Provenance: The Missing Component of the Semantic Web for Privacy and Trust

Harry Halpin
H.Halpin@ed.ac.uk

School of Informatics
University of Edinburgh
2 Buccleuch Place
EH8 9LW Edinburgh
Scotland, UK

Abstract. Data on the Semantic Web currently does not have any standardized or any *de-facto* agreed upon way to exhibit provenance information, yet provenance is the foundation for any reasonable model of privacy and trust. Yet, currently every RDF triple does not have any coherent way of storing provenance information on the Semantic Web. We present the hypothesis that provenance is by far the most important data needed on the Semantic Web for privacy and trust, and review previous work in database systems on provenance. We put forward the concept that the three main provenances operators (insertion, deletion, and copy) from provenance work in database systems can be used on the Semantic Web. Furthermore, we hypothesize that such information naturally should be stored in or using the name URI of named graphs. We show that such an approach can help solve practical issues of privacy and trust in social networks using a real-world example.

Keywords: *provenance, trust, Semantic Web*

1 A Theory of Provenance

Provenance has remained for the most part an undeveloped area of research, both in traditional database systems and on the Semantic Web. The most well-known approach presented by Berners-Lee, Kagal, and others [1] attempts to capture provenance information in terms of proofs using a Datalog-like language [7]. An alternative approach to provenance has been that of 'RDF Molecules' by Ding et al., which proposes a level of granularity (the 'molecule') that allows the original RDF statements to be re-constructed from disparate graphs [8]. Neither of these approaches have reached large-scale usage on the Semantic Web, much less standardization, and both approaches present a number of disadvantages. While the approach of RDF molecules allows one to reconstruct a graph, it does not allow the tracing of provenance of the graph if any of the information in the graph changes. Despite appeals to the fact that 'cool' URIs should not change, the data hosted at URIs *will* change. For example, in my social-networking profile my organizational affiliation will likely change over time. How can we keep track of this? Proof-based systems around a variant of Datalog rules are too heavy-weight. Is it not odd that one can stick to standard RDF for describing data, but then move to a more expressive query and rule language to explain provenance? Also, the two most dominant Datalog-like Semantic Web languages, the upcoming W3C RIF language [2] and the N3 language [1], both present the syntactic problem that they themselves are not easily represented as RDF. While the W3C RIF language has been built so that it can deal with RDF data, the standard syntax of RIF is itself in idiosyncratic XML. N3 has its own syntax, that while close to the Turtle syntax for RDF, expands RDF in a number of ways that makes it semantically incompatible with RDF. Work on the Open Provenance Model for workflows is similar to RDF but seems to attempt to prematurely optimize an entire range of provenance operations without determining whether or not these operators can be derived from a few simple operators or are even necessary [12]. It seems a more advantageous route to some sort of provenance information would be to create a minimal vocabulary for provenance that would be expressible in standard RDF.

Recent work in provenance on relational databases has aimed precisely at creating such a minimal vocabulary, albeit for traditional relational data rather than RDF. In particular, the foundational work on provenance distinguishes between two kinds of provenance, the *where* provenance, which is the "locations in the source databases from which the data was extracted," and the *why* provenance, which is "the source data that had some influence on the existence of the data" [4]. Buneman et al. [4] present a Datalog-based model-theoretic semantics for calculating this kind of *where* and *why* provenance over queries. However, the traditional database community has been confronted with the same issues at the Semantic Web community with regards Datalog, as it would be better for most databases to keep the provenance in pure relational data. Therefore, current theoretical database work attempts to create more realistic models of provenance whose formal semantics do not rely on Datalog yet can still trace both the *where* and *why* provenance and can be implemented on top of run-of-the-mill SQL databases [3].

The most simple and accessible work from the database community is focused on providing a simple update language that tracks provenance, by focusing on three primary *provenance operators*: *insertion* (ins), *deletion* (del), and *copy* (copy). This is given by the grammar u, where for given fields in database q and p and a specific value v and a specific field a, $u ::= (ins\ a : v\ into\ p) \vee (del\ a\ from\ p) \vee (copy\ q\ into\ p)$ [3]. Buneman et al. considers this a 'cut-and-paste' model that can take into account the movement of data [3]. Such a model can then be easily implemented on top of normal databases by expanding the database so that sections of the database can store their provenance information, called the *provenance trace.* This trace can be queried, such that a user should be able to query the ultimate 'source' of some data and its change history. Implementation-wise, every sector of rows and columns that has either been changed or copied from another database can have its provenance tracked by expanding the database with further rows and columns. So, whenever an insert, delete, or copy operation is committed, the corresponding provenance trace is tracked with identifiers for the source and target. A simple approach that records provenance with every interaction that changes the state of the database would lead to an explosion of database size, so current research is studying ways of optimizing provenance storage [3]. In an attempt to find a more solid semantic foundation for provenance, Cheney et al. [6] proposed a semantic characterization of provenance using functional dependency analysis, although they also proved that such a minimal dependency provenance is not computable, although dynamic and static techniques can approximate it. Another alternative for the semantics has been suggested by Green et al. [10], who used semi-rings to generalize algorithms over both *why*-provenance and relational algebras.

