SUMMARY

LITERATURE REVIEW

PROVENANCE FOR THE SEMANTIC WEB

In his book *Weaving the Web: the original design and ultimate destiny of the World Wide Web*, Tim Berners Lee, the inventor of the WWW, states:

> *I have a dream for the Web [in which computers] become capable of analysing all the data on the Web – the content, links, and transactions between people and computers. A "Semantic Web", which makes this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The "intelligent agents" people have touted for ages will finally materialize*. (Berners-Lee, Weaving the Web: the original design and ultimate destiny of the World Wide Web, 1999)

In this vision, which represents the first formulation of the Semantic Web, it is already possible to identify the criticality that in the following years would have led to discuss the provenance topic. The data must be reliable in a world where automatic data analysis systems manage trade, bureaucracy, and daily lives. However, the Web is an open and inclusive dimension in which it is possible to find contradictory and questionable information. Therefore, it is essential to own indications such as the primary source of data, who created or modified it, and when that happened. However, the underlying technologies of the Semantic Web (RDF, OWL, SPARQL) were not originally intended to express such information. In the following paragraphs, the main innovations in this regard will be reconstructed, with particular attention to those concerning the Scientometrics field.

A first step was taken in 2005 when Named Graphs were introduced: graphs associated with a name in the form of a URI. They allow RDF statements describing graphs, with multiple advantages in numerous applications. For example, in Semantic Web publishing, named graphs allow a publisher to sign its graphs so that different information consumers can select specific graphs based on task-specific trust policies. Different tasks require different levels of trust. A naive information consumer may, for example, decide to accept any graph, thus collecting more information as well as more false information. A more cautious consumer may instead require only graphs signed by known publishers, collecting less but more accurate data (Carroll, Bizer, Hayes, & Stickler, 2005).

Subsequently, to revise state of the art and develop a roadmap on provenance for Semantic Web technologies, the Provenance Incubator Group (Provenance Incubator Group Charter, 2010) was established in 2010. One of the first problems was identifying a shared and universal definition of "provenance", a task that proved impossible given its broad and multisectoral nature. Therefore, a working definition was accepted, restricted the context of the Web:

> *Provenance of a resource is a record that describes entities and processes involved in producing and delivering or otherwise influencing that resource. Provenance provides a critical foundation for assessing authenticity, enabling trust, and allowing reproducibility. Provenance assertions are a form of contextual metadata and can themselves become important records with their own provenance*. (Provenance XG Final Report, 08 December 2010)

Starting from this working definition, the group has compiled 33 use cases to formulate scenarios and requirements. Topics covered included eScience, eGovernment, business, manufacturing, cultural heritage, and library science, to name a few. The analysis of these use cases led to the elaboration of three scenarios: a news aggregator, the study of an epidemic and a business contract. The second scenario, the study of the epidemic, is particularly interesting for the case study of this work because it focuses on the reuse of scientific data. Alice is an epidemiologist studying the spread of a new disease called owl flu. Alice needs to integrate structured and unstructured data from different sources, to understand how data has evolved through provenance and version information. In addition, she needs to justify the results obtained by supporting the validity of the sources used, reusing data published by others in a new context and using the provenance to repeat previous analyses with new data. Introducing the problem with a concrete and complex example is helpful to understand how multifaceted and multidimensional it is. Specifically, provenance can be evaluated under three categories: content, management, and usage, each with various dimensions summarized in Table 1.

Category	*Dimension*	*Description*
Content	Object	The artefact that a provenance statement is about.
	Attribution	The sources or entities that contributed to creating the artefact in question.
	Process	The activities (or steps) that were carried out to generate or access the artefact at hand.
	Versioning	Records of changes to an artefact over time and what entities and processes were associated with those changes.
	Justification	Documentation recording why and how a particular decision is made.
	Entailment	Explanations showing how facts were derived from other facts.
Management	Publication	Making provenance available on the Web.
	Access	The ability to find the provenance for a particular artefact.
	Dissemination	Defining how provenance should be distributed and its access be controlled.
	Scale	Dealing with large amounts of provenance.
Use	Understanding	How to enable the end-user consumption of provenance.
	Interoperability	Combining provenance produced by multiple different systems.
	Comparison	Comparing artefacts through their provenance.
	Accountability	Using provenance to assign credit or blame.
	Trust	Using provenance to make trust judgments.
	Imperfections	Dealing with imperfections in provenance records.
	Debugging	Using provenance to detect bugs or failures of processes.

