Publishing and Consuming Provenance Metadata on the Web of Linked Data

Olaf Hartig[1] and Jun Zhao[2]

[1] Humboldt-Universität zu Berlin
hartig@informatik.hu-berlin.de
[2] University of Oxford
jun.zhao@zoo.ox.ac.uk

Abstract. The World Wide Web evolves into a Web of Data, a huge, globally distributed dataspace that contains a rich body of machine-processable information from a virtually unbound set of providers covering a wide range of topics. However, due to the openness of the Web little is known about who created the data and how. The fact that a large amount of the data on the Web is derived by replication, query processing, modification, or merging raises concerns of information quality. Poor quality data may propagate quickly and contaminate the Web of Data. Provenance information about who created and published the data and how, provides the means for quality assessment. This paper takes a first step towards creating a quality-aware Web of Data: we present approaches to integrate provenance information into the Web of Data and we illustrate how this information can be consumed. In particular, we introduce a vocabulary to describe provenance of Web data as metadata and we discuss possibilities to make such provenance metadata accessible as part of the Web of Data. Furthermore, we describe how this metadata can be queried and consumed to identify outdated information.

1 Introduction

During recent years an increasing number of data providers adopted a set of best practices for publishing and connecting structured data on the Web, leading to the creation of a globally distributed dataspace – the Web of Data [1]. While this dataspace holds an enormous potential, using data from the Web poses questions of information quality and trustworthiness. These questions can be addressed by methods that use provenance information about the data. We present approaches how such provenance information can be made available in the Web of Data.

1.1 The Web of Data

The best practices that enable the creation of the Web of Data are basically four principles that became known as the *Linked Data principles* [2]. These principles require to identify entities with HTTP URIs that can be resolved over the Web into data that describes the identified entity. This data is represented

D.L. McGuinness, J.R. Michaelis, and L. Moreau (Eds.): IPAW 2010, LNCS 6378, pp. 78–90, 2010.

using the Resource Description Framework (RDF). RDF is a generic data model that represents data using triples of the form (subject, predicate, object). Each element of such an RDF triple can be a URI or a local identifier for unnamed entities; objects can also be a literal. A set of RDF triples is called an RDF graph. The predicate in an RDF triple specifies how subject and object of the triple are related. These relationships as well as classes of entities are defined in vocabularies. Since vocabulary definitions can be represented as RDF data, vocabularies can also be published as Linked Data; the terms introduced in vocabularies just have to be identified with dereferencable HTTP URIs, enabling a Linked Data aware application to retrieve and utilize the definition of terms used in the currently processed data. Furthermore, the Linked Data principles require that the provided RDF graphs should include RDF links pointing to RDF data from other data sources on the Web. An RDF link is an RDF triple where the subject is a URI in the namespace of one data source and the object is a URI in the namespace of another source. By connecting data from different sources via RDF links a single, globally distributed dataspace emerges.

We call RDF graphs that can be retrieved by resolving URI references *Linked Data object*. Usually, Linked Data objects are part of a *linked dataset* which is a larger RDF graph that contains data about multiple entities. Typical approaches to create a linked dataset are Linked Data interfaces over native RDF stores and wrappers over relational databases or over Web APIs. Some wrappers materialize the created linked dataset, others convert the data on the fly.

1.2 The Need for Provenance Metadata in the Web of Data

The rapid growth and the wide adoption of the Web of Data is driven by the openness of the Web. The same linked dataset can be replicated and hosted at different locations on the Web, under the same or different URI namespaces. Different copies of linked datasets can be created using the same source data. Datasets can be connected by different sets of RDF links, created using different tools or methods and maintained by different publishers. The openness of the Web means that once the data and links are made available on the Web, these different copies of statements about the same set of entities –which might be in conflict and of varied quality– become completely interconnected and intertwined. Finding data about a specific entity may result in multiple URIs identifying this entity and linking to Linked Data objects from different sources. Which of these links should be followed? Which of the Linked Data objects provides more trustworthy or more up-to-date information about the entity? To answer these questions we need not only data about the entity but also information about how the data became available. Hence, we require information about the provenance of Linked Data.

We identify two main sources for obtaining provenance information about data: information recorded by the application that performs the provenance-based evaluation of the data and provenance-related metadata published by the providers of the data. Only a small amount of provenance can be recorded by applications itself if these applications process data consumed from the Web.

