DCPAPERS
Proc. Int'l Conf. on Dublin Core and Metadata Applications 2013

Provenance and Annotations for Linked Data

Kai Eckert
University of Mannheim, Germany
kai@informatik.uni-mannheim.de

Abstract

Provenance tracking for Linked Data requires the identification of Linked Data resources. Annotating Linked Data on the level of single statements requires the identification of these statements. The concept of a Provenance Context is introduced as the basis for a consistent data model for Linked Data that incorporates current best-practices and creates identity for every published Linked Dataset. A comparison of this model with the Dublin Core Abstract Model is provided to gain further understanding, how Linked Data affects the traditional view on metadata and to what extent our approach could help to mediate. Finally, a linking mechanism based on RDF reification is developed to annotate single statements within a Provenance Context.

Keywords: Provenance; Annotations; RDF; Linked Data; DCAM; DM2E;

1. Introduction

This paper addresses two challenges faced by many Linked Data applications: How to provide, access, and use provenance information about the data; and how to enable data annotations, i.e., further statements about the data, subsets of the data, or even single statements. Both challenges are related as both require the existence of identifiers for the data. We use the Linked Data infrastructure that is currently developed in the DM2E project as an example with typical use-cases and resulting requirements.

1.1. Foundations

Linked Data, the publication of data on the Web, enables easy access to data and supports the reuse of data. The Hypertext Transfer Protocol (HTTP) is used to access a Uniform Resource Identifier (URI) and to retrieve data about the resource. Correspondent to links in HTML documents, the data can provide links to other resources, thus forming the Web of Data.

The **Resource Description Framework (RDF)** is used to provide the data and fits perfectly to the idea of a distributed data space. In RDF, data is split into single statements about resources. A statement consists of a subject, a predicate, and an object; it relates a resource to a literal value or a resource to another resource. Thus, the statements – or triples, both terms are used equivalently in this paper – form a directed, labeled graph of resources and literals. As URIs are used to identify the resources and the predicates, it is guaranteed that both are globally unique and that statements from various sources can be merged without syntactical conflicts on the level of RDF – although it can happen that two statements contradict each other on the level of semantics. For this reason, all the RDF data in the world (or at least on the Web) is sometimes referred to as the Giant Global Graph, in contrast to the World Wide Web.

RDF does not address the problem how to organize the RDF statements within an application. In particular, it provides no means to distinguish them, for instance based on their source. Therefore, **Named Graphs** (Carroll et al., 2005) have been proposed, where an RDF graph is identified by a URI. Named Graphs are commonly used to organize RDF data within a triple store (an RDF database) and are supported by the RDF query language SPARQL. They will also be part of the upcoming RDF version 1.1, according to the latest working draft of the specification (Cyganiac and Wood, 2013).

https://doi.org/10.23106/dcmi.952136081

Linked Data imposes a similar view on RDF data as Named Graphs: when a URI of a resource is dereferenced, the RDF data describing the resource is not delivered directly; instead an HTTP redirect is used to refer to the URL of the document that contains the data. This URL can be used as URI to identify the data.

1.2. Motivating example: DM2E

The project "Digitised Manuscripts to Europeana" (DM2E) is an EU funded project with two primary goals:

1. The transformation of various metadata and content formats describing and representing digital cultural heritage objects (CHOs) from as many providers as possible into the Europeana Data Model (EDM) to get it into Europeana.
2. The stable provision of the data as Linked Open Data and the creation of tools and services to re-use the data in the Digital Humanities, i.e., to support the so-called Scholarly Primitives (Unsworth, 2000). The basis is the possibility to annotate the data, to link the data, and to share the results as new data.

All metadata in DM2E stems from various cultural heritage institutions across Europe and is maintained and provided in various formats, among others MARC 21, MAB2, TEI, and METS-MODS. These formats generally provide enough degrees of freedom to reflect specific requirements of the providers, hence it has to be expected that for each format and each provider, a different transformation has to be created and maintained. Changes in the original metadata and adaptions of the transformation process lead to new **versions** of the published data, for which **the provenance has to be provided**.