Table 1 Dimensions of provenance

Historically, many vocabularies and ontologies have been introduced to meet some of the above requirements. Among them, the Open Provenance Model stands out because of its interoperability and for being among the first to describe the history of an entity in terms of processes, artefacts and agents, a pattern that will be discussed later on the PROV Data Model (Moreau & Missier, PROV-DM: The PROV Data Model, 2013). Moreover, about domain-relevant models, there is the Provenir Ontology for eScience (Sahoo & Sheth, 2009), PREMIS for archived digital objects, such as files, bitstreams and aggregations (Caplan, 2017) and Semantic Web Applications in Neuromedicine (SWAN) Ontology to model a scientific discourse in the context of biomedical research (Ciccarese, et al., 2008). Finally, the Dublin Core

Metadata Terms allows to express the provenance of a resource and specify what is described (e.g. dct:BibliographicResource), who was involved (e.g. dct:Agent), when the changes occurred (e.g. dct:dateAccepted), and the derivation (e.g. dct:references), sometimes very precisely (DCMI Metadata Terms, 2020).

All the requirements and ontologies mentioned have been merged into a single data model, the PROV Data Model (Moreau, et al., 2011), translated into the PROV Ontology using the OWL 2 Web Ontology Language (Lebo, Sahoo, & McGuinness, 2013). It provides several classes, properties, and restrictions, representing provenance information in different systems and contexts. Its level of genericity is such that it is even possible to create new classes and data model-compatible properties for new applications and domains. Just like the Open Provenance Model, PROV-DM captures the provenance under three complementary perspectives:

- *Agent-centred provenance*, which people, organizations, software, inanimate objects, or other entities are involved in the generation, manipulation, or influence of a resource. For example, concerning a journal article, it is possible to distinguish between the author, the editor, and the publisher. PROV-O maps the responsible agent with prov:Agent, the relationship between an activity and the agent with prov:wasAssociatedWith and an entity's attribution to an agent with prov:wasAttributedTo.
- *Object-centred-provenance*, which is the origin of a document's portion from other documents. Taking the example of the article, a fragment of it can quote an external document. PROV-O maps a resource with prov:Entity, whether physical, digital, or conceptual, while the predicate prov:wasDerivedFrom expresses a derivation relationship.
- *Process-centred provenance*, or the actions and processes necessary to generate a resource. For example, an editor can edit an article to correct spelling errors using the previous version of the document. PROV-O expresses the concept of action with prov:Activity, creating an entity with the predicate prov:wasGeneratedBy and the use of another entity to complete a passage with prov:used.

The diagram in Figure 1 provides a high-level view of the discussed concepts' structure, constituting the so-called "starting point terms". PROV-O is much more extensive and provides sophisticated entities, agents, activities, and relationships in a modular way, namely "expanded terms" and "qualified terms".

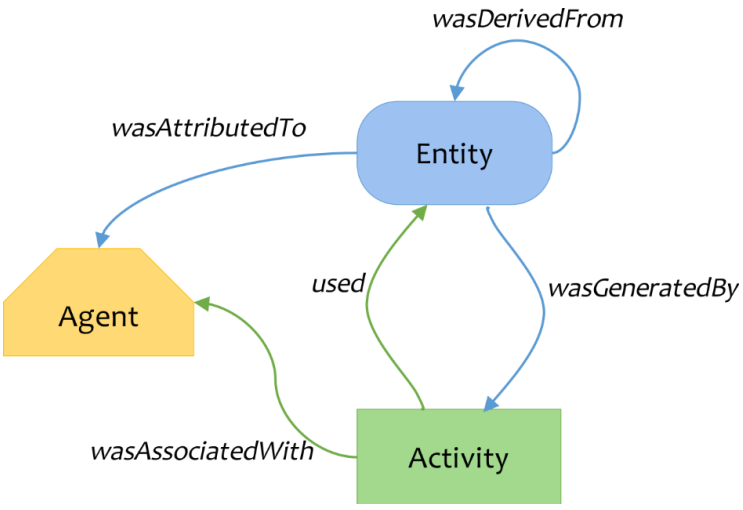

Figure 1 High level overview diagram of PROV records (Gil & Miles, 2013)