Hence, to obtain more complete knowledge these applications rely on provenance metadata from third parties such as the data providers. However, in a recent study [3] we discovered a general lack of provenance-related metadata about data on the Web. Reasons are the lack of suitable vocabularies to describe Web data provenance and a lack of tools to generate and provide provenance metadata.

1.3 Contributions and Structure

To overcome the problem of missing provenance metadata about Linked Data we present approaches to publish such metadata; and we discuss how this metadata can be retrieved and used in applications. To allow for a successful consumption of provenance metadata we conceive it as an absolute necessity that this metadata becomes an integral part of the Web of Data. This is only possible if the publication of this metadata adheres to the same principles that are used for the data itself. Therefore, we present a vocabulary that allows providers of Linked Data to describe the provenance of their data with RDF. Furthermore, we discuss how these RDF based provenance descriptions can be published as Linked Data on the Web. To reduce the required effort for this publication we extended several Linked Data publishing tools, enabling them to automatically provide provenance metadata. The main goal of consuming this provenance metadata is to assess quality and trustworthiness of data retrieved from the Web. Hence, we also discuss how this metadata can be retrieved and we demonstrate its use in an example scenario, identifying outdated information in the Web of Data.

This paper is organized as follows: Section 2 introduces our Provenance Vocabulary; in Section 3 we describe options to publish provenance metadata and we present our provenance extensions to Linked Data publishing tools. Section 4 discusses consuming provenance metadata and describes our experiment of using this metadata to compare the timeliness of data. Section 5 reviews related work and we conclude in Section 6.

2 Describing Provenance of Linked Data

Our aim is to enable Linked Data providers to offer provenance-related metadata in the form of Linked Data. Providing provenance information as Linked Data requires vocabularies that can be used to describe the different aspects of provenance. In this section we introduce our Provenance Vocabulary[1] and illustrate its use by a running example. Furthermore, we describe the design principles applied to the development of the vocabulary.

2.1 Overview of the Provenance Vocabulary

The Provenance Vocabulary is defined as an OWL ontology[2] and it is partitioned into a core ontology and supplementary modules. To avoid making the core

[1] http://purl.org/net/provenance/

[2] The introduction in this paper refers to revision 0.5 of the Provenance Vocabulary as is available at http://purl.org/net/provenance/ns-20100710

ontology too complex the modules provide less frequently used concepts and a broad range of specializations of the core concepts. At present we provide three supplementary modules: Types, Files and Integrity Verification.

The development of our vocabulary is motivated by the need to describe the main aspects of provenance of data consumed from the Web. In [3] we identify two main dimensions of provenance that are typical in this context: data creation and data access. Some, more general concepts, such as actors, processes, and artifacts, are relevant in both these dimensions. Consequently, the Provenance Vocabulary consists of three parts: general terms, terms for data creation, and terms for data access.

The **general terms** include classes for the general types of provenance elements: `Actor`, `Execution` and `Artifact`. `Actor` has sub-classes `HumanActor` and `NonHumanActor`; `Artifact` has sub-classes `DataItem` and `File`. Furthermore, the general terms include properties that relate individuals of the general classes with each other (cf. Figure 1, central section): an `Artifact` was `yieldedBy` an `Execution` which may have used further `employedArtifacts`. An `Execution` was `performedAt` a specific time; it was `performedBy` an `Actor`, and it might have had other `involvedActors`. A `NonHumanActor` was `operatedBy` a `HumanActor` and it may have `deployedSoftware`. An `Artifact` might have been `serializedBy` a `File`; a `DataItem` might have been `containedBy` another `DataItem`; and a `DataItem` might have been `precededBy` a former version of this item. Notice, some of these properties are abstract (`yieldedBy`, `involvedActor`, and `employedArtifact`) which means they are not intended to be used to describe instance data but to provide an abstract base for other properties.