Data annotation requires that **every single statement can be the subject of annotations and links** to other resources. The annotations and links are created as part of the scholarly work and become important research data that forms the basis for publications and fosters the communication among the scholars. Therefore, it is crucial that **stable versions** of the metadata are provided as a trustable foundation of this work. Finally, a **complete provenance chain** for every single metadata statement and every annotation has to be provided to trace them back to the original data file located at the provider and to the creator of the annotation, respectively.

Figure 1 shows the general architecture of a Linked Data Service that applies also to DM2E. Metadata from various **input files** is transformed to RDF and ingested into an RDF database, a **triple store**. Within the triple store, the RDF data is organized via Named Graphs. In DM2E, for each ingestion a **Named Graph** is created. The data is accessed via a Linked Data API where all URIs used in the data are dereferenced and subsets of the RDF data describing the requested resource are returned as single **web documents**.

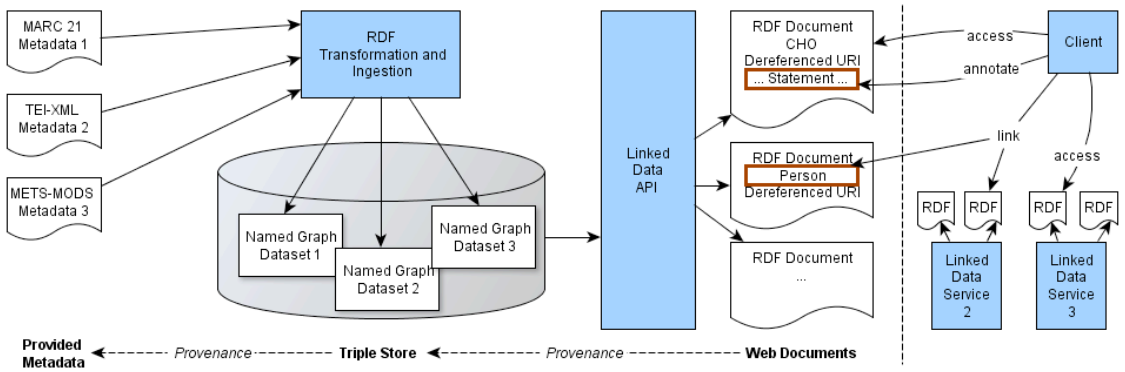

FIG. 1. DM2E Architecture

https://doi.org/10.23106/dcmi.952136081

For each web document, provenance information must be provided so that it can be related to a specific ingestion in the triple store and subsequently to the originally provided metadata.

1.3. Prior and Related Work

A comprehensive introduction to Linked Data provenance from the metadata perspective is provided in (Eckert, 2012). This present paper is in part a continuation and provides the framework for the DM2E data model as an extension of the Europeana Data Model that fits better with the Linked Data principles. The contributions of this paper are the introduction of the Provenance Context, the recognition of the Linked Data principles as a limitation for provenance applications and the proposal of fragment identifiers for statements.

In (Eckert et al., 2011), we investigated the Dublin Core Abstract Model (DCAM) regarding its suitability to model metadata provenance and proposed an extension in line with Named Graphs. In this paper, we use DCAM to relate Linked Data and our proposed implementation to the terminology of the metadata community.

In (Eckert et al., 2009, 2010) we presented several use-cases for the provenance of single metadata statements and demonstrated implementations with RDF Reification and Named Graphs. In this paper, we show that this fine-grained representation has to be sacrificed in the Linked Data context and examine the consequences for our use-cases.

Many papers – let alone blog posts and presentations – have been published about Linked Data best practices, including proposals how to provide provenance information for Linked Data. In a recent contribution, Zhao and Hartig (2012) compare provenance vocabularies, including the upcoming PROV ontology, for the representation of the provenance of Linked Data. Of particular interest for our work is the characterization of Linked Data resources as state-ful and state-less resources, which we will use as a basis for our considerations on versioning.

Another foundation for our work is the distinction of datasets and published RDF documents, which is addressed by the VoID vocabulary (Keith et al., 2009). Omitola et al. (2011) propose an extension of VoID that uses several existing vocabularies to describe the provenance of VoID datasets. We do not restrict our approach to the use of VoID; instead we generalize the best practices presented in these publications and focus on a consistent data model that extends to the metalevel where the provenance information is linked to.