2 A Provenance Framework for the Semantic Web

It is our hypothesis that a simple vocabulary, composed of *insert*, *delete*, and *copy* operations as introduced by Buneman et al. provides a flexible format for provenance on the Semantic Web [3]. The current activity in the database community can then be used by the Semantic Web community in order to bootstrap a realistic implementation and scalable model of provenance. However, a few design issues are needed to be made in order to make the database approach compatible with the Semantic Web.

The first design choice is that the provenance operations be rephrased to operate over graphs rather than traditional relational data. Another question is whether or not separate *ins* and *copy* operators are needed by the Semantic Web. The *ins* operator in the context of databases (including XML databases) inserts a specific value, which may be a new value not in the graph or a replacement of a current value, at a determined location in the graph, while *copy* merely copies an entire graph. Thus, a *ins* function should be employed when one wishes to actually change or 'update' a given particular graph, as when changing the subject literal of a graph, while the *copy* function merely changes the name (often the host) URI of a graph or merges graphs, not necessarily changing any values. So, re-phrasing the provenance operator grammar u into RDF, given source graph G and target graph T as well as graph update a, u *::= (ins* a *into* T*)* $\vee$ *(del* G *from* T*)* $\vee$ *(copy* G *into* T*)*. Traditional graph merge is then just when *ins* a into T without any deletion and where at least one node x is shared by at least one a and T.

The problem with phrasing provenance operators in RDF is that for the most part they deal with operations amongst entire graphs, and RDF lacks official support for identifying graphs. However, there is unofficial yet widely implemented support in the form of *named graphs*, where each graph G is then identified with a *name URI* [5]. The idea of using named graphs for provenance is not new [5], but previously has been restricted to assertions that identify who is making a claim, not the provenance story championed by Buneman et al. [3]. If we assume the implementation of named graphs and the restriction of insertion values to (possibly typed) literal values, then we can phrase each of the provenance operations as RDF properties between the URIs of named graphs. The named graph approach solves the problem of *where* to store the provenance traces. Obviously, the provenance stores for each graph should be accessible, ideally using Linked Data principles, from the name URI of the named graph. This allows the name URI to ideally also serve as a SPARQL endpoint through which provenance queries can be done.

3 An Example of Provenance for Privacy and Trust

Most current efforts at privacy and trust within social networks make two assumptions. The first is that they assume that privacy can be handled via some sort of access control language, such as the unstandardized access language or an ontology like the Rei ontology developed by Kagal [11]. Assuming that policy languages can be phrased in terms of access control makes a certain amount of sense. Yet at minimum, this viewpoint is predicated upon certain behavior being either explicitly allowed or not, with the main behavior likely being the copying (equivalent to 'viewing,' as with a view one can doubtless copy), insertion, and deletion operations of the provenance. So, any access control language should operate over at least the three provenance operators. Second, Semantic Web research like Goldbeck's Trust Ontology makes the radically simplistic assumption that trust can be quantified as integer-valued rating between 1 to 10 (or some other arbitrary numbers, such as real-valued rating between 0 and 1) [9]. Where do such ratings come from? Obviously, they may come from the often unreliable subjective impression of the users. It is far better to calculate 'trust' from actual data sources and its changes. Therefore, it is of utmost concern to track the provenance of the data, and the use of provenance traces with named graphs allow such trust ratings to be calculated in a principled and algorithmic manner, and then privacy constraints built on top of such trust ratings.

For a quick example, we will use the infamous real-world 'dump your pen friend' scenario.[1] A young student has her picture taken and uploaded by a friend to Flickr with a Creative Commons license that allows commercial use. A major company finds the photo on Flickr and proceeds to use it in an advertising campaign across Australia (where the student lives), with an unflattering slogan beneath the photo that says "Dump Your Pen Friend." The young student feels emotionally damaged and sues the company in court. The problem at hand can be confronted one of two ways. The first is to assume that the owner of the photo

[1] *http://ipandentertainmentlaw.wordpress.com/2007/10/15/update-dumb-your-pen-friend/*

perhaps did not fully understand what situation they were putting their friend in by releasing their photo under the Creative Commons License, in which case all that is needed is to offer a more full explanation of the license. However, it is more likely to assume that the original owner of the photo understood Creative Commons, but did not notify their friend that an unflattering picture of them was being uploaded that could be used for-profit by a company, and the company did not alert the young student. However, in order to both determine who was in the picture and the chain of events of copying the photo that led to the incident, what precisely is needed is *provenance*.

