Towards an architecture and adoption process for Linked Data technologies in Open Government contexts *

A case study for the Library of Congress of Chile

Francisco Cifuentes-Silva
WESO Research Group
Department of Computer Science
Universidad de Oviedo
francisco.cifuentes@weso.es

Christian Sifaqui
Biblioteca del Congreso Nacional de Chile
csifaqui@bcn.cl

Jose Emilio Labra-Gayo
WESO Research Group
Department of Computer Science
Universidad de Oviedo
labra@uniovi.es

ABSTRACT

The idea of "Web of data"[?] has been widely enhanced by the establishment of Linked Data principles on the Web [?]. The emergence of Linking Open Data Project [1] opens the doors to the concept of Linked Open Data, establishing the basis for publishing open data, usable by anyone on the Web. However, even though the form (Linked Data) and the goal (Web of data) have been formally defined, the definition of a components architecture to support the implementation of such technologies is still fuzzy, as is a methodology of implementation associated with this architecture. The problem is that the definition and methodology together should enable both publishing and maintenance of semantic data in a standardized way for a subset of publishers, namely public administrators. In this paper we describe a first approach to an adoption process of Web Semantic technologies, specifically with regard to tools and methods that enable the publication and maintenance of Linked Open Data within the context of public administration. To this end, we review the related basic concepts, define infrastructure, describe its components and functions, and propose a sequence to be followed. Finally, we present a case of the use of our methodology in the Library of Congress of Chile.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; I.0 [**Computing Methodologies**]: GENERAL

*(Produces the permission block, and copyright information). For use with SIG-ALTERNATE.CLS. Supported by ACM.

[1]http://esw.w3.org/topic/SweoIG/TaskForces/ CommunityProjects/LinkingOpenData

I-SEMANTICS 2011 2011 Graz, Austria

General Terms

Design, Theory, Standardization

Keywords

Semantic Web, Linked Data, Linked Open Data, Semantic Web Methodology, RDF, SPARQL, Open Government

1. INTRODUCTION AND MOTIVATION

Each year, more and more organizations in the world publish datasets of Linked Data openly, queryable from any point on the Web. It is estimated that there are officially published, as of April 2011, over 35,8 billion triples [2] distributed in approximately 280 datasets all over the world. Making data available openly has multiple justifications, especially in the field of public administration:

- Generate trust, promoting information transparency.
- Facilitate studies and research.
- Open systems facilitate external contributions.
- The public data belongs to the nation, because it is funded by taxes.

Currently there are a large number of tools supporting the adoption of Semantic Web technologies, oriented to the creation, publication and management of data, and a large subset of these tools are focused on our interest object: Linked Open Data. However, an important weakness in this area of web engineering is that a formal framework has been not established to define infrastructure generally in terms of components and the sequence in which this infrastructure must be implanted, always focused on the context of Open Government. This deficiency implies slower technological adoption, in both public sector and private sectors [?]. Although currently general approaches related to publishing and consuming Linked Data [?, ?] exist, our proposal adds aspects related to maintenance, use and adaptation of Linked Data content in the domain of Open Government that have not been covered by other proposals. We present the functions of each component of an architecture that enables the use of Linked Open Data in a public use context. Under this scenario, our main contribution is the description of a base

[2]http://www4.wiwiss.fu-berlin.de/lodcloud/

architecture and an adoption process of Semantic Web technologies that enable a Linked Open Data environment for public administrations.

To this end, we defined an architectural view of the technological components that support the publishing and maintenance of Linked Open Data, and we describe the process needed to implement component infrastructure. In this way, in a first stage we define the architecture and its components; subsequently we describe the sequence of implementation, delivering a reference that may be applied in different contexts related to public administration such as education, employment or health, for instance. Finally, we present a case of application of our adoption process in the legislative context.

For easy understanding, we suppose that the readers have a background of the main concepts about the Semantic Web.

2. PARTICULARITIES OF OUR OPEN GOVERNMENT CONTEXT

For our application context, we have important considerations that have allowed defining our systemic model in terms of components architecture and implantation model. These main particularities are as follows:

1. The organization is the Library of Congress of Chile.
2. Currently it does not have a team of specialists on Semantic Web, although some engineers have had several courses in this area. Current work loads make efforts in this regard very difficult.
3. The organization offers services through multiple web portals, so the integration of a new project should not interfere with ongoing projects, both in development and production environments.
4. The project will be implanted in two phases: the first phase concerns about the technological infrastructure and its implementation, and the second phase concerns the modeling of data. This data will be added progressively, because of the multiple data domains existing in the organization.