2. Linked Data Provenance

It is not difficult at all to provide provenance information for Linked Data. There are various options and best practices, as described for example in (Eckert, 2012). The straight-forward approach following the Linked Data principles is to use the URI of the RDF document the data is retrieved from as subject for statements about its provenance. This works in principle, but has a major flaw: it conveys the notion that the RDF documents are independent data resources with own provenance each. In a typical Linked Data application (as in Figure 1), the RDF documents are rather views on subsets of larger datasets and the provenance of the statements in an RDF document is the same as the provenance of the dataset itself.

The relation between RDF documents and larger RDF datasets can be modeled using the Vocabulary of Interlinked Datasets (VoID).[1] It is reasonable to provide the provenance information for the void:Dataset instead of each RDF document, as it is proposed by Omitola et al. (2011), as well as in the documentation of VoID. There are, however, various reasons to group RDF statements into datasets, with provenance tracking being only one of it. VoID consequently supports the hierarchical relation of datasets as subsets, which leaves the problem to the

[1] http://vocab.deri.ie/void

https://doi.org/10.23106/dcmi.952136081

consumer to determine which dataset is the one that provides the boundary for the provenance of the contained statements.

Figure 2 illustrates a possible organization of RDF data into documents and nested datasets. The depiction of the smallest sets of RDF statements as RDF documents is a concession to VoID and the Linked Data setting, where data is accessed via HTTP; it is not possible to distinguish multiple sets within one RDF document.[2] From a modeling perspective, an RDF document is just a special dataset that forms a subset of the dataset it is associated with.

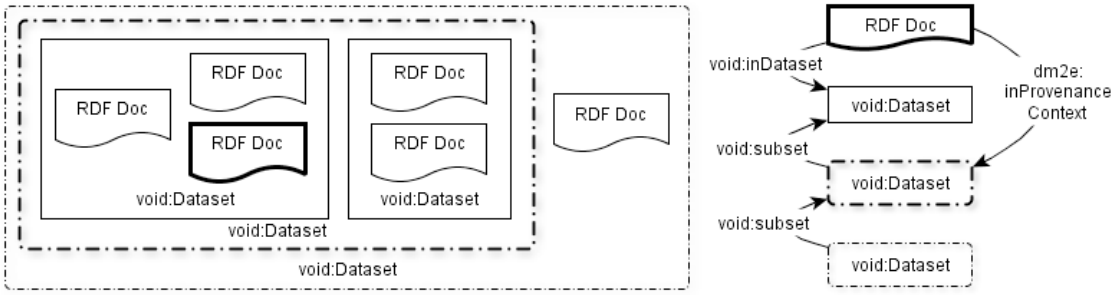

FIG. 2. Nested graph boundaries and the Provenance Context

Besides the modeling perspective, there is also the implementation perspective: all datasets and documents in the scenario of Figure 2 are Named Graphs, as they all represent a well-defined RDF graph and they provide a name by means of their own URI. The implementer decides how statements are organized internally, which – using a quad store – affects the amount of space required to store the data as well as the possibilities to query the data; and the implementer decides which portions of the data are exposed as RDF documents via HTTP.

Likewise, data consumers have to decide how to replicate remote data in local systems. A typical pattern is the creation of Named Graphs in a local triple store for each RDF document that is consumed, with the URL of the document as URI of the Named Graph. Another possibility is the creation of a Named Graph with the URI of the void:Dataset, if one is given, which would lead to an exact replication of the original data in the DM2E system (Figure 1), at the price of discarding the information via which API call (i.e., which RDF document) the data was retrieved.

2.1. The Provenance Context

We propose to provide the data consumer with some guidance which graph boundary has to be preserved in order to track the provenance of the consumed statements. This is particularly important for the second challenge addressed in this paper: the annotation of statements. Therefore, a statement needs an identity and it is not only purely philosophical to relate identity to provenance. Therefore, we introduce the Provenance Context.

