2016-12-01T16:06:22+00:00)
entity(qs:graphic/diagram, [prov:type="linechart", prov:label="Line chart"])
entity(userdata:steps, [prov:type="steps", prov:label="Steps database"])
agent(qs:user/regina@example.org, [prov:type="prov:Person", prov:label="Regina Struminski"])
agent(qs:app/stepcounter, [prov:type="prov:SoftwareAgent", qs:device="smartphone", prov:label="StepCounter"])
wasAttributedTo(qs:graphic/diagram, qs:user/regina@example.org)
wasAttributedTo(userdata:steps, qs:user/regina@example.org)
actedOnBehalfOf(qs:app/stepcounter, qs:user/regina@example.org, -)
used(method:visualize, userdata:steps, 2016-12-01T16:06:21+00:00)
wasDerivedFrom(qs:graphic/diagram, userdata:steps, -, -, -)
wasAssociatedWith(method:visualize, qs:app/stepcounter, -)

endDocument
Standard Graph Visualizations and Textual Representations of Provenance Data are not Easy to Understand by Non-experts
Idea: Provenance Visualization Using Comics

Provenance Comics

• Presenting the provenance of processes in visual representation that people can understand without prior instructions or training (“Provenance for people”)

• Assumption

  • People are familiar with comics from every day life
  • See *daily strips* in newspapers etc.
Provenance Comics

Design considerations

• Data provenance has a temporal aspect: origin, manipulation, transformation, and other activities happen sequentially over time

• The directed acyclic provenance graph guarantees that, while moving through its nodes, one always moves linearly forward in time

• It’s possible to derive a temporal sequence of happenings from the graph that can be narrated like a story

Mapping provenance graph to comics

• We generate a comic strip for each basic activity in the provenance graph

• Each strip consists of a varying number of panels, which are small drawings that provide further details about the activity

• The complete set of comic strips shows the “story” of the data
First Sketches

February 23, 2016

Hi, I'm Onyame!

I use JSON to track my changes. Now I'd like to show them in a diagram.
First Sketches

Rome time ago...

Onyame recorded his weight.

Weights

The weight data was uploaded to a cloud.

June 6, 2016

The WeightCompanion App did an export:
It fetched the weight data from the cloud...

... and created a CBV file from it.

Onyame now has a local copy of his weight data.
Current Graphical Style
Single Comic Strip Shows a Single Data-related Action
Communicate to People Where Data is Stored
Understand How Data is Analyzed
Distinctive Features

- Shapes
- Colors
- Icons
- Letters
- Labels
Representation of PROV Elements

Agents

Entities

Activity-related
Collecting QS Provenance

Weight Tracking App

Collecting QS Provenance Visualization with Python Script

```python
# Provenance-related Imports
from prov.model import ProvDocument, PROV
from provstore.api import Api
from time import gmtime, strftime

# Create a new provenance document
prov = ProvDocument()

# Add namespaces
prov.add_namespace('qs', 'http://software.dlr.de/qs/')
prov.add_namespace('userdata', 'http://software.dlr.de/qs/userdata/')
prov.add_namespace('user', 'http://software.dlr.de/qs/user/')
prov.add_namespace('graphic', 'http://software.dlr.de/qs/graphic/')
prov.add_namespace('library', 'https://pypi.python.org/pypi/)
prov.add_namespace('python_method', 'http://www.python.org/)

# The user
agent_user = prov.agent('user:onyame@gmail.com', {'prov:type': PROV['Person']})

# Application Import
from pandas import DataFrame, Series, read_csv
import matplotlib.pyplot as plt
prov.entity(('library:pandas', {'library:version': '0.17.2'})
prov.entity(('library:matplotlib', {'library:version': '1.5.3'})

# Import weights from CSV file
WC_FILE = 'WeightReport-3-2-21-31.34.44.csv'
entity_csvfile = prov.entity('userdata:WeightReport-3-2-21-31.34.44.csv')
prov.wasAttributedTo(entity_csvfile, agent_user)
wc_data = read_csv(WC_FILE, parse_dates=True, index_col='Time

# Get just the weights
weights = wc_data[['Weight']]
entity_weights = prov.entity('userdata:WeightReport-3-2-21-31.34.44.csv')
prov.activity('python_method:read_csv',
prov.wasGeneratedBy(entity_weights, strftime('%Y-%m-%d %H:%M:%S'))
prov.wasDerivedFrom(entity_weights, other_attributes=['prov:type:', prov.used('python_method:read_csv')]

# Plot the weights
weights.plot(title='Weight', legend=True
entity_plot_weights = prov.entity('Writer')
```
PROV Comics
Web Application

http://provcomics.de

- Implemented in JavaScript
- Single page website
- Reads provenance graph from PROVSTORE
- Uses PROVSTORE jQuery API
- Code:
  http://github.com/DLR-SC/prov-comics
Implementation Details

Additional attributes

agent(qs:app/stepcounter, [prov:type="prov:SoftwareAgent", qs:device="smartphone", prov:label="StepCounter"])
agent(qs:service/fitbit, [prov:type="prov:Organization", prov:label="Fitbit"])
wsgeneratedby(userdata:activities/steps, method:request, 2016-12-01T16:06:22+00:00, [prov:role="uploading"])

http://provcomics.de/?username=rstruminski&docId=115547
Open Issues

Current implementation is a prototype *with limitations*

- Flexibility and generalization
- Handling of
  - large provenance graphs
  - incomplete provenance data
  - branches and multiple data sources
- Expects a single PROV document
Future Work and Use Cases

Future Work

• Different comic styles
  • Comparative user studies

• Quantitative comics
  • Geographical information
  • Glyph-based depiction

• Technical improvements
  • Large Provenance graphs
  • Provenance templates
  • “Intelligent” generation of pictures

Possible Use Cases

• Journalism
• Generation of handbooks
• Communicating incidents
Thank You!

Andreas Schreiber
www.DLR.de/sc/ivs
andreas.schreiber@dlr.de
@onyame