Taking into account these considerations, the next step is to define our proposal.

3. LINKED DATA INFRASTRUCTURE

Fig. **??** shows our architectural proposal for the Linked Data implementation. The model is based on the following components:

- **RDF Storage System** [**?**]: component used for the management of RDF triples. The SPARQL Endpoint and the Output RDF Graph will work on this.
- **Cache database**: a database management system optimized for cache, oriented for storing queries and results from the RDF Storage System, achieving improve the response time in complex SPARQL queries that involving reasoning tasks. This component will be used for the SPARQL Endpoint.
- **Documentation Web Portal**: This Web portal has three targets, to hold the ontologies of the domain model, to include the documentation about URIs schema and RDF graph for developers, and finally to be the access point for all the services and resources of the Linked Data infrastructure.

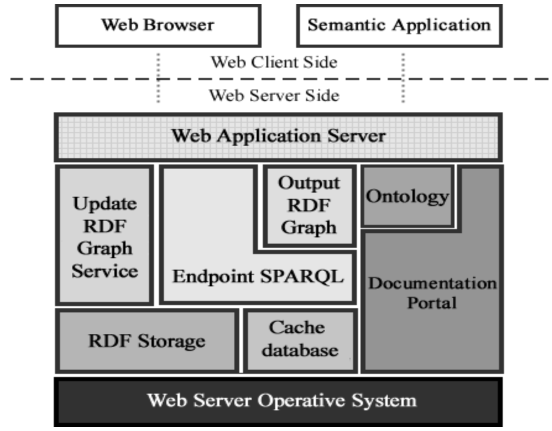

Figure 1: Our proposal of Linked Data architecture.

- **Ontology**: one or a set of ontologies that model the domain over which Linked Data are generated.
- **Output RDF Graph**: an application that generates RDF representations for URIs previously designed, or in simple terms, an application that generates a RDF graph over an HTTP URI. To simplify the model, we consider that for the data collection, this tool makes queries to SPARQL Endpoint, processing and showing results in URIs defined for the graph model.
- **Endpoint SPARQL**: A tool that fulfills the SPROT specification (SPARQL Protocol for RDF) [**?**], which allows executing SPARQL [**?**] queries over a RDF Graph, generating utilizable results for clients. In our architecture, the RDF Storage System and the Cached component are located under this tool, and over this is located the Output RDF Graph.
- **Update RDF Graph Service**: this must update the RDF Storage System from the corporative database. This component will automatically create, update and delete RDF triples based on changes in the corporate database.

4. ADOPTION PROCESS

For the implementation of the architecture that we defined above, we propose an adoption process based on phases defined by Fig. **??**, which shows each one of the phases applied in a temporal dimension.

Now, we will explain each phase of the adoption process.

4.1 Contextualization

This phase should identify the application context and the characteristics of the data to be published. Using a method of describing requirements such as UML notation, three elements should be defined from a high-level systemic perspective:

- **What kind of data will be delivered**: describe the domain model of data to be published, considering referencing external data sources. As an example for our

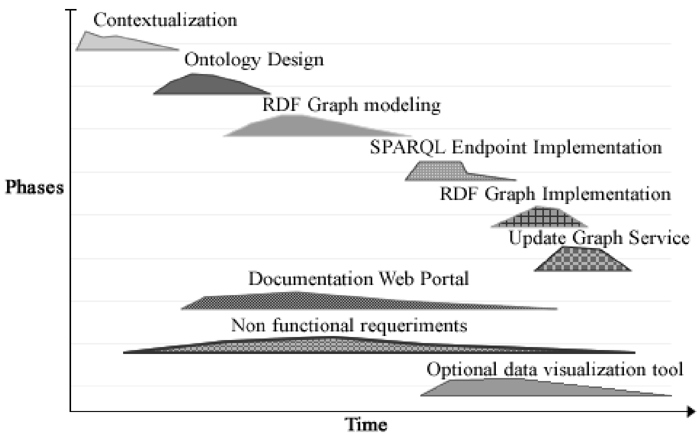

Figure 2: Our proposal of process for implementation of Linked Data.

case study, the data the system will deliver are basically legislative norms. This involves the description of the domain model in the form of classes, attributes and relationships that can be expressed using UML notation.

- **How to deliver the data**: describe access and request models of the data, using open standards such as REST or SOAP. Define output formats to be delivered in RDF data, for example, some output formats are HTML, RDFa, XML, JSON, N3 or YAML.

- **Who will consume the data**: describe who will consume the data. If there are specific agents, open interfaces for own or third party applications, or humans willing to visualize the data.

