Electronic Government and Electronic Participation
E. Tambouris et al. (Eds.)

© 2015 The authors and IOS Press.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License.
doi:10.3233/978-1-61499-570-8-101

Fusepool P3: A Linked Data Platform for Open Government Data

Luigi SELMI[a,1] and Alessia NEURONI[a]
[a] *Bern University of Applied Sciences*

Abstract. Many public administrations have been publishing data sets as open data for the last few years through portals based on CKAN or other platforms. Applications have been developed using, most of the time, one single data set as developers, data journalists and small businesses cannot afford the cost of re-using and integrating different data sets. This is due to many factors as the different formats used for the data, lack of documentation, different metadata and lack of public, trustworthy registries for entities of interest that could ease the task of connecting information provided by different authors about them. The Linked Data principles offer the guidelines to solve all these issues by leveraging the Semantic Web standards for describing resources on the Web. Moreover the Linked Data Platform specification, a recent W3C Recommendation, provides a realization of the guidelines defining some basic interactions between a client and a server to manage resources using the HTTP protocol. The purpose of this article is to present the Fusepool P3 platform that extends the LDP specification to support services which transform raw data sets into RDF format and enrich them.

Keywords. Linked Data, Linked Data Platform, Data Integration, Application Integration

1. Introduction

In the last years there has been an increase in the awareness of the importance of citizens' engagement and participation in society. Governments have been often criticized for being opaque and the mainstream media are seen by many mostly at the service of interested parties. Participation and engagement must be based on information and above all information produced by public administrations. Furthermore, opening the public sector information assets could result in economic gains estimated up to €40 billion a year in the EU [1]. Citizens, SMEs, tax payers accustomed to use the Web to communicate, buy stuff, learn, share information, do business, want to have access to the facts not just opinions and have boosted the request for the administrations to provide their information and services online. A strong impulse to publish raw data sets as they were produced by the public administrations has come from the European Commission directive of 2003 on the re-use of public sector information. The subsequent request for open data was raised to address two main issues: the formats in which the data sets were provided and the licenses for re-use. Clearly open, non-proprietary formats like CSV or XML are easier

[1] Corresponding Author: Senior Researcher, Bern University of Applied Sciences; E-mail: luigi.selmi@bfh.ch

to extract information than Excel spreadsheets or pdf files but cannot help in overcoming the semantic interoperability issue that arises when trying to merge different data sets. A solution to this last issue is offered by the Semantic Web standards in particular when they are used with the Linked Data principles as guidelines. Briefly, the proposal of the Semantic Web is to give any entity of interest a global identifier and a description using the RDF data model and terms from shared vocabularies. The Linked Data principles add some clarification such as the use of HTTP URI as global identifier and to provide links to other entities. Opening the data is the first step to a data driven society that requires huge changes in workflows and roles within each public administration and a commitment to deliver useful and reliable information to the citizens. Taking the next step towards the Semantic Web vision requires a stronger commitment to provide, maintain and make available name registries of entities of public interest using shared vocabularies. The idea is that an investment in such direction will provide further indirect returns to governments.

This paper discusses the issues that have to be addressed to merge the growing number of open data sets delivered by public administrations at different level and presents the Fusepool P3[2] platform that is being developed for publishing Linked Data from raw data sets. The paper is organized as follows. In the next Section 2 a short description of the Linked Data principles is introduced. Section 3 describes the Linked Data Platform, a formal specification of the principles. Section 4 describes different stages of a resource in the process of being transformed from raw to RDF format and how the Fusepool P3 platform enables the data transformation process and the publication of the result as Linked Data. In Section 5 are described some of the platform components that are being developed to transform and enrich the data sets. Last, in Section 6, the conclusions and future work.

2. Linked Data Principles

Linked Data [2] is a proposal published by Tim Berners-Lee in 2006 as a new way to publish data on the Web. It is based on four rules

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL)
4. Include links to other URIs so that they can discover more things.

The Linked Data principles aimed at providing some implementation guidelines to the realization of the Semantic Web. URI and RDF, the two main standards of the Semantic Web used to name and describe distributed resources, are quite abstract and do not enforce developers to use any other Web standard such as the HTTP protocol to access and use those resources. In the Semantic Web a URI is a mean to name an object, real or abstract, but it does not mandate for a representation of the object to be retrievable as a resource on the Web, not even when it is a HTTP URI. An application or software agent cannot rely on the availability of a description on the Web of any of the component of a RDF triple. The Linked Data principles do enforce the use of

[2] http://p3.fusepool.eu/

dereferenceable HTTP URIs, namely URLs, as name for things and the provision of information in RDF format when dereferencing any URI named thing. The guidelines have been successful as many research projects have been started from them. One of the main projects is LOD2[3] in which a complete set of tools have been integrated in the LOD2 Stack distribution to support the whole life cycle of linked data sets from data modeling to semantic lifting, interlinking, information extraction and data storage. Many RDF data sets, that have been published as Linked Data and interlinked, form the Linked Open Data Cloud[4], a growing RDF graph of interconnected data sets available online.

