Guidelines for Linked Data Generation and Publication: an Example in Building Energy Consumption

Filip Radulovic*, María Poveda-Villalón, Daniel Vila-Suero, Víctor Rodríguez-Doncel, Raúl García-Castro

Ontology Engineering Group, ETSI Informáticos, Universidad Politécnica de Madrid

Abstract

Linked Data is the key paradigm of the Semantic Web, a new generation of the World Wide Web that promises to bring meaning (semantics) to data. A large number of both public and private organizations have published their data following the Linked Data principles, or have done so with data from other organizations. To this extent, since the generation and publication of Linked Data are intensive engineering processes that require high attention in order to achieve high quality, and since experience has shown that existing general guidelines are not always sufficient to be applied to every domain, this paper presents a set of guidelines for generating and publishing Linked Data in the context of energy consumption in buildings (one aspect of Building Information Models). These guidelines offer a comprehensive description of the tasks to perform, including a list of steps, tools that help in achieving the task, various alternatives for performing the task, and best practices and recommendations. Furthermore, this paper presents a complete example on the generation and publication of Linked Data about energy consumption in buildings, following the presented guidelines, in which the energy consumption data of council sites (e.g., buildings and lights) belonging to the Leeds City Council jurisdiction have been generated and published as Linked Data.

Keywords: Linked Data, guidelines, buildings, energy consumption

*Corresponding author. Tel: +34 913363670, Fax: +34 913524819

Email addresses: fradulovic@fi.upm.es (Filip Radulovic), mpoveda@fi.upm.es (María Poveda-Villalón), dvila@fi.upm.es (Daniel Vila-Suero), vrodriguez@fi.upm.es (Víctor Rodríguez-Doncel), rgarcia@fi.upm.es (Raúl García-Castro)

1. Introduction

Last years have witnessed the growing interest of many practitioners in publishing semantic data on the Web, mainly powered by the Linked Data[1] initiative, the key paradigm in the next generation of the World Wide Web called the Semantic Web [1]. The concept of Linked Data comes from the idea of using the Web to connect data and aims at transforming the Web into a global knowledge base. The key concept in Linked Data are links between data from different data sets, which ensure that data sets are not just isolated data islands and support data integration.

By describing the concepts in a domain and the relationships between them, ontologies are formal representations of knowledge about a certain domain and are the cornerstone of the Linked Data initiative since they are the formal models for representing data on the Web. Ontologies contain different components (e.g., classes, properties, instances and axioms), and can be implemented in various languages, being the most widely used and accepted language the one standardized by the W3C, the Web Ontology Language (OWL) [2].

The basic principles for developing Linked Data are the following[2]: i) to provide URIs for each entity to be represented; ii) to provide HTTP URIs for those entities; iii) to use web standards such as RDF [3] for describing data; and iv) to include links to resources already available in the Web. In addition to these principles, in order to realize the notion of Linked Data, not only data must be available in a standard format (i.e., RDF), but also concepts and relationships among data sets must be defined by means of ontologies.

A significant number of energy-related companies posses data about energy consumption, which is one aspect of Building Information Modelling (BIM), that are represented in different formats (e.g., SQL, CSV or XLS), have different update frequencies, and are accessed through different means (e.g., web services or files). Furthermore, having in mind that these data belong to private companies, legal aspects such as licensing are of high importance [4].

The technologies and principles underlying Linked Data are successfully applied in various domains in order to enhance interoperability among systems, and are starting to be be applied to the architecture, engineering and

[1] http://www.w3.org/standards/semanticweb/data

[2] Adapted from http://www.w3.org/DesignIssues/LinkedData.html

construction (AEC) and BIM domains. These opportunities are being analyzed, discussed and promoted along different initiatives such as, for example, the LDAC workshop series[3].

Linked Data generation and publication are intensive engineering processes that demand high attention in order to achieve high quality and, because of this, some general guidelines and best practices have been developed to this date. However, as experience has shown, generic guidelines are often not sufficient to be applied in every domain. In order to overcome this problem, more domain-specific guidelines have been developed with the aim of addressing particular characteristics and of providing concrete domain-specific examples that help practitioners and Linked Data adopters to better understand and use these guidelines.