The OpenCitations Data Model, used in this research, relies on the flexibility of PROV-O to record the provenance of bibliographic datasets (Daquino, et al., 2020). Each bibliographical entity described by the OCDM is annotated with one or more snapshots of provenance. The snapshots are of type prov:Entity and are connected to the bibliographic entity described through prov:specializationOf, predicate present in the mentioned "expanded terms". Being the specialization of another entity means sharing every aspect of the latter and, in addition, presenting more specific aspects, such as an abstraction, a context or, in this case, a time. In addition, each snapshot records the validity dates (prov:generatedAtTime, prov:invalidatedAtTime), the agents responsible for both creation and modification of the metadata (prov:wasAttributedTo), the primary sources (prov:hadPrimarySource) and a link to the previous snapshot in time (prov:wasDerivedFrom). The model is summarized in Figure 2.

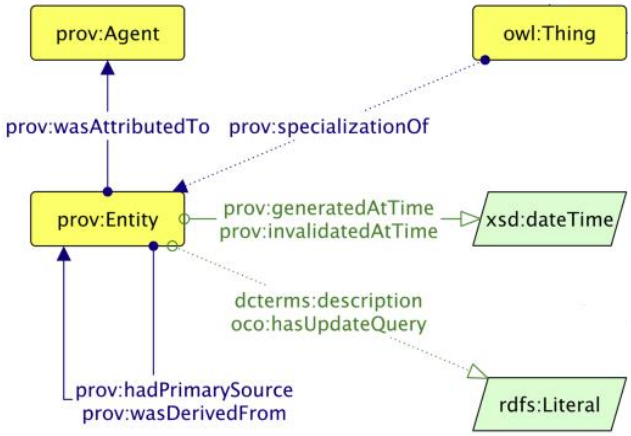

Figure 2 Provenance in the OpenCitations Data Model

In addition, OCDM extends the Provenance Ontology by introducing a new property called *hasUpdateQuery*, a mechanism to record additions and deletions from an RDF graph with a SPARQL INSERT and SPARQL DELETE query string. The *snapshot-oriented* structure, combined with a system to explicitly indicate how a previous snapshot was modified to reach the current state, makes it easier to recover the current statements of an entity and restore an entity to a specific snapshot. The current statements are those available in the present dataset, while recovering a snapshot s_i means applying the reverse operations of all update queries from s_n to s_{i+1} (Peroni, Shotton, & Vitali, 2016).

This expedient, initially adopted for the OpenCitations Corpus, was designed to foster reusability in other contexts and is the added value of the provenance model proposed in the OCDM, which is the basis for the library to perform time agnostic queries presented in this work.

In the next section, the existing literature on tracking changes in RDF data will be deepened, focusing on the sources that inspired the OpenCitations provenance model.

Recording document changes is a problem that goes beyond RDF and has its origins in the notion of the delta. In the article *Introduction to the Universal Delta Model*, Gioele Barabucci defines the delta as:

> *A delta $\Delta_{S,T}$ is a tuple of changes (C) and change relations (R) that describes how to transform the source document (S) into the target document (T)* (Barabucci, Introduction to the Universal Delta Model, 2013).

$$\Delta_{S,T} \equiv (C, R)$$

Since all documents are linear at the bitstream level, it is theoretically possible to apply the same delta algorithms for textual documents to RDF data. However, the lower the compared abstraction level, the less meaningful the delta produced. The meaningfulness indicates how much delta concision is due to complex changes, that is, to a high level of abstraction (Barabucci, Ciancarini, Iorio, & Vitali, 2016).

An RDF document is a *graphical document* in which relationships of any kind link the elements, which means that the information is modelled as a graph at a higher abstraction level. Therefore, there are at least three problems to solve to get a significant delta for RDF documents, which will be explored one at a time:

1. It is necessary to define the level of abstraction to consider to compute the differences between two graphs: the input.
2. It is necessary to determine how to represent the difference: the output.
3. Specific algorithms shall be introduced to obtain the established output given the input.