A first approach of the system will be the result of this stage. This approach will be detailed in a document that should take as its basis the three elements mentioned above.

4.2 Ontology Design

This phase concerns about the definition of vocabularies or ontologies required according to the domain model established in the previous phase. This stage is critical in our semantic system because this model will apply validations, logical rules and inferences about the data. For doing this, we consider two possible alternatives:

1. **Use existing models**: if the domain model we need to use has been already modeled by others as an ontology and it is available for use, simply use this in a total or partial form without extra effort. To find out if the domain model has been modeled before, you can use search engines of semantic material such as Falcons [3], Swoogle[4], Watson[5] or the like.

2. **Design reusable models**: if the domain model we need to model was not modeled before, it will necessary to design an ontology that allows us to model the domain problem. For doing the design, there exist several ontology matching techniques [?, ?, ?] that enable reuse. In this point, note the specific meaning of the principles of "reuse, not reinventing" and "mix freely" as defined in [?] .

[3]http://ws.nju.edu.cn/falcons/
[4]http://swoogle.umbc.edu/
[5]http://kmi-web05.Open.ac.uk/WatsonWUI/

For the implementation of the ontology, the ideal is to use OWL [?], RDF Schema [?] or a mix of them. Take into account that RDF Schema allows less expressiveness that OWL, because RDF Schema can only express classes, properties and hierarchies between these elements, while OWL adds the definition of negations, quantifiers, cardinalities and attributes of properties. Another point to take into account at this phase is the establishment of HTTP URIs and a prefix for the publication of the ontology. We suggest putting all the ontologies under a directory "ontologies". With respect to the prefix, if you are dealing with an organization, ideally this should be related to the name or an acronym that will be unique for web publication. On the other hand, if you are modeling only one part of the domain of the organization, we recommend implementing the ontology in a separated form and referencing it from the root ontology. This way, the prefix of the new part of the organization will be composed with the prefix of the base ontology linked with a prefix of the sub-domain model. By applying this modeling design it is possible to extend the ontology and the application domains through sub-ontologies without limits. Finally, we believe that the ontology should be commented and documented in English language and in the native language of the project. In our case study, the ontology will be implemented in both Spanish as English.

4.3 RDF Graph modeling

In this phase the HTTP URIs of the RDF Graph will be defined. Thus, the first step is to define the URI patterns that will form the RDF Graph. To do so, there are many approaches that support modeling through URI design patterns, such as the book of Davis and Dodds [?], as well as the works by Berners-Lee URIs [?] and Cool URIs [?]. Another recommendation is to hide the implementation details, designing URIs based on abstractions that enable readability [?]. Not only should the RDF Graph take into account the design of the URIs pattern, but also the data description that will be delivered in each URI access. Our approach also takes into account that all the data delivered are matched with the ontology or vocabulary designed. Good advice for defining the data output is to define an URI example describing the output RDF. For this exercise you could use N3 format which is more human readable. Once all URI patterns and RDF outputs are defined, the RDF Graph will be almost finished.

Other important issues to take into account are the negotiation of the content when delivering content and data visibility. In the first case there are methods that enable this feature applied to Linked Data environments [?, ?] . And for data visibility the creation of a semantic site map will help crawlers from search engines [?]. In the robots.txt file there should be a few lines with the SPARQL Endpoint location, the base URI of web site and the Linked Data sources and the location of a data dump if it exists.

Our proposal includes the internationalization of URI patterns when the project contextualization is not restricted to the English language. There are application contexts that require data to be published in the native language. For example, if the government of Spain wants to publish data on the Web about education, it makes more sense to provide access to URIs with the words "escuela" and "profesor" instead "school" or "teacher", mainly because the people in this country do not necessarily speak English. However, if some programmer that does not speak Spanish wants to develop a mash up comparing countries, it makes sense to publish in English because this language is used as "lingua franca" today. Going further, there is another problem related to this, namely the ambiguous expressions when a word is translated without concern for domain. For example, in our case study, there is an entity called "Organismo", that identifies governmental organizations. The mismatch in this case is that the literal translation of "organismo" to English is "organism", which has a more biological sense. On the other hand, translating "Organismo" to "Organization", the sense is wider than necessary, because the English word involves both private and public organizations. To deal with this problem, we propose to translate the static parts of the URI patterns in the graph. This requirement involves maintaining the URIs in the project context language, adding a translation to the static parts of each pattern to the English language, in order to achieve standardized access for all possible data consumers. The last issue in this phase, and related with our case study, is the adoption of the FRBR bibliographic standard [?] of IFLA [6] in the URI design, to define versions and implicit metadata.