Definition 1 (Provenance Context): *A Provenance Context is a set of RDF triples that share the same provenance, identified by a URI.*

Formally, a Provenance Context is a Named Graph. Any identified resource that represents an RDF graph can be a Provenance Context, particularly any void:Dataset and any foaf:Document that contains serialized RDF data. Note that we deliberately ignore the difference between a foaf:Document as a concrete RDF serialization of an RDF graph and a Named Graph as a purely abstract resource without any inherent semantics or specifications on what is retrieved if the URI is dereferenced. We focus on the role of these resources as boundaries of RDF graphs.

[2] Of course, different Named Graphs could be serialized in one document – e.g., using the TriG syntax – but this is not a common practice and would not be interpretable by today's linked data applications.

https://doi.org/10.23106/dcmi.952136081

For discussions about Linked Data provenance, it is very helpful to talk about the Provenance Context of statements, instead of vague or arbitrary embodiments like Named Graphs or documents. In line with Linked Data practices and the common use of the VoID vocabulary, the Provenance Context of a statement is determined as follows:

1 Per default, the Provenance Context of a triple is the document identified by the URL it is retrieved from or the Named Graph that contains the statement.

2 If the document or the Named Graph is related to a void:Dataset via void:inDataset, the Provenance Context is the void:Dataset.

3 If this does not determine the Provenance Context, it can be stated explicitly by relating the document, the Named Graph, or the dataset to the Provenance Context using the property **dm2e:inProvenanceContext** (see Figure 2).

The following condition holds: there must always be one and only one Provenance Context for each RDF statement, as the Provenance Context determines the identity of a statement, in order to make further statements about it.

From this condition and Definition 1 follows that every RDF graph (RDF document, Named Graph, void:Dataset) either **is** a Provenance Context or is contained **completely** within one Provenance Context.

Furthermore follows that the Provenance Context determines the maximum permissible set of RDF statements that are published together. This is often neglected in practice when additional statements – ironically often provenance statements – are published together with the actual data. It depends on the application if this is a problem: in DM2E, the Provenance Context is the dataset that is created for each ingestion. As the provenance information is created as part of the ingestion process as well, we put them in the same Provenance Context, which allows us to provide provenance information together with the actual data in the published RDF documents.

2.2. State-ful Resources and Versioning

Zhao and Hartig (2012) use the terms state-less and state-ful resources to distinguish resources whose characterization in terms of RDF statements can change over time (state-less) and resources whose RDF representation remains stable (state-ful). An example for a state-less resource is the London weather forecast at http://example.org/forecast/london that changes every day, a state-ful resource could be provided via http://example.org/forecast/london_0702. If the provided RDF data – as in the case of DM2E – is subject of further annotations, the Provenance Context should be state-ful, i.e., its content should be immutable. Otherwise the annotations could be misleading if they are interpreted in view of the changed content. To indicate the immutability of a resource, Zhao and Hartig (2012) propose to classify resources explicitly as prv:Immutable.

In DM2E, the immutability of the Provenance Contexts is ensured by means of versioning. Whenever updated metadata is ingested or an updated transformation process is used, a new dataset is created determining a new Provenance Context for the ingested statements. It is important to relate the versions, not only to track changes, but also to consume annotations made on earlier versions. They might not be applicable any more, but deciding that is left to the consumer. An example would be an indication to the user that there is an annotation for a statement that does not exist anymore in this dataset. We use the following simple vocabulary for the versioning:

- **dm2e:previousVersion:** links to the previous version of this dataset.
- **dm2e:firstVersion:** links to the oldest available version of this dataset.
- **dm2e:version:** serial number of this version, starting with 1.
- **dm2e:versionName:** provides a human-readable name for this version.

https://doi.org/10.23106/dcmi.952136081

- **dm2e:nextVersion:** links to the next version of this dataset.
- **dm2e:latestVersion:** links to the latest available version of this dataset.
- **dm2e:availableVersions:** number of available versions of this dataset.

The latter three properties are special as they are obviously not immutable and have to be adjusted when new versions are created. If and how they are used depends on the application and the requirements regarding the stability of the data. Purists only deliver them when the Provenance Context is dereferenced, using a URL that belongs to a different Provenance Context.