This paper aims at guiding through the process of developing Linked Data related to energy consumption in buildings, including the process of transforming the data available in any format into Linked Data and its publication according to the Linked Data principles. To this end, it provides a methodology for generating and publishing Linked Data with advice on design decisions. In this paper, we describe each task of such methodology in detail through several important aspects, which include a detailed description and the steps to be performed within the task. Furthermore, where possible, we give a list of tools that help in performing the task or some parts of the task, different alternatives to perform the task, or we outline the best practices and recommendations that help in achieving a better quality in the task outputs.

The paper also presents an instantiation of the Linked Data generation and publication methodology through the transformation into Linked Data of a data set about building energy consumption. The selected data set comes from the Leeds City Council Open Data office and includes data about electricity, gas and oil consumption from various council sites (e.g., buildings and lights) belonging to the Leeds City Council jurisdiction.

This paper is organized as follows. Section 2 expounds related research efforts. Sections 3 and 4 present the Linked Data generation and publication processes respectively, together with the Leeds City Council example. Finally, Section 5 gives some concluding remarks and discusses lessons learnt and future lines of work.

[3]2014 edition http://linkedbuildingdata.net/events/ldac2014/

2. Related work

Different works have explored the advantages and potential of using the Linked Data approach for integrating and enriching AEC data as exposed by Pauwels and colleagues [5], Abanda and colleagues [6], Madrazo and Costa [7] or, more recently, by Törmä [8]. Other related work, such as the work by Törmä and colleagues exposed in [9], already points out specific research problems in this area (e.g., link-type modelling and link generation). In this section we review existing works regarding the generation of Linked Data in the AEC field and also existing general and cross-domain literature about Linked Data generation and publication.

Regarding the main topic addressed in this paper, that is, Linked Data generation and publication of AEC data, we can mention the work carried out by Pauwels and Van Deursen [10] to transform BIM data based on the IFC standard [11] into RDF. In this case, the authors reported following no particular methodology and only the development of ad-hoc wrappers is mentioned.

Other approaches focus on consuming and integrating existing Linked Data data sets with AEC data in order to overcome interoperability issues, such as the work described by Curry et al. [12, 13] and by O'Donnel et al. [14], or also acting as consumers and publishers of data as the reegle data portal[4] [15] in the energy field.

On the Linked Data side, we should mention key publications such as Heath and Bizer's book for driving the Linked Data generation and publication process [16] and consequent works such as the outcomes of the LOD2 project [17]. These represent the starting point for following the process of contributing to the Linked Data initiative; however, some resources might be specialized depending on the field at hand, as has happened in other areas (e.g., biology or cultural heritage) where domain experts together with Linked Data developers have accommodated tools, techniques, and guidelines to their specific requirements.

Linked Data related to buildings is in its infancy and, since experience unveils that practices are too general and not enough to be directly applied to every single domain, it still needs methodological guidelines supporting its evolution towards a mature and repeatable process and providing clear examples.

[4] http://data.reegle.info/

3. Linked Data Generation Process

This section presents the guidelines for the generation of Linked Data for some existing data by describing the different tasks to be performed in the process.

3.1. Select Data Source

The first step of the Linked Data generation process is the selection of the data source that will be transformed into Linked Data. Such data source is usually owned by the organization and is selected depending on the specific needs of the organization or the expected value to be obtained. Alternatively, an organization may be interested in extending its data with data from other sources not owned by the organization.

This task is achieved by first defining the requirements for the selection of the data source and, then, by selecting one or several data sources that satisfy those requirements.

LCC example. We specified several requirements for the example data: i) to include data about energy consumption in buildings; ii) to have a clear license stated and to be available for use; iii) to be represented in some machine-processable format (e.g., Excel, CSV, XML); iv) to be easily linked with generic real-world entities; and v) to come from some real scenario.

After searching, we decided to use the Leeds City Council (LCC) data set on electricity, gas and oil consumption[5] of a number of council sites in Leeds, a metropolitan district in the United Kingdom. This particular data set was selected because it was the one that complied most with the requirements.

3.2. Obtain Access to Data Source

A data source that is already owned by the organization is easier to access and in most cases such data can be accessed without obstacles, while external data sources can be accessed in a straightforward way only when they are in the public domain. However, not all data sources are in the public domain and some of them are not accessible; in those cases, it is necessary to first obtain access to the data source.

To this end, it is necessary first to identify the person to contact in order to request access to the data source and then to request the access to it. After the access is obtained, the data can be retrieved from the data source.

