Provenance of Data and Metadata

Session 3A: Tuesday, October 18, 2016 14:00 – 16:45

Contributors: Jay Greenfield, Steve McEachern, Daniella Meeker, Nicholas Car, Darren Bell, Larry Hoyle, Jared Lyle, Gregg Kellogg, Eric Prud'hommeaux, Joachim Wackerow, Michelle Edwards

What is Provenance?

Provenance can be defined as the "change metadata", where changes in things you are interested in are recorded. The primary purpose of provenance, from a preservation perspective, is to lend understanding to the processes used to generate an object throughout its life cycle which may then give it a measure of integrity. It is a record of facts: who, what, where, when, and possibly why, changes are made.

There are different granularities of provenance ranging from the least granular – describing facts about whole datasets – to a granularity at an element level – describing facts about sub-parts of a dataset. At the least granular level, provenance can be used to establish Chain of Custody (as per the non-computer science definition familiar with the provenance of artwork: who made it, who owned it etc.). finer granular provenance can assist with data production transparency and ultimately facilitate reproducibility. A way of viewing these granularities is on a spectrum

Goals of provenance will depend on your goals. What pieces of information are important for you to keep? What elements of provenance does DDI want to claim? What are the objectives of DDI when it comes to provenance?

In an archive situation, we are interested in the historical (what has already happened), we are also interested in what is currently happening, and in the future, we are interested in capturing what will happen. Historical, current, derive what will happen. Provenance about things that happen to a particular dataset after its production, such as how that dataset is used in derived datasets or for publications, is called 'forward provenance' from the point of view of the particular dataset.

Key items to consider when thinking about using provenance, if it is valuable to do, then use provenance. If there is no added value, then do not use! Always back to the goals!

What items should be considered in scope when considering provenance?

Anything may be considered in scope, however, this would be impossible to manage. A good starting point are when items become digital. This can include digital-born items as well as digitized items. Example would be in an archive, digitize codebooks and/or older documentation – this is where provenance would begin. Challenges will still exist however, changes to a survey instrument as an

example or within a survey firm, a paper survey is created, then scanned, and reprinted – where would or should provenance begin? Remember, to always go back to the goals and whether capturing the provenance of the item makes sense and has value.

Another raised concern was: are we collecting enough information to move from transparency to reproducibility as outlined above? Example included was the protocol used to collect information used in the CPI basket of products (http://www.bls.gov/cpi/). One year you may be collecting prices on striped shirts, but the following year it may be only coloured shirts. Are these changes in protocol easily captured by provenance to ensure that we can attain a level of reproducibility? The answer is yes (highlight how this can be accomplished in later example by using a set of instructions).

Use cases for provenance

1. Data Life cycle
2. Microdata to aggregate data
3. Changes to codelists

2) Microdata to aggregate data

Sample data collected as a microdata file – unit records:

1	m	s
2	m	s
3	m	s
4	m	s
5	f	s
6	f	n
7	m	n
8	m	n
9	m	n
10	m	n
11	m	n

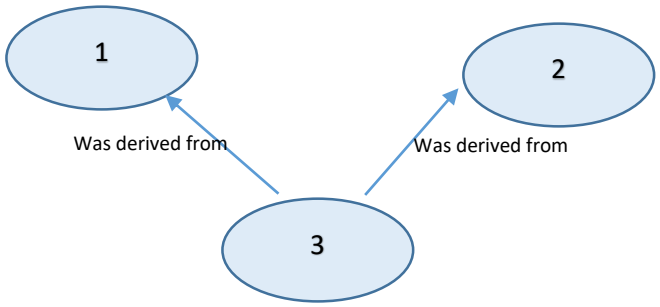
PROV Classes

Aggregated to create a cross tabulation table:

	m	f
s	4	1
n	5	1

Two ways to capture this using PROV.