Taking into account a single RDF document, the highest level of abstraction and granularity is the document itself. However, using the document to compute a change leads to a coarse output, full of irrelevant information that unnecessarily subtracts RAM and storage space. This argument also applies to the immediately lower level of granularity, the named graph. On the other hand, using a triple as granularity level conducts to a correct result only in the absence of blank nodes, while it fails in case two triples share the same blank node. The reason is apparent by looking at the example in Figure 3, where two triples containing blank nodes represent two different people with the same surname, not the same person who changes name.

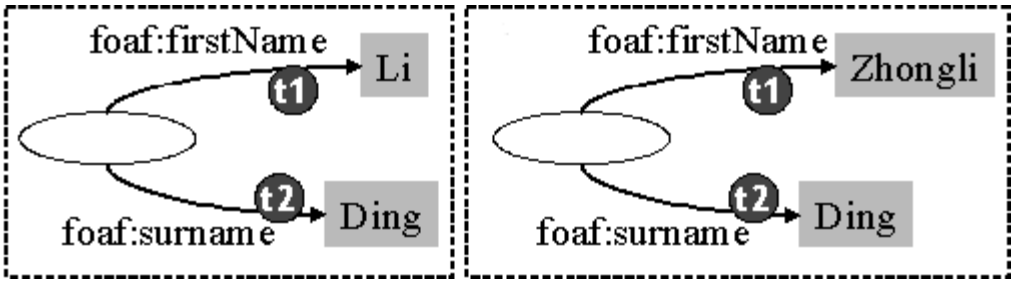

Figure 3 Two triples with blank nodes representing two different people.

Therefore, it is necessary to identify an intermediate level between the triple and the named-graph, a so-called "RDF molecule", that is "the finest and lossless component of an RDF graph" (Ding, Finin, Peng, Silva, & L., 2005). A sub-graph is *lossless* if it can be used to restore the original graph without introducing new triples, while it is said *finest* if it cannot be further decomposed into lossless sub-graphs. This principle was applied in the Opencitations Data Model by choosing the graph associated with an entity as a contextual space to compute the delta and record the provenance.

Once the level of granularity is identified, that is, the input of the diff algorithm, an output format must be chosen. A first formal model to represent changes in RDF repositories was proposed in 2002 and is based on four assumptions (Ognyanov & Kiryakov, 2002):

- *The RDF statement is the smallest directly manageable piece of knowledge*. It is not possible to add, remove or edit a resource without changing at least one statement.
- *An RDF statement cannot be changed — it can only be added and removed*. Only the constituents of a triple determine its identity and, if the constituents change, the triple is converted to a new triple.
- *The two basic types of updates in a repository are the addition and removal of a statement*. Therefore, only these two events need to be tracked by a tracking system. If more complex modifications occur, such as substitution or simultaneous additions, these must be treated and represented in their atomic operations.
- *Each update turns the repository into a new state*. Since a repository state is determined by its statements set, the repository is turned into a new state when an update changes the statements.

The fourth principle derives a corollary: the history of changes in a repository is the sequence of its states. If equivalent states exist with isomorphic graphs, they are treated as different states at different times. Moreover, some repository states can be marked as versions, depending on the user and the application's needs.

Based on the four assumptions and the corollary, the article by Ognyanov and Kiryakov finally proposes an implementation approach. It involves an *update counter* (UC), which increases its integer variable each time the repository is updated, to form an *update identifier* (UID). For each state, the added or deleted statements are indicated, according to the format: **UID:nn {add|remove} <subj, pred, obj>** (Figure 4).

```
History:
UID:1 add <A, r1, B>
UID:2 add <E, r1, D>
UID:3 add <E, r3, B>
UID:4 add <D, r3, A>
UID:5 add <C, r2, D>
UID:6 add <A, r2, E>
UID:7 add <C, r2, E>
UID:8 remove <A, r2, E>
UID:9 add <B, r2, C>
UID:10 remove <E, r3, B>
UID:11 remove <B, r2, C>
UID:12 remove <C, r2, E>
UID:13 remove <C, r2, D>
UID:14 remove <E, r1, D>
UID:15 remove <A, r1, B>
UID:16 remove <D, r3, A>
```

Figure 4 Representation of the history of an RDF repository according to the model of Ognyanov e Kiryakov

The Opencitations Data Model incorporates many of the concepts contained in the Ognyanov and Kiryakov model: it records changes as deletions and additions of statements and, at each update, turns the provenance graph of the related entity to a new state, numbered by an integer, similar to the UID. Each provenance graph is identified by a URL in the form of [snapshot entity URL]:[entity provenance URL]se/[iterative number], e.g.: https://w3id.org/oc/corpurs/br/1423490/prov/se/2.