3. Relation to the Dublin Core Abstract Model

In (Eckert et al., 2011), we proposed an extension of the Dublin Core Abstract Model (DCAM, Powell et al., 2007) to support metadata provenance. The crucial point was the declaration of a *Description Set* as an identifiable resource of a to be defined class dcam:DescriptionSet. We demonstrated that this fits to the notion of a Named Graph in an RDF implementation. We further defined an *Annotation Set* as a specialization of a Description Set containing statements *about* a Description Set, and required that all statements about Description Sets belong to an Annotation Set. This facilitates good data modeling practice, but might hamper the adoption in legacy metadata applications.

The role and the purpose of DCAM are subject of long lasting discussions in the Dublin Core community. It is the basis for Description Set Profiles in the Singapore Framework (Nilsson et al., 2008) and is generally used as a common language to talk about the inherent characteristics of metadata from a metadata practitioner's perspective, for example to teach the semantics of metadata in information science lectures. In this section, we investigate how DCAM fits to Linked Data and our proposed modeling, how Linked Data changes the perception of DCAM, and where both even contradict.

The definitions of a **Description** and a **Statement** in DCAM fit to RDF, although the perspective is slightly different. A statement in DCAM is a property-value pair, not a triple; a Description is a set of statements describing one and only one resource. Together, they resemble the RDF statements, where the notion of a Description would be implicit as the set of triples that share the same subject.

In DCAM, a **Description Set** is defined as a set of one or more Descriptions, so any RDF graph is a Description Set. We used this liberal interpretation of a Description Set in our proposed DCAM extension. This was arguably an even more important paradigm shift that we implicitly introduced, than the mere definition of a class to define Description Sets as described resources. This interpretation does not contradict DCAM, but DCAM at least suggests a narrower interpretation of a Description Set (Powell et al., 2007):

> However, real-world metadata applications tend to be based on loosely grouped sets of *descriptions* (where the *described resources* are typically related in some way), known here as *description sets*. For example, a *description set* might comprise *descriptions* of both a painting and the artist. Furthermore, it is often the case that a *description set* will also contain a *description* about the *description set* itself (sometimes referred to as 'admin metadata' or 'meta-metadata'). *Description sets* are instantiated, for the purposes of exchange between software applications, in the form of metadata *records*.

So the Description Set in DCAM forms a logical boundary – physically embodied as a Record – that creates an identity for the metadata. This interpretation does not hold in a Linked Data setting, not for nothing the record has been pronounced dead at various occasions.

Instead, any RDF publication is a Record containing a Description Set. These Description Sets are parts of larger Description Sets; the Records are rather created on the fly to make subsets of these larger Description Sets available. In other words: the record is not dead, but it is no longer

https://doi.org/10.23106/dcmi.952136081

the central entity for metadata. The central entity is the Provenance Context, a special Description Set that creates the identity for the contained metadata.

Our proposed introduction of an Annotation Set (Eckert et al., 2011) implied the separation of metadata and its provenance data, particularly to ensure that the provenance of the provenance can be expressed. This fits generally to RDF and Named Graphs, but is subject to limitations when it comes to Linked Data. A separation in order to express the provenance of the provenance means that the Provenance Context of the provenance is different from the Provenance Context of the data. This would forbid the common Linked Data practice to provide provenance information together with the actual data. If, however, provenance statements are provided within the same Provenance Context, it must be ensured that their provenance is indeed the same as the data provenance. Otherwise, it is strictly recommended to provide the provenance information in a different Provenance Context, i.e., a dedicated Annotation Set.

4. Linked Data Annotations

Annotating Linked Data means to create statements **about** Linked Data. These can either be direct RDF statements or indirect annotations using for example the Open Annotation Data Model (Sanderson et al., 2013). If single statements are the subject of an annotation, they need an identifier, which is not generally available in RDF.[3] Therefore, we propose to identify a statement within a Provenance Context by means of a fragment URI, similar to XPointer for the identification of elements in an XML file (Grosso et al., 2003) or the recently published W3C recommendation for media fragment URIs (Troncy et al., 2012).