4.4 SPARQL Endpoint Implementation

The next phase involves the implementation of the SPARQL Endpoint. Commonly organizations maintain and manage their data in corporate databases giving access to applications. To enable use of this data in the Linked Data context, it will be necessary to perform an extraction, transformation and loading process, also called for short ETL [?] in the Business Intelligence context, or at least a transformation process from the corporate database to RDF data. So, in this phase we observe two main tasks:

- **Extraction, transformation and loading of data**: this task involves taking the data from the organization data environment to the publishing on the Web. We recommend the use of ETL tools such as Kettle [7] used in Business Intelligence context, because is free for use and offers improved features for making the job in an automated way.

- **SPARQL Endpoint implementation**: This task can be divided giving implementation to two elements: a database management system with a SPARQL query defined as input, and a reasoner for the logic operations (if reasoning capabilities are required). For our solution the use of an RDF storage system like the defined in[?], where the reasoning capabilities are implemented in an architecture layer of the component is well suited. In the practice, there exists many implementations for this component such as Sesame [8], Openlink Virtuoso [9], RDFStore [10], 4Store [11], Redland [12], Bigdata [13] or OWLIM [14].

4.5 RDF Graph implementation

In this phase, the component that generates a RDF Graph over the previously designed HTTP URIs must be implemented. Basically, the application must receive an HTTP request in a defined HTTP URI, and must return a RDF data set in some format representation established by one parameter identifying the format. The application communicates with the SPARQL Endpoint, dispatching SPARQL queries for each HTTP request. This application could be developed from scratch completely, but there are many alternatives of free software that implement this functionality (also called Linked Data Front end), such as Pubby [15], Elda [16], djubby [17] or D2R server [18], the latter used in architectures where Linked Data publishing is based on mapping from relational databases.

4.6 Update Graph Service

In this phase a tool that allows maintaining an updated RDF graph over time must be implemented. The main idea is to design an autonomous service capable of adding new data from the corporate database to the RDF storage system. For implementation, one alternative is to compose the tool using the ETL transformations defined previously, but changing the manual execution of the process for automated executions. In technical terms, the tool should incorporate the whole set of instructions needed for creating RDF triples from the corporate database. After this the tool must load these triplets onto the server running the service or in another remote server previously defined. For this goal we propose the use of the Kettle API to execute the transformations that are similar to those defined in the upload processes. These transformations are considered similar, because it is necessary to include criteria for obtaining updated triplets. Once the transformations are finished, the application must connect to the RDF store and load the triplets. The latter can be implemented by using secure shell commands (ssh). Alternatives to ETL usage could be the implementation of external applications based on web services, sockets or other communication methods.

4.7 Documentation Web Portal

In this phase a web portal that allows the publishing of documentation and support for data use is developed. The purpose contents of the portal are defined as follows:

- Ontologies or Vocabularies modeled.

- Documentation guide for the use of the data published, describing datasets and consume methods.

[6] International Federation of Library Associations and Institutions
[7] http://kettle.pentaho.com
[8] http://www.Openrdf.org/
[9] http://virtuoso.Openlinksw.com/
[10] http://rdfstore.sourceforge.net/
[11] http://4store.org/
[12] http://librdf.org/
[13] http://www.systap.com/bigdata.htm
[14] http://www.ontotext.com/owlim
[15] http://www4.wiwiss.fu-berlin.de/pubby/
[16] http://elda.googlecode.com/hg/deliver-elda/src/main/docs/index.html
[17] http://code.google.com/p/djubby/
[18] http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/

- A tutorial or user guide for the SPARQL Endpoint, which includes examples of queries.
- The necessary content that allows access to the entire Linked Data infrastructure.

With regard to the most important features of the web portal, we consider:

- **Internationalization**: if the context language of the project is not English, a main version of the web portal in the native language should exist, as well as an English version, enabling use by international developers.
- **Accessibility**: as the portal is developed in the context of Open Government, we consider that all the conformity levels of accessibility described in WACG [19], with the idea of ensure the access for all the citizens should be met.
- **Based on W3C standards**: The implementation of the Web portal must be in conformity with the W3C standards, ensuring the compatibility and interoperability.

As a final consideration, a useful idea is the implementation of content negotiation for the delivery of documentation.