[5] http://data.gov.uk/dataset/council-energy-consumption/

Provided the user has the required credentials, data can be retrieved through: file or files containing the data, a programming interface (e.g., an API or a web service), a database, or a stream of data (e.g., a sensor network, a social network feed).

LCC example. The data source of the LCC data set is available in the public domain from the *data.gov.uk* web page and is provided in the CSV format.

3.3. Analyse Licensing of the Data Source

Licenses declared for a data set specify the legal terms under which a data set can be exploited.

Therefore, in order to prevent legal conflicts, it is necessary to determine who is the rightsholder and which licenses have been declared for the data. In practice, this might not be an easy task, since one data set can be offered through different sources and have different licences associated, and since there are no legal prescriptions nor standard practices on how to declare the license.

The first step to perform in order to obtain and analyse the licence is to identify the authoritative data set publisher. Prior knowledge of who is the rightsholder is essential to assess if that data has been published by (or in behalf of) the rightsholder or by an authorized distributor.

The second step is to find the applicable license, which can be performed by: i) browsing the web page hosting the data, since typically licensing information is provided as a text in the HTML footer (possibly in a separated page), as a well-known icon (e.g., Creative Commons), or as a combination of both; ii) browsing the data set metadata, for example for RDF data looking in the VoiD/DCAT description for structured information (DublinCore *license*, DublinCore *rights*, or XHTML *license* are the most common licensing elements); iii) inspecting the data set, since licensing information is sometimes present within the data; and iv) contacting the data set publisher if the above steps have not proven sufficient, or if doubts exist about the applicable license.

Finally, the third step is to read the license and to determine if the terms are satisfactory. The analysis of the licenses of a data source should be performed upon all the available copies and formats of the data. Furthermore, all analyses should be performed by the same person or group of persons. In the case when data are to be integrated within a larger data set, it should

be ensured that licenses are compatible and their terms are not mutually exclusive.

LCC example. The data set publisher is *data.gov.uk* and the rightsholder is the Leeds City Council, as can be directly observed on the data set web page. Furthermore, the web page states that the license for the data set is the Open Government License[6], which grants permission of copying and publishing of the data and, therefore, they can be freely used.

3.4. Analyse Data Source

When the license of the data source permits its further use, the next step is to analyse the data source in order to get insight into the data in it and into how such data are structured and organized.

The first step is to analyse the data in order to observe the characteristics of the data, such as quantities, value ranges, etc. Data can be more or less structured; the more unstructured the data are the harder their use is.

The second step is to obtain the schema of the data, identifying the domain concepts that are described in the data set, together with all the relevant relationships between them. In some cases, the schema already exists and can be completed with the results of data analysis. If the schema is not available, it has to be extracted directly from the data.

LCC example. As mentioned before, the LCC data set is available as a CSV file containing electricity, gas and oil consumption data for a number of council sites (e.g., buildings, lights, parks) in Leeds.

For each council site, the data set contains the unique property reference number and the site name and full address (street, place, and postal code). Since the data source does not contain precise information about location types, the beginning and end of time intervals, and units of measurement, we contacted the Leeds City Council Open Data office in order to obtain the required information and to complete the schema of the data than can be observed in the first row of the CSV file.

3.5. Define Resource Naming Strategy

Since the principles of Linked Data already state that URIs must be used for naming resources, the next step is to define the strategy to define the URIs to be used to name the generated resources.

[6]http://www.nationalarchives.gov.uk/doc/open-government-licence/version/2/

There are two basic forms of URIs. One form is the hash URI, in which a URI contains a fragment that is separated from the rest of the URI by a hash character ('#'). An example of this type of URI for an energy company could be *http://www.energycompany.com/about#energyCompany*. In the case of hash URIs, access is always provided to data as a whole and the fragment part has to be stripped off when the URI is requested from the server. Because of this, a hash URI does not necessarily identify an information resource and cannot be retrieved directly; however, hash URIs can be used to identify non-information resources.

Slash ('/') URIs imply a 303 redirection to the location of a document that represents the resource and content negotiation. In this case, resources can be accessed individually or in groups. An example of this type of URI for an energy company could be *http://www.energycompany.com/about/energyCompany*. Drawbacks of slash URIs include HTTP roundtrips, redirects and the need for web server configuration.

When designing URIs, it is advisable to consult well-established guidelines, such as Cool URIs [18], design guidelines for the UK public sector [19], ten rules for persistent URIs [20], or Linked Data patterns [21].