4.1. Fragment Identifiers for Statements

A URI consists of the following parts:

<scheme name>:<hierarchical part>[?<query>][#<fragment>]

Usually, the fragment identifier (denoted by <fragment>) is used to identify a fragment within the content identified by the first part of the URI, before the # sign. The recommendation for media fragment URIs allows to use the <query> part as well for the fragment identifier, which makes the fragment accessible independent of the containing resource. We use the fragment identifier in the query part as it allows us to dereference the URI as usual and to provide explaining statements to clients that do not directly support our fragment identifiers.

The fragment identifier for a statement is created as a key value pair in the following form:

```
spo=subject,predicate,object
```

Subject, predicate, and object are percent-encoded (RFC 3986) representations following the N-Triples syntax.[4] Consider the following example:

```
http://example.org/provcontext1?spo=%3Chttp%3A%2F%2Fexample.org%2Fdata%
2Fdoc1%3E,%3Chttp%3A%2F%2Fpurl.org%2Fdc%2Fterms%2Fcreator%3E,%3Chttp%3A
%2F%2Fexample.org%2Fpersons%2Fkai%3E
```

This URI refers to the following statement in the Provenance Context `<http://example.org/provcontext1>`:

```
<http://example.org/data/doc1> <http://purl.org/dc/terms/creator>
<http://example.org/persons/kai>.
```

[3] RDF-XML provides optional statement identifiers as a shortcut for reification, but they are usually not created for all statements in a document, not portable to other serializations and not part of the data model.

[4] http://www.w3.org/TR/rdf-testcases/#subject

https://doi.org/10.23106/dcmi.952136081

4.2. Contextual Reification

These statement URIs can be created on the fly wherever statements about statements are made. If the meaning of such a URI is known to the application, it can directly be interpreted as a statement identifier.

Putting semantics into a URI, however, is usually considered an anti-pattern, and rightly so. Fortunately, we can make the semantics explicit any time and provide the necessary information in full compliance with the linked data principles. The generated URI just needs to be dereferenceable (hence the use of the query part, not the fragment part) and the web server could provide the following RDF statements for the URI above that can directly be derived from the URI itself:

```
<http://example.org/provcontext1?spo=%3Chttp%3A%2F%2Fexample.org%2Fdata
%2Fdoc1%3E,%3Chttp%3A%2F%2Fpurl.org%2Fdc%2Fterms%2Fcreator%3E,%3Chttp%3
A%2F%2Fexample.org%2Fpersons%2Fkai%3E> a rdf:Statement ;
   rdf:subject <http://example.org/data/doc1> ;
   rdf:predicate <http://purl.org/dc/terms/creator> ;
   rdf:object <http://example.org/persons/kai> ;
   dm2e:context <http://example.org/provcontext1> .
```

This actually would be an RDF reification, i.e., the definition of a resource that represents a statement in order to make further statements about it. Reification is not very popular among RDF users, even its deprecation in the next version of RDF has been discussed (Hawke, 2011).[5] Among the reasons is the cumbersome representation of a reified statement that leads to a triple explosion, as four additional statements are needed to reify one statement. The actual statement still has to be made.

We propose to use reification as a means to characterize statement URIs referring to statements **within a provenance context**. This is in line with the current semantics[6] of RDF reification (Hayes, 2004): "… the reified triple that the reification describes […] [is required] to be a particular token or instance of a triple in a (real or notional) RDF document, rather than an 'abstract' triple considered as a grammatical form." There is only one missing link, also stated in the RDF semantics: "Although RDF applications may use reification to refer to triple tokens in RDF documents, the connection between the document and its reification must be maintained by some means external to the RDF graph syntax." This has been written before the advent of Named Graphs, the Linked Data principles, and the VoID vocabulary. Today, we can easily close this gap and introduce a new property to link a reification to the Provenance Context in which the statement is made: **dm2e:context**.

Such a **Contextual Reification** is stable: the context information is directly associated with the reification and is preserved across graph and system borders.