However, the Ognyanov and Kiryakov's model has a limit. In case of multiple additions and deletions, the repository is turned to a different state for every single statement added or removed. Berners-Lee and Connolly solved this problem, proposing in 2004 a patch file format for RDF deltas, or three new terms (Berners-Lee & Connolly, Delta: an ontology for the distribution of differences between RDF graphs, 2004):

1. diff:replacement, that allows expressing any change. Deletions can be written as {…} diff:replacement {}, and additions as {} diff:replacement {…}.
2. diff:deletion, which is a shortcut to express deletions as {…} diff:deletion {…}.

3. diff:insertion, which is a shortcut to express additions as {…} diff:insertion {…}.

The main advantage of this representation is its economy: given two graphs G1 and G2, its cost in storage is directly proportional to the difference between the two graphs.

The OpenCitations Data Model takes up Berners-Lee and Connolly's proposal, perfecting it using SPARQL 1.1 to express additions and deletions uniquely. It is important to note that the first version of SPARQL was released in 2008, four years after Berners-Lee and Connolly's article, while SPARQL 1.1, which introduced the INSERT and DELETE operations, is 2013.

Established inputs and outputs, the last step is to design an algorithm that can compute the difference between two RDF graphs. In this respect, it is necessary to distinguish between graphs with or without blank nodes. Given two graphs G1 and G2, if each triple's node within them is either a literal or a URI, the two graphs are *grounded* and computing their difference is immediate and straightforward.

> *If G1 and G2 are ground RDF graphs, then the ground graph delta of G1 and G2 is a pair (insertions, deletions) where insertions is the set difference G2 − G1 and deletions is G1 − G2* (Berners-Lee & Connolly, Delta: an ontology for the distribution of differences between RDF graphs, 2004).

If blank nodes are present, the discourse is more complex: in the absence of an ontology, it is impossible to compute the diff without the risk of generating inconsistencies. Then, it is necessary to derive a so-called *fully labelled* graph, in which every node is *functionally ground* to an ontology. For a node *n* to be functionally grounded, at least one of the following conditions must occur:

- There is a triple (n, p, o) in G, p is an inverse functional property according to an ontology W, and either *o* is grounded or functionally grounded. In other words, it is possible to disambiguate *n* since, given *o* and W, there is only one possible value of *n*.
- There is a triple (s, p, n) in G, p is a functional property according to an ontology W, and *s* is grounded or functionally grounded. In other words, it is possible to disambiguate *n* since, since given *s* and W, there is only one possible value of *n*.
- There is a node *n'* in G such that *n* is equivalent to *n'* compared to W, and *n* is grounded or functionally grounded.

In the presence of fully labelled graphs, context-free "strong" diffs can be obtained, similarly as seen for ground RDF deltas:

> *Given a background ontology W, a strong delta between fully labelled graphs G1 and G2 is a pair (insertions, deletions) where insertions is the set difference F2 − F1, deletions is F1 − F2, and F1 and F2 are functional analogs of G1 and G2 respectively* (Berners-Lee & Connolly, Delta: an ontology for the distribution of differences between RDF graphs, 2004).

BIBLIOGRAPHY

Barabucci, G. (2013). Introduction to the Universal Delta Model. *Proceedings of the 2013 ACM Symposium on Document Engineering* (p. 47–56). Florence, Italy: Association for Computing Machinery. doi:10.1145/2494266.2494284

Barabucci, G., Ciancarini, P., Iorio, A. D., & Vitali, F. (2016). Measuring the quality of diff algorithms: a formalization. *Computer Standards & Interfaces, 46*, 52-65. doi:10.1016/j.csi.2015.12.005

Berners-Lee, T. (1999). *Weaving the Web: the original design and ultimate destiny of the World Wide Web.* San Francisco: Harper San Francisco.