One can see how a provenance framework could have led to a solution to this scenario. When the photo was originally taken, the provenance information about the date the picture was taken and its owner could be taken, phrased as FOAF RDF statements involving the URI of the picture, using *ex* as the *http://www.example.org/* namespace, *foaf:* is *http://xmlns.com/foaf/spec/*, *prov:* as a yet undetermined provenance namespace, and all examples given using the Turtle syntax (assuming the student whose picture was taken is 'Jane Doe' and the taker of the picture was 'John Doe'). So the graph *ex:photo.jpg rdf:type foaf:Image, foaf:maker ex:JohnDoe* could be given a name URI *ex:thephoto#it*. When the picture was uploaded to a popular social-networking site, the student could have seen themselves in the photo and so 'tagged' themselves in the photo, and thus inserted a new value using *prov:ins*, with the new value being the triple *ex:newtriple* which contains *ex:photo.jpg foaf:depicts ex:JaneDoe*. This provenance information is recorded using *ex:/photo#it prov:ins ex:newtriple*. After the photo was uploaded, the company then copied the photo, and this is stored in the provenance trace of the named graph by using the *prov:copy* operator. Assuming the company's URI to be *http://thecompany.info*, this could be recorded as *ex:photo#it prov:copy http://thecompany.info/copyOf/photo#it*. Relevant temporal information could also be added to the named graph records in the provenance trace. Therefore, the provenance trace of the photo could be followed, so that before the photo was used in a nation-wide advertising campaign, the company and student both could determine who was in the photo.

4 Conclusion and Future Work

We have argued so far in this paper that a simple model of provenance, based around the central three provenance operators and the storage of provenance traces, can be used on the Semantic Web by merging it with an approach based on named graphs. This is at best a sketch towards the completion of a provenance-aware Semantic Web. A small vocabulary of provenance operators need to be standardized and given a namespace URI. Second, the interaction of provenance operators and traces with RDF and named graphs needs to be given a formal semantics and implemented. This step may not be as difficult as it seems, as simple operations such as 'copy' or *where* provenance are already implemented by many as the default use of named graphs, and the deletion and copy operators are already implemented in the form of the proposed SPARQL Update language [14]. There has already been work on the formal semantics for provenance operators and traces within the nested relational calculus [6] as well as giving RDF provenance over named graphs a formal semantics [13]. So, a practical RDF vocabulary for provenance with a formal semantics should be possible to deploy and standardize.

References

1. T. Berners-Lee, D. Connolly, L. Kagal, Y. Scharf, and J. Hendler. N3 Logic: A logic for the Web. *Journal of Theory and Practice of Logic Programming*, pages 249–269, 2007.
2. H. Boley and M. Kifer. RIF Basic Logic Dialect. Working draft, W3C, 2008. http://www.w3.org/TR/rif-bld/ (Last accessed Aug. 8th 2008.
3. P. Buneman, A. Chapman, and J. Cheney. Provenance management in curated databases. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 539–550, New York, NY, USA, 2006. ACM Press.
4. P. Buneman, S. Khanna, and W. chiew Tan. Why and where: A characterization of data provenance. In *In ICDT*, pages 316–330. Springer, 2001.
5. J. Carroll, C. Bizer, P. Hayes, and P. Stickler. Named graphs. *Journal of Web Semantics*, 4(3):247–267, 2005.
6. J. Cheney, A. Ahmed, and U. A. Acar. Provenance as dependency analysis. *CoRR*, abs/0708.2173, 2007.
7. P. daSilva, D. McGuinness, and R. Fikes. A proof markup language for semantic web services. *Information Systems*, 31:381–395, 2006.
8. L. Ding, T. Finin, Y. Peng, P. P. da Silva, and D. L. McGuinness. Tracking RDF graph provenance using rdf molecules. In *Proc. of the 4th International Semantic Web Conference (Poster)*, 2005.
9. J. Goldbeck. Trust ontology, 2006. http://trust.mindswap.org/ont/trust.owl (Last Accessed March 5th 2009.
10. T. J. Green, G. Karvounarakis, and V. Tannen. Provenance semirings. In *PODS '07: Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 31–40, New York, NY, USA, 2007. ACM.
11. L. Kagal, T. Berners-Lee, D. Connolly, and D. Weitzner. Using Semantic Web Technologies for Open Policy Management on the Web. In *Proceedings of the National Conference of the Association for Artificial Intelligence*, 2006.
12. L. Moreau, J. Freire, J. Futrelle, R. McGrath, J. Myers, and P. Paulson. The open provenance model, December 2007.
13. P. Pediaditis, G. Flouris, I. Fundulaki, and V. Christophides. On explicit provenance management in rdf/s graphs. In *Workshop on the Theory and Practice of Provenance*, 2009.
14. A. Seaborne, G. Manjunath, C. Bizer, J. Breslin, S. Das, I. Davis, S. Harris, K. Idehen, O. Corby, K. Kjernsmo, and B. Nowack. SPARQL update: A language for updating RDF graphs. Member submission, W3C, 2008. http://www.w3.org/Submission/2008/SUBM-SPARQL-Update-20080715/ (Last accessed March 10th 2009).