4.8 Non functional requirements

The Linked Data support infrastructure implies the implementation of some non-functional requirements that ensure some advantages in production. For instance, we consider cache management relevant. This feature will allow improving response times in SPARQL queries, especially when queries imply reasoning capabilities or high data volume. In our architecture, this feature is considered as a cache management between the RDF storage system and the output RDF graph. Another very important non-functional requirement is security. For instance, the Update Graph Service should transfer the news triples from the corporate database to the RDF storage system in a secure way over the Internet. In this case, it must not be possible to change the data if the data stream were intercepted.

4.9 Optional Data visualization tool

Possibly the most attractive applications by using data in Linked Data format, are the visualizations or mashups, specially in Web 2.0[?] contexts. Whereas under this context is possible to interoperate with diverse sources, without need of consuming data through APIs or transformations from specific formats, it is a good idea to design a mashup or visualization that represents data in an attractive form, and at the same time, to give more attraction to the project. For the implementation of Linked Data visualization tools, there are free alternatives like Linked Data Browsers [?], Faceted Navigation Tools and Mashups. Some examples of this type of tools are Tabulator[20], Marbles [21], FOAFNaut [22], DERI Pipes [23], Openlink Data Explorer [24], Disco [25], Zitgist [26] or gFacet [27].

Other choice is the design of our own Mashup. In this sense, we propose some guidelines to be taken into account in the design:

- **Queries Option**: In our opinion, one of the more valuable features in a visualization tool is the possibility of obtaining unique results. In this sense, it is a good idea to have a small set of previously calculated data, but offering as the main focus the possibility of making queries that generate new information, in real time and from various sources.
- **Focus on usability**: the tool must be easily understandable, both the controls with which users interact and the results that are obtained.
- **Visual impact**: the more attractive the tool, the more it will be used. In this aspect, it is very important to take care with the user interface, both in the graphic design as well as in the interaction.
- **Use several data sources**: one of the big eye-catching features of Linked Data visualization tools is the possibility of use and join information from different data sources, in line with Linked Data principles.
- **Functional over the Web**: if the tool uses Linked Open Data, it must ideally be available to every user in the world.

5. CASE STUDY

5.1 Project fundamentals

As we previously mentioned, our case study is in the legislative context. We will define "legislation" to be the set of norms (acts, laws, decrees) that frame the national juridical order. Legislation is information that comes from the public sector; it is generated by public organizations and financed by public resources. This information is by nature of public interest, because it is about general interest topics and in particular affects the citizens lives. Also this information has high public value, because it generates citizen experience that can be considered valuable. After saying that, we realize that legislation is indeed public information and must be of public domain.

Furthermore, legislation must meet the "fulfillment of legal certainty" concept, which is the founded expectation that citizens have about in-force legislation that must be met. However, to meet these two requirements, two further issues related to security must be addressed: "juridical safety", i.e. the guaranteed and founded certainty that the norms will be accomplished, and the "juridical certainty", i.e., the perception of the certainty of the content of the norm. To satisfy these demands, each nation has a mechanism for the publication of the legislation, known as the Official Gazette.

In the particular case of Chile, we will show three articles of the Civil Code [?] that deal with this. The articles are as follows:

- Art. 7° "The publication of the law will be made by its insertion in the Official Gazette, and from that date it will be known by everybody, and it will be mandatory".

[19] http://www.w3.org/TR/WCAG10/
[20] http://www.w3.org/2005/ajar/tab
[21] http://marbles.sourceforge.net/
[22] http://crschmidt.net/semweb/foafnaut/
[23] http://pipes.deri.org/
[24] http://ode.Openlinksw.com/
[25] http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco/
[26] http://zitgist.com/
[27] http://code.google.com/p/gfacet/

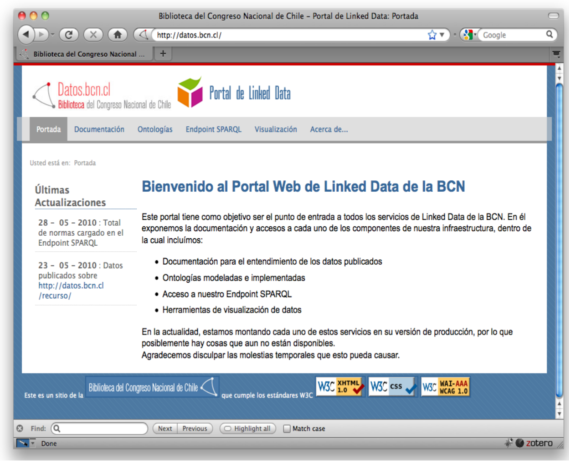

Figure 3: Linked Data Web portal http://datos.bcn.cl

- Art. 8° "Nobody will be able to plead ignorance of the law after it has entered into force".
- Art. 706° "...the error of law is a presumption of bad faith, that does not admit evidence to the contrary".