3. Linked Data Platform

As the Linked Data principles promote integration of data sets at the Web scale through standards such as dereferenceable HTTP URI, RDF, SPARQL and others, they do not provide a formal specification on how a client and a server should communicate in order to create a resource, or organize collections of resources. Most of the websites that publish data as Linked Data provide only read access to resources through a SPARQL endpoint or RDF data dumps following best practices [3] and patterns [4]. No write access is usually provided. The main effort of many research projects was to produce a reasonable amount of RDF data to bootstrap the Semantic Web. Once the data sets were published on a SPARQL 1.0 endpoint or in form of zipped RDF files, the job was mostly done. In 2009 Tim Berners-Lee wrote a follow-up [5] to the Linked data principles focusing on write access to web resources. HTTP, WebDAV and SPARQL Update protocols were foreseen as good candidates to support write access to Linked Data. The first attempt to implement a full read-write platform based on the Linked Data principles has been done by IBM as they were investigating the Linked Data approach for integrating their requirements engineering tool IBM Rational with other development tools. IBM then joined the W3C to work on a specification. The Linked Data Platform specification [6] has become very recently a W3C recommendation. A Linked Data Platform can handle all type of resources such as plain text, semi-structured text and multimedia like images, video or audio. A resource is named by a HTTP URI and can be a RDF source or any other type of resource, namely a Non-RDF source. A LDP container is a RDF source into which other resources can be put. A LDP container can be further specialized as basic, direct or indirect. A basic container has a containment relationship with its resources while the relationship between a direct or indirect container with its resources can be further specified. An HTTP server that implements all the mandatory requirements listed in the LDP specification becomes a LDP platform. A client can look up a representation of a resource in a LDP platform by simply sending an HTTP GET request to its URI. A client can create a LDP resource sending an HTTP POST request to a LDP container with a representation of the resource. If the creation of the resource is successful the server will send a response message with the “Location” header containing the URI of the created resource. A client can create a new container sending an HTTP POST request to a container with a RDF description of the type of container. A client can also retrieve a list of all the resources within a container. An LDP container can be seen as a

[3] http://lod2.eu/
[4] http://lod-cloud.net/

folder in a file system that can be accessed to read and write LDP resources from the Web.

4. Fusepool P3: An Extended Linked Data Platform Implementation

The European FP7 project Fusepool P3 aims at extending the LDP specification to address the needs of open data publishers and consumers. It adds to LDPs a new type of container that can be configured to host the result of a data transformation processes.

Data owners and publishers that want to get four or five stars of the Tim Berners-Lee's deployment scheme[5] for their data sets must send them through one of more transformation steps such as semantic lifting, reasoning, interlinking and information extraction. Starting from a Non-RDF resource the data can be transformed into RDF through a process called semantic lifting. Creating a RDF representation of a Non-RDF resource implies putting in an explicit form the semantics of the resource metadata. This task is performed designing a model of the resource based on ontologies and defining the rules to map the resource metadata to the terms in the ontologies. The mapping rules can be defined for example by XSL transformations or other mapping languages. During this process new entities are given a URI following a pattern defined in the XSL stylesheet. When more data sets are transformed in RDF, terms coming from different vocabularies can be used that define the same meaning or there might be an interest in inferring new relations among the entities described in different data sets. As an example, in a data set some entities can be described as being hotels, according to an ontology like schema.org[6], while in another data set other entities can be described as restaurants and a user might want to know all the local businesses in both data sets. A local business can be defined in the same ontology as an upper class of both hotel and restaurant but an application cannot answer the user query as the fact that each hotel or restaurant is also a local business has not yet been made explicit [7]. Most of the time these inferences are not stored in a database as they can be made at runtime by a reasoner on behalf of a user agent. There might be cases in which different URIs are used in a data set to denote the same entity. Finding and replacing such duplicates is a well-known issue in any data integration project [8]. The task is performed comparing descriptions of the entities through functions that return a numeric value depending on the similarity of the descriptions. When the returned value is higher than an upper threshold, the entities are automatically denoted as equal and a triple is added to the RDF data set to state the inferred fact between the two entities, often using the *owl:sameAs* relation from the OWL ontology. When the returned value falls below a lower threshold, the entities are different and no action is performed. In a third case, when the returned value falls between the thresholds, a human intervention can be required to state whether two entities are equivalent or not. The same task can be performed to interlinking entities described in different data sets in order to merge their descriptions. Further enrichments can be added to a data set extracting information from plain text provided in the description of a resource through NLP processes or enabling manual text annotations by experts.