With these general terms we can describe the main provenance elements of a running example using RDF data[3]:

```
<> a prv:DataItem ;
   foaf:primaryTopic <http://example.org/gene/0030840> ;
   foaf:topic <> .
<http://example.org /flybase> a void:Dataset ;
                              void:exampleResource <http://example.org/gene/0030840> .
<http://example.org/triplify> a prv:Actor, prv:NonHumanActor ;
                              prv:operatedBy <http://example.org/orga> ;
                              prv:deployedSoftware _:b1 .
_:b1 rdf:type doap:Version ;
     doap:revision "0.5" .
_:b2 rdf:type doap:Project ;
     doap:release _:b1 ;
     doap:homepage <http://triplify.org> .
<http://example.org/orga> a foaf:Organization , prv:Actor, prv:HumanActor .
```

This data describes: a data item which primarily represents data about a gene identified by the URI `http://example.org/gene/0030840`; a linked dataset, identified by `http://example.org/flybase`; and an instance of the Triplify service [4], a Linked Data publishing tool. This instance, identified by the URI `http://example.org/triplify`, is operated by organization `http://example.`

[3] We use RDF Turtle notation (http://www.w3.org/TeamSubmission/turtle/); URI namespace prefixes used are: `rdfs` for `http://www.w3.org/2000/01/rdf-schema#`, `prv` for `http://purl.org/net/provenance/ns#`, `prvTypes` for `http://purl.org/net/provenance/types#`, `doap` for `http://usefulinc.com/ns/doap#`, and `void` for `http://rdfs.org/ns/void#`

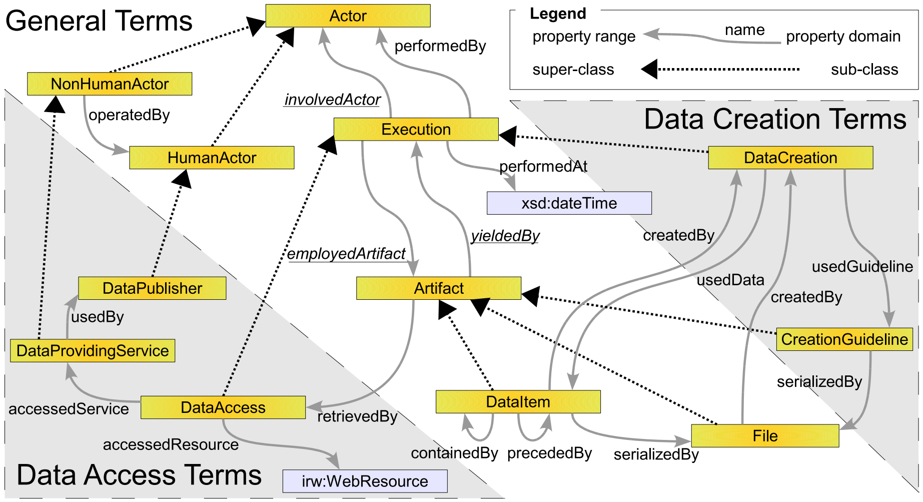

Fig. 1. Classes and properties defined by the Provenance Vocabulary core ontology

`org/orga`. In our running example, the linked dataset is a Linked Data version of FlyBase, the central genetic database for *Drosophila* research. Triplify publishes this dataset by creating Linked Data objects on the fly, using results of queries to the JDBC endpoint of the relational FlyBase database. The data item in the description represents such a Linked Data.

The terms in the **data creation** dimension (cf. Figure 1, upper-right section) describe how a `DataItem` has been `createdBy` a `DataCreation`. The property `usedData` refers to source data used during a `DataCreation`; `usedGuideline` refers to guidelines such as transformation rules or mapping definitions that were used to guide a `DataCreation`. Using the data creation terms, the creation in our running example could be described as follows:

```
<http://example.org/triplify> a prvTypes:DataCreatingService .
<> prv:createdBy [
        a prv:DataCreation ;
        prv:performedAt "2010-03-01T12:38:42+00:00"^^xsd:dateTime ;
        prv:performedBy <http://example.org/triplify> ;
        prv:usedData _:x ;
        prv:usedGuideline _:y ] .
_:x a prv:DataItem ;
    foaf:homepage <http://flybase.org/> ;
    prv:createdBy [ a prv:DataCreation ;
                    prv:performedAt "2010-02-19T00:00:00+00:00"^^xsd:dateTime ] .
_:y a prv:CreationGuideline , prvTypes:TriplifyConfiguration  ;
    prv:createdBy [ a prv:DataCreation ;
                    prv:performedBy <http://example.org/orga> ] .
```

The example data item was created by a `DataCreation` execution performed by the Triplify service on Mar.1, 2010. The creation was based on unnamed source data from the Feb.19, 2010 release of the FlyBase database; the creation was guided by an unnamed Triplify configuration created by the organization who operates the Triplify service.