These articles imply what we call a "legal fiction of knowledge", because in Chile the Official Gazette is not free; the publishing of the laws is basically about modifying norms, and not about texts in force; and access to old legislation is very difficult.

The Library of Congress of Chile (Biblioteca del Congreso Nacional - BCN), released in 2008 the system LeyChile [28], a database about legislation which contains the whole text of the legal norms, their versions (available from 1998), as well as their amendments (modifications, regulations, recasted texts and concordances). This system offers a search box of legal norms, and also web services that offer the norms in XML and other complementary services and applications such as widgets and to facilitate the consumption and use of the legal information stored in the database. The first objective of this system was to bring a solution to the "juridical certainty" for the National Congress of Chile and also citizens in general, especially with regard to in-force texts. Today, this system has on average 14.000 daily visits, reaching daily peaks of 18.000 visits, which is very high considering the content and the 7,3 million Internet users in Chile [**?**]. The BCN, aware of the public value that its database brings to the citizens, has been including new services for access the norms [29], and in this way a natural extension has been to offer access as Linked Open Data for LeyChile. We believe that by offering this new service layer, the whole system will be a reference model in terms of entities and rules, that will be published through ontologies, including he possibility to perform complex queries on the information of the legal norms through the SPARQL endpoint, allowing in this way access to results in more outputs format, such as RDF/XML, JSON, HTML+RDFa or N3.

5.2 Solution developed

In practice and from the technical perspective, the case study is being implemented in its final stage as a first initiative by the BCN in relation to the publication of linked data and it has been developed by using our proposal both for architecture and the adoption process. In a first stage, we developed the contextualization phase about a very bounded domain: norms and their amendments, always within the legislative context. For these, we have written a contextualization document, in which the three main elements of context are described: the data that will be delivered, the form of delivery, and who will consume it. In basic terms, the data to be delivered are legal norms (acts, laws, decrees) and their relationships, without considering in a first stage the internal structure of a norm. The form of data delivery is through two mechanisms: a SPARQL Endpoint and a Output RDF Graph over HTTP. Finally, the consumers of the data are, in first instance, visualization applications of the Library of Congress of Chile and other third-party applications.

Then, we defined an ontology [30] of legal norms and a namespace prefix for the ontology, which is related with the particular context of the national reality. We considered a structure extensible to others domains such as parliament, education, health and others. This ontology has been written using both RDF Schema and OWL, making possible the application of inferences to RDF graph. Another important feature of the ontology, is that it has been composed using previous ontologies and datasets such as SKOS[31], Dublin Core[32], FOAF[33], Geonames[34], Organization[35] and DBPedia[36]. By using the last two of these, we were able to link data from the legal norm graph to external record sets, specifically with international treaties and countries. This task was not trivial because it requires intense manual labor. Finally, this ontology was stored in the RDF store in order to allow inferences such as those already published in the web by using text files in RDF/XML and N3 syntax and its documentation was published in Spanish as also in English.

Once we had modeled the ontology, we modeled the output RDF graph. In practice, we defined an URI schema with all the possible URI patterns that could be consulted in a valid way. To build this model, we took into account the use of the IFLA FRBR standard as an URI for the legal norms. In general terms, the graph follows an hierarchical schema on each one of the resources available for consultation. On the other hand, we also modeled some queries (e.g., obtaining legal norms for specific dates). Then for each URI pattern we defined an RDF output using N3 syntax. Finally, at this stage the output formats for the resources were established. For this project, RDF/XML, JSON, Ntriples, N3 and HTML+RDFa were defined.

Subsequently, we generated the process for transformations and data loading. To this end, we built a Java update service by using the Kettle API, for the loading process, the updating process and the transformation process. Thus, by using the ETL designer, we implemented the different trans-

[28] http://www.leychile.cl
[29] http://www.leychile.cl/Consulta/legislacion_abierta
[30] http://datos.bcn.cl/ontologies/bcn-norms/doc/
[31] http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20050510/
[32] http://dublincore.org/
[33] http://xmlns.com/foaf/0.1/#sec-status
[34] http://www.geonames.org/ontology/documentation.html
[35] http://www.epimorphics.com/public/vocabulary/org.html
[36] http://wiki.dbpedia.org/Ontology