Berners-Lee, T., & Connolly, D. (2004). Delta: an ontology for the distribution of differences between RDF graphs. Retrieved from https://www.w3.org/DesignIssues/Incs04/Diff.pdf

Caplan, P. (2017). Understanding PREMIS: an overview of the PREMIS Data Dictionary for Preservation Metadata. Library of Congress. Retrieved from Library of Congress: https://www.loc.gov/standards/premis/understanding-premis-rev2017.pdf

Carroll, J. J., Bizer, C., Hayes, P., & Stickler, P. (2005). Named graphs, provenance and trust. *Proceedings of the 14th international conference on World Wide Web* (p. 613–622). New York: Association for Computing Machinery. doi:10.1145/1060745.1060835

Ciccarese, P., Wu, E., Kinoshita, J., Wong, G. T., Ocana, M., Ruttenberg, A., & Clark, T. (2008). The SWAN Scientific Discourse Ontology. *Journal of biomedical informatics, 41*(5), 739–751. doi:10.1016/j.jbi.2008.04.010

Daquino, M., Peroni, S., Shotton, D., Colavizza, G., Ghavimi, B., Lauscher, A., . . . Zumstein, P. (2020). The OpenCitations Data Model. In J. Z. Pan, V. Tamma, C. d'Amato, K. Janowicz, B. Fu, A. Polleres, . . . L. Kagal (A cura di), *International Semantic Web Conference. 12507*, p. 447-463. Springer, Cham. doi:10.1007/978-3-030-62466-8_28

DCMI Metadata Terms. (2020, 01 20). Retrieved 07 16, 2021, from Dublin Core Metadata Initiative: http://dublincore.org/specifications/dublin-core/dcmi-terms/2020-01-20/

Ding, L., Finin, T., Peng, Y., Silva, P. P., & L., D. (2005). *Tracking RDF Graph Provenance.* Technical report. Retrieved from http://ebiquity.umbc.edu/get/a/publication/178.pdf

Gil, Y., & Miles, S. (Eds.). (2013, 04 30). *PROV Model Primer.* Retrieved 07 16, 2021, from W3C: http://www.w3.org/TR/2013/NOTE-prov-primer-20130430/

Lebo, T., Sahoo, S., & McGuinness, D. (Eds.). (2013, 04 30). *PROV-O: The PROV Ontology.* Retrieved 07 16, 2021, from W3C: http://www.w3.org/TR/2013/REC-prov-o-20130430/

Moreau, L., & Missier, P. (Eds.). (2013, 04 30). *PROV-DM: The PROV Data Model.* Retrieved 07 16, 2021, from W3C: http://www.w3.org/TR/2013/REC-prov-dm-20130430/

Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., . . . Bussche, J. V. (2011). The Open Provenance Model core specification (v1.1). *Future Generation Computer Systems,, 27*(6), 743-756. doi:10.1016/j.future.2010.07.005

Ognyanov, D., & Kiryakov, A. (2002). Tracking Changes in RDF(S) Repositories. In G.-P. A., & B. V.R. (Ed.), *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web* (pp. 373-378). Berlin, Heidelberg: Springer. doi:10.1007/3-540-45810-7_33

Peroni, S., Shotton, D., & Vitali, F. (2016). A Document-inspired Way for Tracking Changes of RDF Data. In L. Hollink, S. Darányi, A. M. Peñuela, & E. Kontopoulos (Ed.), *Detection, Representation and Management of Concept Drift in Linked Open Data. 1799*, pp. 26-33. Bologna: CEUR Workshop Proceedings. Retrieved from http://ceur-ws.org/Vol-1799/Drift-a-LOD2016_paper_4.pdf

Provenance Incubator Group Charter. (2010). Retrieved July 15, 2021, from https://www.w3.org/2005/Incubator/prov/charter

(08 December 2010). *Provenance XG Final Report.* W3C. Retrieved from http://www.w3.org/2005/Incubator/prov/XGR-prov-20101214/

Sahoo, S. S., & Sheth, A. P. (2009). Provenir Ontology: Towards a Framework for eScience Provenance Management. Retrieved from https://corescholar.libraries.wright.edu/knoesis/80