The **data access** dimension (cf. Figure 1, lower-left section) focuses on retrieving data items from the Web. Using the data access terms in provenance descriptions is, in particular, recommended to provide information about the retrieval of source data items and of creation guidelines. The Provenance Vocabulary allows to describe how a `DataItem` has been `retrievedBy` the execution of a `DataAccess`. The retrieved `DataItem` is a Web representation of the `accessedResource`. The `accessedService` is a `DataProvidingService` which was `usedBy` the `DataPublisher`; furthermore each `DataProvidingService` is usually `operatedBy` a `HumanActor`. In our running example, the Triplify service retrieved the FlyBase relational data that was used to create the example Linked Data object from the FlyBase JDBC endpoint:

```
_:x prv:retrievedBy [
            a prv:DataAccess ;
            prv:accessedService [ a prv:DataProvidingService , prvTypes:JDBCService ;
                                  foaf:homepage <http://flybase.org/> ] ;
            prv:performedAt "2010-03-01T12:38:42+00:00"^^xsd:dateTime ;
            prv:performedBy <http://example.org/triplify> ] ] .
```

Notice, since the Linked Data object was created on the fly, the execution time of the data access described is equal to the creation time of the object.

To allow for a wide range of applications the vocabulary does not prescribe a specific granularity by which provenance information has to be described. Hence, the classes in the core ontology are quite general. For instance, a `DataItem` could a single RDF triple or it could be a specific RDF graph that represents a Linked Data object as in the example. More specific specializations of the general classes are provided with the types module. However, while our vocabulary, including its modules, provides a basic framework to describe the provenance of data from the Web it does not aim to support the description of every aspect and detail of provenance. In particular, to provide a detailed description of a specific data creation we propose to use more specialized vocabularies and associate these descriptions with the corresponding `DataCreation` entity. In the documentation for our vocabulary we propose some examples of how other vocabularies can be used together with the Provenance Vocabulary.

2.2 Design Principles of the Provenance Vocabulary

We develop the Provenance Vocabulary with understandability and usability in mind. For this reason we apply a consistent scheme for property names, using the simple past form of a verb followed by a class name or the preposition *by*. Furthermore, we omit inverse properties to avoid interoperability problems in Linked Data consuming systems that do not apply OWL-DL based reasoning in many cases.

Some of the properties in our vocabulary are shortcuts, allowing for a more convenient use. For instance, many data creations are based on the creation of a file that serializes the created data item. Since it is more convenient to describe these file-based data creations implicitly by referring to the creation of the file instead of the data item itself, our vocabulary provides additional terms for these file-based descriptions. Hence, it is also possible that a `File` has been `createdBy` a `DataCreation`; this implies the `DataItem` that was `serializedBy` the `File` was also created by the same `DataCreation`. The vocabulary definition includes rules to enable reasoners to infer such kind of implications. Similarily, the properties `usedGuidelineFile` and `usedDataFile` introduced in the Files module are alternatives to `usedGuideline` and `usedData`, respectively.

Another good practice for Linked Data vocabularies is the interlinking of related terms between vocabulary definitions. Such "schema-level links" improve the degree to which published data is self-describing. The Provenance Vocabulary adheres to this practice. For instance, the `Actor` class is defined to be equivalent to the `Agent` class in the FOAF vocabulary. This relationship enables a FOAF-aware application to infer actors in a provenance description are FOAF agents and to deal with them accordingly, e.g., in visualizations.

3 Publishing Provenance Descriptions about Linked Data

To achieve the goal of integrating provenance of Linked Data into the Web of Data it is not only necessary to provide a vocabulary but also to actually make the provenance descriptions available to Linked Data based applications. Therefore, we provide recommendation for publishing provenance-related metadata as Linked Data in this section. These recommendations should be understood as a proposal while best practices still have to emerge.