The contextual reification has more precise semantics than the reification. As the context indicates an RDF graph and an RDF graph is defined as a set of triples, the following holds: Two

---

[5] At the time of this writing (March 2013), the role of the reification in the next version of RDF is not yet clear. Probably, it will not be part of core RDF any more, but still available as part of the RDF vocabulary, i.e., it still can and should be used to talk about statements, but formal semantics or constraints do not exist anymore. Instead, the reification vocabulary would have the same status as any other RDF vocabulary; there might be more or less formally defined semantics and intended meanings, but the exact interpretation would be left to the applications.

[6] The semantics of RDF reification have been discussed at length. Based on the arguments in the text, I (sic!) think that this use of reification is correct. To be safe from any model-theoretic considerations and debates, it would probably be wise to use a dedicated, new (another one) vocabulary for the explanation of the statement resources or some indirection that does not strictly assign the context to the reification.

https://doi.org/10.23106/dcmi.952136081

contextual reifications are owl:sameAs iff their respective context, subject and predicate URI reference, as well as the object URI reference or literal representation are equal.

5. Conclusion

In this paper, we revisited current Linked Data practices for the provision of provenance information and related them to our own theoretical work about metadata provenance and the Dublin Core Abstract Model. Motivated by the requirements derived from the use-cases of the DM2E project, we provided a consistent and complete picture how Linked Data can be published with proper provenance information. We were not concerned with the actual representation of the provenance information – Dublin Core or W3C PROV are both suitable; instead we concentrated on the question how the identity of Linked Data is determined, i.e., which subject should be used to assign the provenance to.

The key finding is that there are typically various candidates, ranging from the published RDF documents over Named Graphs to potentially nested void:Datasets. One of these candidates creates the identity of the published data and we call this the Provenance Context. We provided sensible defaults how the Provenance Context can be determined and a means to indicate it explicitly, if needed or desired.

We further proposed a fragment identifier for statements to unambiguously identify a statement within a Provenance Context and showed that it can be used to annotate single statements. Following the Linked Data principles, we recommended to provide a description for such statement URIs when dereferenced and suggested to use the reification vocabulary, extended by a context property that makes the Provenance Context of the reified statement explicit.

We explained how this model will be implemented in DM2E and provided a simple versioning vocabulary that will be used to deal with updates to the data, and to allow the client to retrieve annotations made on older versions or to provide the original state of a dataset when an annotation was made.

Regarding the Provenance Context and the Fragment-URIs, this paper is mainly a position paper that represents the current point of view of the author. Future work includes the actual implementation of this approach in DM2E, a possibly community-driven specification of the RDF Fragment URIs and further investigation of the idea of a Provenance Context based on feedback from the community. Interesting questions regarding the fragment identifiers include very practical issues arising from very long URIs, for example, when long literal values are included; another field for further investigations would be reasoning based on such statement annotations.

Acknowledgements

The author is funded by the European Commission within the DM2E project (http://dm2e.eu). Valuable input and feedback was provided by Christian Bizer, University of Mannheim.

References

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 WWW 05, 14, 613. ACM Press. Retrieved from http://portal.acm.org/citation.cfm?doid=1060745.1060835

Cyganiac, Richard, Wood, David (eds.) (2013). RDF 1.1 Concepts and Abstract Syntax. W3C Working Draft 15 January 2013. Retrieved from http://www.w3.org/TR/2013/WD-rdf11-concepts-20130115/.

Eckert, K. (2012). Metadata Provenance in Europeana and the Semantic Web. Berliner Handreichungen zur Bibliotheks- und Informationswissenschaft 332. ISSN: 1438-7662. Berlin: Humboldt-Universität zu Berlin, Philosophische Fakultät I, Institut für Bibliotheks- und Informationswissenschaft.

https://doi.org/10.23106/dcmi.952136081

Eckert, K., Garijo, D., Panzer, M. (2011). Extending DCAM for Metadata Provenance. Proceedings of the 2011 Dublin Core Conference, The Hague. Retrieved from http://dcpapers.dublincore.org/pubs/article/view/3621.