In Fusepool P3 a transformation process can be performed by a single component or a chain of components. A component is defined as a "transformer" in the project

[5] http://5stardata.info/
[6] http://schema.org/

documentation and must implement a REST API. The input data can be sent via a HTTP POST request to be transformed or enriched. The input and output formats supported by a transformer can be retrieved sending a HTTP GET request to it. A transformer can take the address of another resource as a parameter to be used as rules to adapt its behavior to the input data. As an example, a transformer has been made available to transform data from XML to RDF. The transformer needs a XSL stylesheet to perform a transformation from a specific XML file into RDF using terms from one or more vocabularies. Different transformers of the same type can be instantiated through a factory changing the parameter value sent in the request, namely the URL of the XSL stylesheet. The architecture based on the REST style and HTTP protocol enables two or more transformers to be used in chain. A pipeline transformer can be configured as an ordered list of transformers that will send the data through each of them. A basic check of the compatibility between the output format of a transformer and the input format of the following one is performed before a pipeline can be instantiated by its factory. A transformer responds synchronously to all requests. The response message can contain the result of the transformation if the transformer has been configured to process the input data synchronously or it contains the URI of the resource that will be created as a result of the transformation if the transformer has been configured to process the data asynchronously. The Fusepool P3 platform has been tested with two LDP implementations, Apache Marmotta [7] and OpenLink Virtuoso[8] which both provide also a triple store for persistence of the RDF data and a SPARQL endpoint. The Fusepool P3 platform adds to the LDPs a proxy to which a client can send a HTTP POST request to create a transforming container. As an example in the tourism domain we considered an application developer that wanted to transform a data set about cultural events in RDF in order to merge the result with information about points of interest in the same area. The data is provided by the local tourism office on a daily basis as an XML file at the URL http://wonderland.com/events.xml. A transformer that is able to interpret XSL stylesheets can be instantiated from its factory and made available at the URI http://sandbox.fusepool.info:8164 pointing to a XSL stylesheet at the URL http://tourismtoday.com/wonderevents.xsl. The full URI of the transformer instance will use the encoded URL of the stylesheet

http://sandbox.fusepool.info:8164/?xslt=http%3A%2F%2Ftourismtoday.com%2Fwonderevents.xsl

A client can send an HTTP POST request with the XML document to the transformer to get the result of the transformation or it can create a transforming container using the URI of the transformer instance. A transforming container can be created sending to the LDP proxy a message with a description of the container in which is stated the URI of the transformer that will be used whenever a client will send some data to it. The description of the transforming container must be provided as RDF

[7] http://marmotta.apache.org/
[8] http://virtuoso.openlinksw.com/

```
@prefix ldp: <http://www.w3.org/ns/ldp#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix eldp: <http://vocab.fusepool.info/eldp#> .

<> dcterms:title "Wonderland Events Transformer"@en ;
    dcterms:description "Wonderland Events Transformer Container" ;
    a ldp:DirectContainer ;
        eldp:transformer
<http://sandbox.fusepool.info:8164/?xslt=http%3A%2F%2Ftourismtoday.com%2Fwonderevents.xsl>
```

A new transforming container with relative path /ldp/wonder/ can be created in the root platform container /ldp/ or in other containers. After creating a file "mycontainer.ttl" with the description of the container a client can send the request to create it to the LDP proxy whose address is http://sandbox.fusepool.info:8181

```
curl -i -X POST -H "Content-Type: text/turtle" -H "Slug: wonder" -d @mycontainer.ttl
http://sandbox.fusepool.info:8181/ldp/
```

If the request is successful the proxy will send a response message with the URI of the created container in the "Location" header. Once the transforming container has been created a client can send to it a request with the XML data to transform

```
curl -i -X POST -H "Content-Type: application/xml" -d @events.xml  \
http://sandbox.fusepool.info:8181/ldp/wonder/
```

The LDP proxy will send the data to the transformer that was linked in the container description. The proxy will create two new LDP resources in the container, a Non-RDF source with the input data and a RDF source containing the result of the transformation. The application developer might also be interested in enriching her RDF data linking the entities to those in DBpedia [9], a multilingual RDF knowledge base extracted from Wikipedia, or she might want to extract the entities named in textual parts and link them to the event. The developer will have to create an instance of the pipeline transformer with the list of the transformers' instances needed for the tasks and then create a new transforming container with a reference to the pipeline URI in the description.

5. Fusepool P3 transformers and administration tools

The Fusepool P3 platform provides a growing number of transformers, based on tools developed in open source projects or online web services. Some transformers that are being developed are shown in Table 1. They can be used in different phases of a transformation pipeline: semantic lifting, interlinking, information extraction, text annotation.