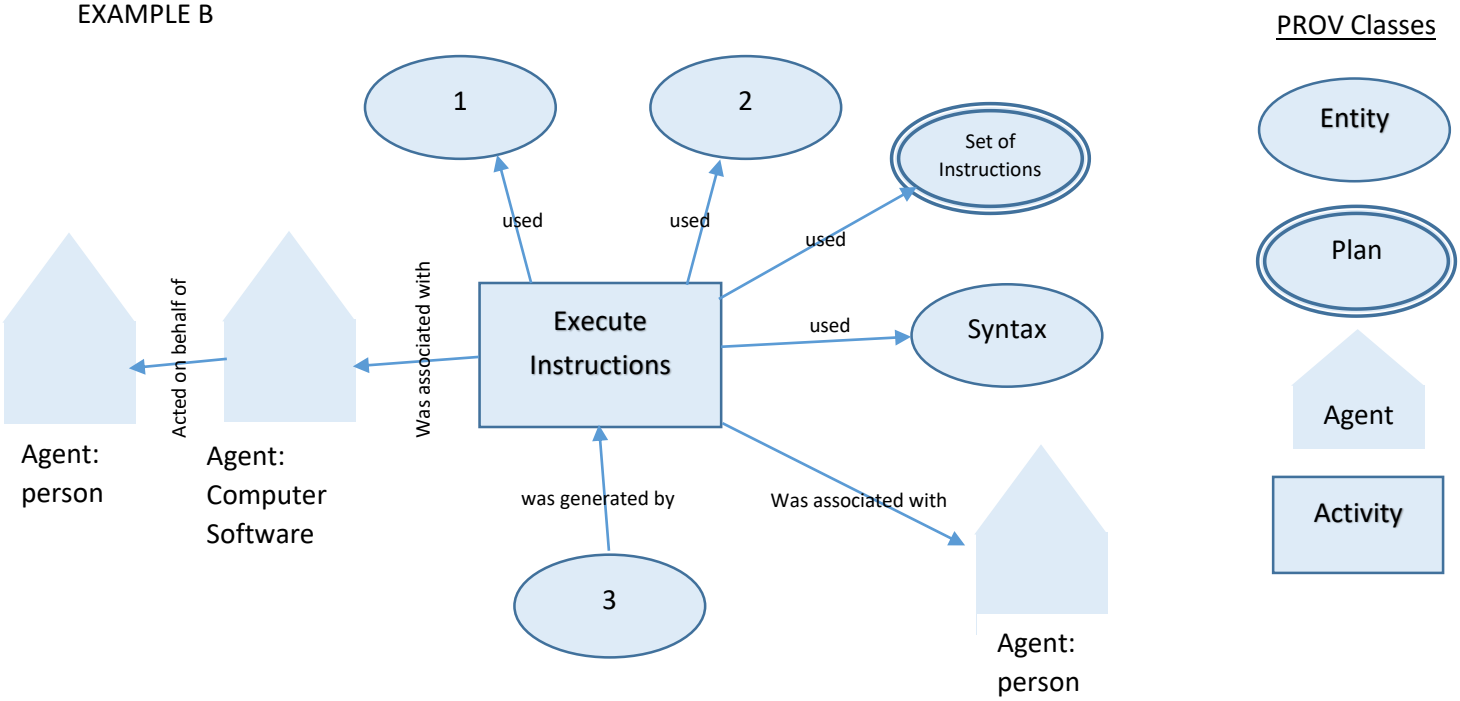
EXAMPLE A



- There are 3 entities, we know that entity #3 came from entity #1 and entity #2
- We know that #3 was "influenced" by #1 and #2, but we do not know how