Eckert, K., Pfeffer, M., & Stuckenschmidt, H. (2009). A Unified Approach for Representing Metametadata. Proceedings of the 2009 Dublin Core Conference, Seoul. Retrieved from http://dcpapers.dublincore.org/ojs/pubs/article/view/973/948

Eckert, K., Pfeffer, M., & Völker, J. (2010). Towards Interoperable Metadata Provenance. Proceedings of the Second International Workshop on the role of Semantic Web in Provenance Management (SWPM 2010) Workshop at the 9th International Semantic Web Conference (ISWC 2010). CEUR-WS.org, online CEUR-WS.org/Vol-670.

Grosso, Paul, Eve Maler, Jonathan Marsh, & Norman Walsh. (2003). XPointer Framework. W3C Recommendation 25 March 2003. Retrieved from http://www.w3.org/TR/2003/REC-xptr-framework-20030325/

Hawke, S. (2011). RDF-ISSUE-25 (Deprecate Reification): Should we deprecate (RDF 2004) reification? [Cleanup tasks]. Mailinglist public-rdf-wg@w3.org. Retrieved from http://lists.w3.org/Archives/Public/public-rdf-wg/2011Apr/0164.html

Hayes, P. (2004). RDF Semantics. W3C Recommendation 10 February 2004. Retrieved from http://www.w3.org/TR/2004/REC-rdf-mt-20040210/

Keith, A, Cyganiak, R, Hausenblas, M, Zhao, Jun. (2009). Describing Linked Datasets. On the Design and Usage of voiD, the "Vocabulary Of Interlinked Datasets". In: Linked Data on the Web (LDOW 2009), Proceedings of the WWW2009 Workshop on Linked Data on the Web, Madrid, Spain, April 20, 2009. CEUR-WS.org, online CEUR-WS.org/Vol-538.

Nilsson, M., Baker, T., Johnston, P. (2008). The Singapore Framework for Dublin Core Application Profiles. DCMI Recommended Resource. Retrieved from http://dublincore.org/documents/2008/01/14/singapore-framework/

Omitola, Tope, Zuo, Landong, Gutteridge Christopher, Millard, Ian C. , Glaser, Hugh, Gibbins, Nicholas, and Shadbolt, Nigel. (2011) Tracing the Provenance of Linked Data using void. WIMS '11 International Conference on Web Intelligence, Mining. Sogndal, Norway — May 25 - 27, 2011. New York: ACM.

Powell, A., Nilsson, M., Naeve, A., Johnston, P., & Baker, T. (2007). DCMI Abstract Model. Dublin Core Metadata Initiative. Available from http://dublincore.org/documents/abstract-model

Sanderson, R., Ciccarese, P., Van de Sompel, H. (2013): Open Annotation Data Model. Community Draft, 08 February 2013. Retrieved from http://www.openannotation.org/spec/core/20130208/index.html.

Troncy, Raphaël, Erik Mannens, Silvia Pfeiffer, & Davy Van Deursen (2012). Media Fragments URI 1.0 (basic). W3C Recommendation 25 September 2012. Retrieved from http://www.w3.org/TR/2012/REC-media-frags-20120925/

Unsworth, John (2000). Scholarly Primitives: what methods do humanities researchers have in common, and how might our tools reflect this? Part of a symposium on "Humanities Computing: formal methods, experimental practice" sponsored by King's College, London, May 13, 2000. Retrieved from http://people.lis.illinois.edu/~unsworth/Kings.5-00/primitives.html

Zhao, J., Bizer, C., Gil, Y., Missier, P., & Sahoo, S. (2010). Provenance Requirements for the Next Version of RDF. Proceedings of the W3C Workshop RDF Next Steps June 2627 2010 hosted by the National Center for Biomedical Ontology NCBO Stanford Palo Alto CA USA. Retrieved from http://www.w3.org/2005/Incubator/prov/wiki/images/3/3f/RDFNextStep_ProvXG-submitted.pdf.

Zhao, Jun and Hartig, Olaf. (2012) Towards Interoperable Provenance Publication on the Linked Data Web. LDOW 2012 Linked Data on the Web, WWW2012 Workshop on Linked Data on the Web. Lyon, France, 16 April, 2012, CEUR-WS.org, online CEUR-WS.org/Vol-937.

https://doi.org/10.23106/dcmi.952136081