The primary location of metadata about a linked dataset is its voiD description, that is, an RDF document on the Web which describes the dataset based on the Vocabulary of Interlinked Datasets (voiD) [5]. A voiD description should comprise general provenance information for the described dataset. In addition to general provenance information about a linked dataset we suggest to provide more detailed information with each access to the dataset. There are basically three options to provide access to a linked dataset on the Web: Linked Data objects, RDF dumps, and SPARQL endpoints. While these options do not exclude each other they require the application of different provenance publication approaches as we discuss in the remainder of this section.

3.1 Adding Provenance to Linked Data Objects

The Linked Data object that can be retrieved by resolving the HTTP URI for an entity is an RDF graph that –according to the Linked Data principles– contains data about the entity identified by the URI. We propose that these Linked Data objects additionally contain provenance-related metadata (i.e. additional RDF triples) about themselves and about the contained RDF triples. Provenance of specific RDF triples could be described using RDF reification. The provenance of the whole Linked Data object should be expressed as illustrated by our running example: the provenance metadata presented in Section 2.1) describes a representation of a Linked Data object. To accelerate the adoption of the practice to augment Linked Data objects with (provenance) metadata we extended several, widely used Linked Data publishing tools as we describe in Section 3.4.

If possible, the provided provenance description should also comprise detailed provenance information about source data and creation guidelines that have been used during the creation of the Linked Data object. Furthermore, the provenance description should cover the linked dataset of the Linked Data object. However, instead of augmenting the object itself with provenance metadata about its dataset we propose to link to a voiD description using an HTTP URI that identifies the dataset (as illustrated in the running example).

3.2 Adding Provenance to RDF Dumps

A linked dataset can be provided as an RDF dump, that is, an RDF document which contains the whole linked dataset. Usually, an RDF dump represents a linked dataset as a single RDF graph. We propose to augment this graph with provenance metadata about itself, similar to the practice proposed in the previous section for Linked Data objects. However, in this case the added provenance metadata describes the provenance of the whole dataset and, thus, is likely to be similar to the information provided with a voiD description for the dataset. In addition to this information the metadata should also describe the provenance of the RDF dump itself.

It is also possible to serialize a linked dataset as a collection of Named Graphs [6], i.e. RDF graphs named with a URI. In this case each of these graphs could contain provenance metadata about itself. Alternatively, the collection of Named Graphs could contain an additional Named Graph that describes the provenance of the other graphs.

3.3 Providing Provenance Information at SPARQL Endpoints

A third possibility to provide access to a linked dataset is via a SPARQL endpoint, i.e., a query service that executes SPARQL queries over the dataset. SPARQL is the query language for RDF data. We propose to make provenance metadata a part of the dataset published via such a SPARQL endpoint so that queries can ask for provenance information. Furthermore, a provenance-enhanced SPARQL query engine could also add provenance metadata automatically to query results. SPARQL defines four different query result forms: select, construct, describe, and ask. The result of construct and describe queries is an RDF graph. Similar to the practice proposed for Linked Data objects, a provenance-enhanced SPARQL query engine could add provenance metadata to these result graphs. The result of a select query is a set of variable bindings that can be represented as a table; ask queries result in a boolean value. To exchange these types of results over the Web, SPARQL endpoints usually serialize the results using a standard XML format or a JSON format. It requires future work to define a possibility how these serializations can be extended with provenance descriptions.

3.4 Metadata Extensions that Simplify the Publication

A large-scale augmentation of the Web of Data with provenance metadata can only be achieved when the effort for creation and for publication is kept to a minimum. For this reason, we extended several tools that are widely used for publishing Linked Data on the Web, including Triplify, Pubby[4] and D2R server[5], with a metadata component [7]. These new components automatically generate and serve provenance metadata with Linked Data objects as proposed in Section 3.1. Due to our extensions data publishers can easily enrich their data with provenance metadata by simply configuring a few parameters, such as the name and the URI identifying the publisher or the URI of the dataset. Hence, with data providers upgrading their Linked Data servers to the latest release of these tools we can expect a significant increase in the amount of provenance information added to the Web of Data.

4 Consuming Provenance from the Web of Data

Consuming provenance information from the Web of Data includes retrieving provenance metadata from the Web and making use of it. In this section, we present approaches to query for provenance metadata and we demonstrate its use in an example scenario, identifying outdated information in the Web of Data.