PROV Classes

EXAMPLE B

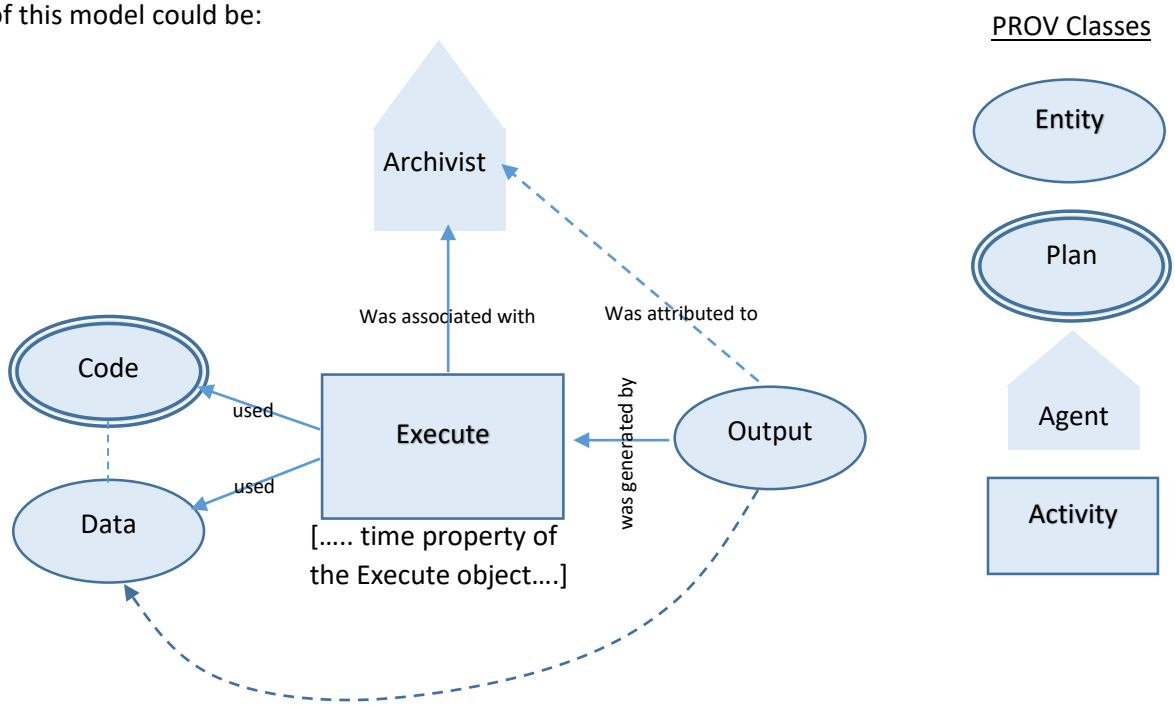


- Start with 2 entities (1=N and 2=S)
- Invoke an Activity – an event following some plan (a set of instructions) – in this case, aggregation instructions
- We can a specific syntax to use – by adding an entity called Syntax in this example
- We have an Agent in this example which will specify the computer and/or software
- We can also have another Agent – a particular person which will use the given software: the software agent then acted on behalf of the person, from the dataset production point of view

NOTES:

- If the entities are very well defined, then we can infer from one entity to another
- Loose relationships are a good place to start
- Always comes down to “what you want to know about X”
- If we have missing values in our microdata we can add another entity to our PROV model that will list restraints – therefore the activity may not allow missing values
- An entity is NOT a transient occurrence of a “thing” such as an event

Another view of this model could be:



NOTES:

- We can infer some additional relationships from basic relationships, for instance that the Output "was derived from" the input Data and that the Output "was attributed to" the Agent who invoked the Activity that generated it, in this case the Archivist who invoked the Execute.
- We can also infer that the Code was used against the Data by going from Code to an Activity and from that Activity back to some Data
- Object property – represent where everything is
- Subclass of code could be a prov Plan: subclasses specialize something so potential DDI subclasses of PROV classes will allow DDI to specialize items in PROV and to add constraints to them
- Can create subclass as required and subproperties which are specialized properties, e.g. perhaps "executed code" as a specialized form of "used" for an Activity using a Plan
- Properties of the execution event, such as the start and end times of execution are simple properties of the Activity object

This is a pattern used by the PROV model

Remember use the minimum set of "stuff" to allow you to make meaning of your model.

Difference between a Process step in DDI4 model vs. an Activity in the PROV model?

The current definition used in DDI4 process model of a process step can be defined as a subclass of an Activity in the PROV model. If it is possible to record a step in the DDI4 model, then record is as an Activity in PROV. However, there will be situations where the instructions aspect of a process step will need to be pulled apart and listed as a set of rules (a PROV Plan) or a 'named Activity' in PROV which could be an Activity of a known type, the occurrence of which indicated a certain type of step with known details.

Different layers of PROV and their relations to the DDI4 model

1. PROV-DM – conceptual details of generic provenance
2. PROV-O – implementation of the PROV-DM using OWL (Web Ontology Language) modelling
3. DDI4 model – domain-specific model

At any point in time, the reader should be able to review the documentation and move to a level where they have a comfort in understanding of what is happening. Individuals may begin at the DDI4 model level, but move up to the PROV level where items are generically represented and thus have a lower level of detail but where general patterns of execution, derivation and attribution are still represented. The notion is that anyone should be able to review and obtain the information at their level of understanding.

PROV-DM, or PROV Data Model, is something that the DDI4 community should review to identify opportunities for better interoperability by matching DDI concepts to classes and properties in PROV-DM. A DDI OWL model would allow the direct association of DD classes with PROV classes.