URI Pattern

```
cl/{type}/{organization}/{publish_date}/{number}/data.{ext}
```

RDF output example

```
<http://uri_a_norma1/> rdf:type
bcn-norms:RootNorm; bcn-norms:type <http://uri_al_tipo>;
    dc:title "norm  1";
    bcn-norms:publishDate "1982-10-17";
    bcn-norms:hasNumber "11111";
    bcn-norms:hasVersion <http://uri_a_version_1_norma1>;
    bcn-norms:hasVersion <http://uri_a_version_2_norma1>;
    ...
    bcn-norms:hasVersion <http://uri_a_version_n_norma1> .
<http://uri_a_version1_norma1> bcn-norms:isLatestVersion "true" .
```

Figure 4: Example of URI pattern and RDF output in Notation 3

formations that generate the RDF triples in N3 syntax, for both initial data load and for the triplet updates (usually it will only add new ones). Under these conditions, the update service runs transformations and then loads the triples in the RDF store.

In the next stage, the output RDF graph over HTTP has been implemented according the model designed for the purpose. For implementation we used the WESO DESH tool, a linked data front end, soon to be released as free software. This tool implements, among others, content negotiation, cache, URI internationalization and the ability to make SPARQL queries (DESCRIBE, ASK and CONSTRUCT) to different SPARQL endpoints. Additionally, this tool allows an administration base on an user interface and it can be integrated with any SPARQL endpoint. Finally, this linked data implementation has been validated with the linked data validators such as Vapour and RDF/XML from the CTIC Foundation and W3C respectively.

The documentation Web Portal shown in Fig. **??** is already running and we are adding more documentation every day for use of the Linked Data infrastructure both in English and Spanish. This portal has been implemented using TYPO3 CMS because it provides scalable architecture, internationalization in a native way, template management and one important community contributing with support and additional application modules (extensions) among others.

A highlight of this project is that it was build entirely with free software, so a similar project could be fully replicated without paying software licenses. The project is currently finishing, and now is operating under the URL http://datos.bcn.cl, thus making available a realistic use-case of our methodological proposal in each one of our view points, architecture and adoption process. Each one of these points is in line both with the requirements of the Library of Congress of Chile, with our proposal and with related specialized literature. One interesting thing we could identify through this project is that given the nature of the data, we often found typos, so we took special care in the model design to allow in the future to manually edit the data. As an example of this we could mention the GovernmentalOrganization instance defined on the ontology. Due to different names for the same resource, different classes were defined.

6. RELATED WORK

Some works that have been our fundamental base are mentioned as follows: Berners-Lee [**?**] gives a justification about the advantages by putting data government as Linked Data. Additionally, this work gives ideas and recommendations on how to dispose Linked Data in an open way. The stages related to publishing Linked Data are described in a general way [**?**], covering main tasks that include negotiation of content, ontologies and URIs design, and contextualization. However, this approach does not consider the ETL process, the updating graph service and the topics of documentation and internationalization depending on context. A detailed introduction to technical aspects related mainly to publishing and consuming Linked Data is presented in [**?**]. Through a use scenario, they present an overview of methods, frameworks and good practices related to the implementation of Linked Data in a generic context. However, we added the definition of an architecture and an adoption process that enable the incorporation of components of the architecture in a general context. In [**?**] a case of study with the scope of RDF graph modeling which is related to our work is presented. In this paper, the adoption of the SKOS ontology, the concepts of content negotiation and other related technical aspects are described. Finally, there have been important references for our work, the documentation of the implementations released by The Library of Congress of the United States [37] and the related projects in United Kingdom such as the Opening Up Government project [38] , and in Spain such as the Spanish Association of Linked Data [39].

7. CONCLUSIONS

Our work defines the architecture components and implementation process for Linked Data within the context of Open Government. Through an in-house use-case, we verified that our approach provides a general solution to practical problems, by providing an architectural definition and framework for use when Linked Data initiative implementations, especially in the area of public administration where Semantic Web initiatives are in the adoption process around the world. We propose that our proposal for architecture and adoption processes can provide clear definitions to support project implementation and to resolve related technical issues. Another important element that we consider essential in this type of project involves internationalization and accessibility issues, mainly because we are speaking about Open Government, in the sense that access to the entire population should be guaranteed.

Finally, we found that our use-case validated our initial proposal, keeping always in mind that our work was focused in Open Government. Our work coincides with other approaches discussed in the specialized literature, as it also adapts, structures and delimits them in light of our specific goals.

For the future, we aim to continue improving and extending our proposal, by adding components to the architecture and adoption process in order to make our solution available in other contexts. We think that is possible to generate a macro structure both in terms of architecture and the adop-

[37]http://id.loc.gov/authorities/about.html
[38]http://Data.gov.uk/Linked-Data
[39]http://www.aelid.es/

tion process so as to provide a generic solution that can be applied in a variety of diverse contexts.