4.1 Querying for Provenance Metadata

A simple approach to query for provenance requires that provenance metadata is accessible through the SPARQL endpoints for linked datasets as we propose in Section 3.3. This practice enables applications to issue queries as in the following example:

Example 1. SPARQL query (prefix declarations omitted) that asks for the creation time of the source data used to create a linked dataset.

```
SELECT ?creation_time WHERE {
  <http://example.org/dataset> prv:createdBy [ prv:usedData ?source_data ] .
  ?source_data prv:createdBy [ prv:performedAt ?creation_time ] }
```

If the provenance metadata is provided as a part of Linked Data objects (cf. Section 3.1) and the metadata is properly interlinked (i.e. it includes links to voiD descriptions etc.) then provenance can be queried using the link traversal based query execution paradigm [8] as implemented in SQUIN[6]. This query approach evaluates SPARQL queries as in Example 2 over a dataset that is continuously augmented with Linked Data objects from the Web. These objects are discovered by following RDF links that correspond to partial query results.

Example 2. SPARQL query asking for the creation time of the source data used to create a Linked Data object about a specific gene.

```
SELECT ?creation_time WHERE {
  ?data foaf:primaryTopic <http://example.org/gene/0030840> .
        prv:createdBy [ prv:usedData ?source_data ] .
  ?source_data prv:createdBy [ prv:performedAt ?creation_time ] }
```

In the remainder of this section we present an example scenario in which we retrieve provenance information using SQUIN to execute queries as in Example 2.

[4] http://www4.wiwiss.fu-berlin.de/pubby/

[5] http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/

[6] http://squin.org

4.2 The Example Scenario

Our experiment scenario is based on two databases that provide complementary knowledge for *Drosophila* genetic research: FlyBase and another relational database, FlyTED, which is a specialized gene expression image repository for *Drosophila* testis. Using these databases we create three linked datasets and publish their provenance information using our Provenance Vocabulary and voiD.

Our first dataset, D_{FB}, is created by transforming a subset of FlyBase on-the-fly using Triplify as in our running example. The other two datasets, D_{FT1} and D_{FT2}, are created by transforming two different snapshots of FlyTED into RDF and publishing these RDF dumps as Linked Data using Pubby.

While the provenance of Linked Data objects about genes from D_{FB} corresponds to our running example, Figure 2 illustrates the provenance of a Linked Data object gd_j about a gene from D_{FT1} or D_{FT2}. Each gd_j is created by a Pubby instance that accesses the SPARQL endpoint for the corresponding FlyTED linked dataset; this endpoint executes queries over the RDF dump created by transforming the corresponding FlyTED database snapshot.

Our metadata extensions to Triplify and Pubby (cf. Section 3.4) provide provenance metadata for all the Linked Data objects from the three datasets. Additionally, we, manually, create voiD descriptions with provenance metadata for these datasets and publish these descriptions as Linked Data on the Web.

For biologists interested in a more complete knowledge about genes it is useful to connect FlyBase and FlyTED. We may create `owl:sameAs` links between genes from D_{FB} and those in D_{FT1} and D_{FT2}. However, without additional context information a search for FlyTED gene entities that are `owl:sameAs` to a FlyBase gene will return all matching FlyTED genes no matter when their data was created. Some of these data mappings might no longer be correct because gene names are changed regularly in biological databases, whenever additional knowledge about genes and their functions becomes available. The goal of our scenario is to identify those genes from D_{FB} that are mapped to multiple genes from D_{FTn} and to analyze whether some of the mappings point to outdated information.

