Directed Qualified Pattern, Influence, Non-Influence Relations, Optional Attributes

Luc Moreau[1]

University of Southampton, UK
l.moreau@ecs.soton.ac.uk

Abstract. For computationally intensive applications or applications running for a long time, provenance can become very quickly difficult to handle. Being able to summarise provenance and identify common patterns and outliers become critical. A provenance summary [4] is a summary of a provenance graph, capturing the salient aspects of this graph. Such a summary is itself a provenance graph expressed with the PROV data model extended with count information, indicating how often a kind of node or edge can be found in the original graph. It is natural for such a counting information to be expressed as attributes to the corresponding nodes and edges. This attribute is domain specific and therefore requires PROV extensibility capability. However, PROV does not allow attributes for SpecializationOf, AlternateOf, and HadMember, which prevents them to be handled properly.

1 Subject

Lack of extensibility of SpecializationOf, AlternateOf, HadMember. Conflation of the semantic notion of influence and the syntactic directed qualified pattern.

2 Application

Provenance Summaries have been used with SmartShare and a food provenance application.

3 Background

Provenance summaries [4] require domain specific attributes to be added to all PROV constructs, e.g. Entity, Activity, WasDerivedFrom, etc. These attributes give an indication of the number of such similar constructs in a graph that is being summarised. Among the relations that need to be enriched with such domain-specific attributes, we find SpecializationOf, AlternateOf, and HadMember. PROV does not allow such relations to be extended in an inter-operable way.

In PROV-DM [2], all other terms are equipped with optional attribute-value pairs "representing additional information" about the objects such terms describe, except for SpecializationOf, AlternateOf, and HadMember, which do not allow for such attribute-value pairs.

In PROV-O [3], most properties are accompanied by a form of reification following the "directed qualified pattern", allowing other attributes to be "hooked". The directed qualified pattern is not applied to SpecializationOf, AlternateOf, and HadMember.

In fact, PROV even seems to conflate the directed qualified pattern and the notion of influence, stating that each of SpecializationOf, AlternateOf, HadMember "is not, as defined here, also an influence, and therefore does not have an id and attributes".

The directed qualified pattern is a reification style that can have applications beyond PROV, and has been suggested by commentators to be defined outside the PROV specifications so that it can be reusable by other communities. Influence [2]is the "capacity of an entity, activity, or agent to have an effect on the character, development, or behavior of another": it is a semantic notion, it should not be mixed with a syntactic construct.

4 Solution

1. Define the directed qualified pattern in a standalone fashion, independently of the notion of influence in PROV
2. Make Influence follow the Directed Qualified Pattern.
3. Make SpecializationOf, AlternateOf, HadMember extensible. This means, give them optional identifiers and attributes in PROV-DM, and make them follow the Directed Qualified Pattern in PROV-O. SpecializationOf, AlternateOf, HadMember are not an influence.

5 Solution Rationale

This solution makes the model more uniform. It allows interoperable subtyping of these relations. It allows the Directed Qualified Pattern to be used in other contexts.

References

1. Luc Moreau and Paolo Missier. (2013). PROV-N: The Provenance Notation. W3C Recommendation.. http://www.w3.org/TR/2013/REC-prov-n-20130430/
2. Luc Moreau and Paolo Missier. (2013). The PROV Data Model. W3C Recommendation.. http://www.w3.org/TR/2013/REC-prov-dm-20130430/
3. Timothy Lebo, Satya Sahoo and Deborah McGuiness. (2013). PROV-O: The PROV Ontology. W3C Recommendation.. http://www.w3.org/TR/2013/REC-prov-o-20130430/
4. Luc Moreau. Aggregation by provenance types: A technique for summarising provenance graphs. In Graphs as Models 2015 (An ETAPS'15 workshop), Electronic Proceedings in Theoretical Computer Science, pages 129-144, London, UK, April 2015. http://dx.doi.org/10.4204/EPTCS.181.9