Concerning new data sets, we will be adding new ontologies, such as the history of MPs (already available as RDFa [40]) and transparency [41].

8. REFERENCES

[1] Código civil chileno, DFL-1 30-MAY-2000 MINISTERIO DE JUSTICIA. http://www.leychile.cl/Navegar?idNorma=172986, May 2000.

[2] A. Ankolekar, M. Krötzsch, T. Tran, and D. Vrandecic. The two cultures: Mashing up web 2.0 and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(1):70 – 75, 2008. Semantic Web and Web 2.0.

[3] T. Berners-Lee. Univeral resource identifiers – axioms of web architecture. http://www.w3.org/DesignIssues/Axioms.html, 1996.

[4] T. Berners-Lee. Cool uris don't change. http://www.w3.org/Provider/Style/URI.html, 1998.

[5] T. Berners-Lee. Linked data - design issues. http://www.w3.org/DesignIssues/LinkedData.html, July 2006.

[6] T. Berners-Lee. Putting government data online - design issues. http://www.w3.org/DesignIssues/GovData.html, June 2009.

[7] C. Bizer, R. Cyganiak, and T. Heath. How to publish Linked Data on the Web. http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/, Oct. 2008.

[8] C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *International Journal on Semantic Web and Information Systems*, pages 1–26, Mar 2010.

[9] G. Coskun, O. Streibel, A. Paschke, R. Schäfermeier, R. Heese, M. Luczak-Rösch, and R. Oldakowski. Towards a corporate semantic web. In *International Conference on Semantic Systems (I-SEMANTICS '09)*, Graz, Austria, 09/2009 2009.

[10] I. Davis and L. Dodds. Linked data patterns. http://patterns.dataincubator.org/book/, 2010.

[11] A. Fosk. Latin america's internet population grows 15 percent in past year to 112 million people - comScore, inc. http://www.comscore.com/Press_Events/Press_Releases/2011/3/Latin_America_s_Internet_Population_Grows_15_Percent_in_Past_Year_to_112_Million_People, Mar. 2011.

[12] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. Semantic schema matching. In R. Meersman and Z. Tari, editors, *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*, volume 3760 of *Lecture Notes in Computer Science*, pages 347–365. Springer Berlin / Heidelberg, 2005.

[13] K. Grant, C. Lee, Feigenbaum, and E. Torres. SPARQL protocol for RDF. http://www.w3.org/TR/2008/REC-rdf-sparql-protocol-20080115/, 2008.

[14] T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, Jan 2011.

[15] A. Hertel, J. Broekstra, and H. Stuckenschmidt. *RDF Storage and Retrieval Systems*. 2008.

[16] IFLA Study Group on the Functional Requirements for Bibliographic Records. *Functional Requirements for Bibliographic Records: Final Report*, volume 19 of *UBCIM Publications-New Series*. K.G.Saur, München, 1998.

[17] Y. R. Jean-Mary, E. P. Shironoshita, and M. R. Kabuka. Ontology matching with semantic verification. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):235 – 251, 2009. The Web of Data.

[18] Y. KALFOGLOU and M. SCHORLEMMER. Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(01):1–31, 2003.

[19] R. Kimball and J. Caserta. *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. Wiley, Indianapolis, IN, 2004.

[20] D. L. McGuinness and F. van Harmelen. OWL web ontology language overview. http://www.w3.org/TR/owl-features/, 2004.

[21] A. Polleres and D. Huynh. Special issue: The web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):135 – 135, 2009. The Web of Data.

[22] E. Prud'hommeaux and A. Seaborne. SPARQL query language for RDF. http://www.w3.org/TR/rdf-sparql-query/, 2008.

[23] W. RDF Working Group. RDF - semantic web standards. http://www.w3.org/RDF/, 2004.

[24] J. Sequeda, O. Hartig, P. Sinclair, and J. Taylor. ISWC 2009 Tutorials/How to consume linked data on the web - iswc2009. http://iswc2009.semanticweb.org/wiki/index.php/ISWC_2009_Tutorials/How_to_Consume_Linked_Data_on_the_Web, Oct. 2009.

[25] E. Summers, A. Isaac, C. Redding, and D. Krech. Lcsh, skos and linked data. In *Proceedings of the 2008 International Conference on Dublin Core and Metadata Applications*, pages 25–33. Dublin Core Metadata Initiative, 2008.

[40] http://biografias.bcn.cl
[41] http://transparencia.bcn.cl