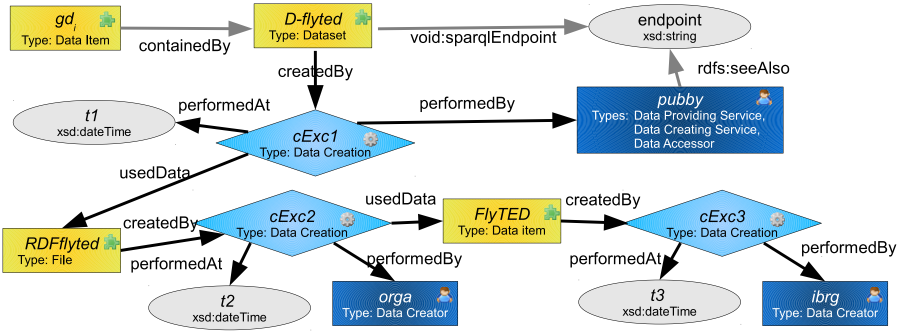

Fig. 2. Illustration of the creation process of a FlyTED gene data object

4.3 Comparing the Timeliness

Each FlyTED gd_i is part of the linked dataset D_{FT1} or D_{FT2}. Since the creation of these Linked Data objects is performed by the Pubby instance accessing the SPARQL endpoints, the timeliness of these objects depends on the freshness of the data used for creating the endpoint, i.e., the RDF dumps. The timeliness of these dumps depends on the timeliness of the original FlyTED database snapshots. In order to identify which gene data is more outdated we need to compare their timeliness. Because all gd_i are generated on the fly by Pubby, they all have the same creation time. Hence, we need to compare the creation time of their source data, the FlyTED database snapshots.

A SPARQL query similar to the query in Example 2 can be used to retrieve this information. For example, the two FlyTED gene entities $CG12993$ and *p-cup* both have an `owl:sameAs` relationship to the same gene in FlyBase. Using link traversal based query execution we search for the creation time of the source data used to create data about these genes. The query result shows that the creation time of the source data about $CG12993$ is earlier than *p-cup* because the Linked Data object $gd_{CG12993}$ about $CG12993$ is derived from a gene record from an older version of FlyTED and therefore it is less fresh than $gd_{\text{p-cup}}$ for *p-cup*. Based on these results, we conclude that $gd_{CG12993}$ is more outdated than $gd_{\text{p-cup}}$. In fact, the gene name CG12993 is no longer used by the community and is now replaced by p-cup. The Linked Data object $gd_{CG12993}$ might contain outdated and misleading information about this p-cup gene. Linked Data users should choose the more up-to-date gene URI if they would like to access more accurate knowledge.

Based on an analysis of all the gene entities from D_{FT1} and D_{FT2} we found that 9 different FlyBase genes are mapped to more than one FlyTED genes. For each of these FlyBase genes we compared the timeliness of the data about the FlyTED genes mapped to the FlyBase gene. This comparison revealed that all these FlyBase genes are linked to at least one outdated FlyTED gene URI, all of which have been replaced in the more up-to-date FlyTED linked dataset.

This small experiment is just one example which shows how crucial it is to assess the timeliness of Linked Data objects. Without this contextual information, users of Linked Data face the danger of using poor quality data that might contain wrong information without even being aware of it. Our experiment is a very first step of demonstrating the importance of integrating provenance in the Web of Data and the importance of provenance metadata for reducing potential errors in Linked Data applications and, thus, enhancing the trust in Linked Data.

5 Related Work

Representing and analyzing provenance is a topic of research since many years [9]. While many approaches exist for representing provenance of data creation [10,11], none of these explicitly addresses the characteristics of data access, e.g., the retrieval of data from the Web. Although this type of provenance is not always required in self-contained systems such as a DBMS or a workflow management system, it needs be captured for the Web of Linked Data. The Provenance Vocabulary presented in this paper allows to describe both aspects, data creation and data access.

Many related work on provenance for the Web have emerged in the context of Semantic Web research. Harth et al. [12] propose a "social dimension to associate provenance with the originator (typically a person) of a given piece of information". Our Provenance Vocabulary encourages to represent human actors and their relation to data items.

Ding et al. [13] understand the provenance of RDF data as the RDF graphs of which parts of an analyzed RDF graph has been derived from. The authors argue that tracking complete RDF graphs is too coarse-grained and that a representation on the level of single RDF statements is unsuitable, too. Hence, Ding et al. introduce RDF molecules as the finest sub-graphs to decompose an RDF graph. Our vocabulary models data items on an abstract level. They can represent data of any level of granularity: RDF graphs, statements, or RDF molecules.

Da Silva et al. use the term knowledge provenance to refer to information about the origin of knowledge and about the reasoning processes used to produce answers [14]. In [15] the authors present the Proof Markup Language to describe justifications for results of an answering engine or a reasoner. These justifications may describe the execution of a specific type of data creation process modeled by our Provenance Vocabulary.