Table 1. Fusepool P3 transformers

Transformer	Scope
P3-BatchRefine	Semantic lifting
P3-Xslt	Semantic lifting
P3-DataTxt	Information extraction
P3-DictionaryMatcher	Information extraction
P3-Silkdedup	Interlinking
P3-Pundit	Annotation

The P3-BatchRefine transformer is based on OpenRefine[9] and its RDF extension. It can be used to transform tabular data in RDF. A JSON resource file with transformation rules specific for the input data must be provided as a parameter to the transformer. The P3-Xslt transformer can be used to transform XML files into RDF format as discussed in Section 4. The P3-DataTxt transformer has been implemented as an Apache Stanbol enhancer and provides a REST API like any other NLP engines available in Apache Stanbol[10]. It is based on an online web service and can be used to extract named entities from text. The P3-DictionaryMatcher extracts named entities in text that match with concepts in a SKOS taxonomy. The P3-Silkdedup transformer is based on SILK [10] and it can be used in deduplication or interlinking tasks. A configuration file with the linkage rules to compare the descriptions of the entities in the source data set and in the target one must be provided as a parameter. The P3-Pundit transformer supports a user interaction mechanism to assist experts in manually annotating text. The Fusepool P3 platform integrates some of the tools available in the LOD2 stack as distributed components that can communicate through REST API without requiring the installation of any of them in the same machine. Other components can be developed in Java or any other programming language, and added in a pipeline to easily implement specific use cases. As an example, many data sets contain geographic information for which further transformations can be required such as coordinates transformation and geocoding. User interfaces are being developed to easily register transformers, create pipelines and import data sets to be transformed and retrieved via LDP protocol or SPARQL.

6. Conclusions and Future Work

The Linked Data Platform specification is a new W3C recommendation that formalize the Linked Data guidelines with which a client can create, read, update and organize resources on a server using the HTTP protocol and standards like HTTP URI to name them and the RDF data model to provide a representation of the resource and its relationship with other resources on the Web. The Fusepool P3 platform aims at adding to it a mechanism to help open data providers and consumers to address the semantic interoperability issue and boost the re-use of open data. The distributed nature of the platform enables different players to collaborate. Developers and SMEs can provide software components that offer services to be integrated in the pipeline of a client. Data analysts and data journalist can set up pipelines to extract information hidden in raw, fragmented data sets. The future work in the development of the Fusepool P3 platform will be focused on the implementation of an authentication service based on WebID

[9] http://openrefine.org/
[10] https://stanbol.apache.org/

and the integration with the CKAN platform. The platform documentation [11][12] and software is available on the Github repository[11].

Acknowledgements

The Fusepool P3 project presented in this paper is partly funded by 7th Framework Programme of the European Commission grant No.609696 and is a collaboration between 8 organizations (Bern University of Applied Sciences, Net7 srl, OpenLink Ltd., Geox Terinformatikai KFT, SpazioDati srl, Salzburg Research Forschungsgeselschaft mbH, Swissdat GmbH, Autonomous Province of Trento, Region of Tuscany).

References

[1] European Commission, Open Data, An engine for innovation, growth and transparent governance, 2011.
[2] Berners-Lee, T., Linked Data, http://www.w3.org/DesignIssues/LinkedData.html, 2006.
[3] Heath, T., Bizer, C., Linked Data, Evolving the Web into a Global Data Space, Morgan&Claypool Publishers, 2011.
[4] Dodds, L., Davis, I., Linked Data Patterns, http://patterns.dataincubator.org/book/, 2012.
[5] Berners-Lee, T., Read-Write Linked Data, http://www.w3.org/DesignIssues/ReadWriteLinkedData.html, 2009.
[6] Linked Data Platform Working Group, Linked Data Platform 1.0, http://www.w3.org/TR/ldp/, 2015.
[7] Allemang, D., Hendler, J., Semantic Web for the Working Ontologist, 2nd Ed., Morgan Kaufmann, 2011.
[8] Fellegi, I.,P., Sunter, A.,B., A Theory for Record Linkage, Journal of the American Statistical Association, 1969.
[9] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontakostas, D., Mendes, P., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., Bizer, C., DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia, 2012.
[10] Volz, J., Bizer, C., Gaedke, M., Kobilarov, G., Silk – A Link Discovery Framework for the Web of Data, 2009.
[11] Gmür, R., The Fusepool P3 Platform, http://fusepool.gitbooks.io/the_fusepool_p3_platform/
[12] Fernandez, S., Frank, J., Westenthaler, R., Blakeley, C. Idehen, K., Gmür, R., Gschwend, A., Fusepool P3 Deliverable 5.1, Technical platform specification and basic implementation.

[11] https://github.com/fusepoolP3