Moreau et al. propose the Open Provenance Model (OPM) which aims to provide a community-compliant, general-purpose provenance model [16]. OPM contains many concepts similar to the general terms in our vocabulary. However, in contrast to the domain-independent approach of OPM our vocabulary explicity addresses the provenance of Linked Data published on the Web. Hence, our vocabulary can be defined as an OPM profile, created for the Web of Data application domain. Such an alignment with OPM will help us to ground our vocabulary with a community data model and, thus, is part of our future work.

Similarily to OPM, Sahoo et al. propose Provenir [17], an upper-level provenance ontology that defines abstract classes and properties which can be refined for a specific domain. An alignment of the Provenance Vocabulary with Provenir is also part of our future work.

6 Conclusion

This paper presents a Provenance Vocabulary that assists providers of Linked Data to describe the provenance of their data using RDF. We explain how this vocabulary can be used to describe data items of different granularity and we propose approaches how metadata with such provenance descriptions can become an integral part of the Web of Data. Furthermore, we discuss how such provenance metadata can be consumed in order to support an example scenario of identifying outdated information from the Web of Data. An alignment with other existing provenance models and vocabularies is part of our future work. We will also continue our investigation of using provenance to support the evaluation of information quality.

References

1. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - The Story So Far. In: Int. Journal on Semantic Web and Information Systems. Special Issue on Linked Data (2009)
2. Berners-Lee, T.: Design issues: Linked data, `http://www.w3.org/DesignIssues/LinkedData.html` (retrieved March 19 2010)
3. Hartig, O.: Provenance Information in the Web of Data. In: Proceedings of the Linked Data on the Web Workshop (LDOW) at WWW (2009)
4. Auer, S., Dietzold, S., Lehmann, J., Hellmann, S., Aumueller, D.: Triplify: Lightweight linked data publication from relational databases. In: Proceedings of the 18th International Conference on World Wide Web, WWW (2009)
5. Alexander, K., Cyganiak, R., Hausenblas, M., Zhao, J.: Describing linked datasets. In: Proceedings of the Linked Data on the Web Workshop (LDOW) at WWW (2009)
6. Carroll, J.J., Bizer, C., Hayes, P., Stickler, P.: Named graphs, provenance and trust. In: Proceedings of the 14th International World Wide Web Conference, WWW (2005)
7. Hartig, O., Zhao, J., Mühleisen, H.: Automatic integration of metadata into the web of linked data. In: Proceedings of the Demo Session at the 2nd Workshop on Trust and Privacy on the Social and Semantic Web (SPOT) at ESWC (2010)
8. Hartig, O., Bizer, C., Freytag, J.C.: Executing SPARQL queries over the web of linked data. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, Springer, Heidelberg (2009)
9. Bose, R., Frew, J.: Lineage retrieval for scientific data processing: A survey. ACM Computing Surveys 37(1) (2005)
10. Simmhan, Y., Plale, B., Gannon, D.: A Survey of Data Provenance in e-Science. SIGMOD Record 34(3) (2005)
11. Tan, W.C.: Provenance in Databases: Past, Current, and Future. IEEE Data Engineering Bulletin 30(4) (2007)
12. Harth, A., Polleres, A., Decker, S.: Towards a Social Provenance Model for the Web. In: Proceedings of the Workshop on Principles of Provenance (2007)
13. Ding, L., Finin, T., Peng, Y., da Silva, P.P., McGuinness, D.L.: Tracking RDF Graph Provenance using RDF Molecules. Technical Report TR-CS-05-06, UMBC (2005)
14. da Silva, P.P., McGuinness, D.L., McCool, R.: Knowledge Provenance Infrastructure. Data Engineering Bulletin 26(4) (2003)
15. da Silva, P.P., McGuinness, D.L., Fikes, R.: A Proof Markup Language for Semantic Web Services. Information Systems 31(4-5) (2006)
16. Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., Kwasnikowska, N., Miles, S., Missier, P., Myers, J., Plale, B., Simmhan, Y., Stephan, E., den Bussche, J.V.: The open provenance model core specification (v1.1). In: Future Generation Computer Systems (in Press 2010) (accepted Manuscript)
17. Sahoo, S., Thomas, C., Sheth, A., York, W., Tartir, S.: Knowledge modeling and its application in life sciences: a tale of two ontologies. In: Proceedings of the 15th International Conference on World Wide Web, WWW (2006)