The first step to develop a resource naming strategy is to choose a URI form (hash or slash). In the case of choosing slash URIs, it is also needed to choose one of the two specified content negotiation alternatives[7].

The second step is to choose a domain and a path for the URIs which form the base URI. Finally, the third step is to choose a pattern for ontology classes and properties in the ontology, as well as for individuals.

Unambiguity is of high importance for this task, and one URI should identify only one item. Furthermore, URIs should be persistent and should not contain anything that can change (e.g., state information). One possible way to achieve this is to use a domain that is under direct control of the organization generating the data or to use persistent uniform resource locator (PURL), which is a service for resource management and redirection settings.

When defining resource URIs, it is advisable to separate the ontology model from its instances. To this extent, the string "ontology" and the ontology name should be appended to the base URI in the case of an ontology model, and the string "resource" and the ontology class name should be appended to the base URI in the case of instances. Finally, URIs should be

[7]http://www.w3.org/TR/cooluris/#choosing/

defined in a readable manner so that people are able to understand them.

LCC example. According to the tips provided in [18], since our data set will contain a significant volume of data, and since it can grow in the future with the availability of more data, slash URIs with forwarding to one generic document will be used. In the case of the ontology, since it is rather small, hash URIs will be used. The URI domain to be used is *http://smartcity.linkeddata.es/*, which has been chosen because it is under our direct control, and the base URI to be used for the ontology model and the data is *http://smartcity.linkeddata.es/lcc/*.

Following the tips, the classes in the ontology will have the path form */ontology/<ontologyName>#<className>*. Similarly, the properties in the ontology will have the form */ontology/<ontologyName>#<propertyName>*. Finally, the instances in the ontology will have the form */resource/<className> /<identifier>*.

3.6. Develop Ontology

The ontology is developed through several consecutive steps [22]. The first step is to define the requirements that have to be fulfilled by the ontology [23]. These requirements can be related to the purpose of usage of the ontology, to the domain that the ontology is covering, or to technical details of the ontology, among others.

The second step is to extract the terms from the data schema and from the data, where basic concepts and the relationships between those concepts are extracted. These extracted terms should consist of not only the terms from the data source, but also of synonyms of those terms.

The third step is to define the ontology conceptualization by defining a simple model with the main concepts of the ontology and the relationships between them.

Since reusability is one of the main principles to follow when developing ontologies, the fourth step is to search for existing ontologies that best fit the previously-extracted terms (and their synonyms).

In those cases when widely-used ontologies are already known and can be reused with certain classes or properties, terms from these ontologies can be selected for reuse and there is no need to search for other ontologies.

The fifth step is to select for reuse [23] ontologies and/or ontology elements found in the previous step in such a way that: i) the semantics of the class or property in the ontology is related to the term; ii) if the term relates to a class, the class in the ontology has as much properties that correlate to

the term as possible; iii) the ontology that describes the class or property related to the search term is widely accepted and used.

The sixth step is to implement the ontology according to an ontology implementation language and following the resource naming strategy; this step is performed through ontology integration and ontology completion. The integration of concepts from the selected ontologies into an initial implementation could be done either by importing the ontology to be reused into the ontology being developed or by referring to element URIs so that only those element references are included in the ontology being built [24]. If existing ontologies do not provide all the information needed to represent the data, it is necessary to complete the ontology by introducing new classes and properties that are related to the terms. In this case, it is advisable to expand abbreviations and acronyms and to follow common lexical conventions, such as those presented in [25].

The seventh step is to evaluate the ontology [26]. For doing so, several dimensions for ontology evaluation could be taken into account [27] (e.g., logical consistency, modelling issues, human understanding, ontology language conformance). In order to carry out this activity, it is advisable to use online ontology evaluation services, reasoners and syntax validators.

For searching existing ontologies, the smart city ontology catalogue, Linked Open Vocabularies and search engines (e.g., Google) can be used.

LCC example. For the LCC example ontology, several requirements were specified: i) the ontology should adopt concepts and design patterns in other ontologies where possible and ii) the ontology should be implemented in OWL 2 DL [2].

As the schema of the LCC example is already available within the CSV file, it was used (together with available data) as a reference for the terms and their synonyms, presented between brackets. These terms include: unique property reference number, council site (public building, public structure), suburb, metropolitan district, address, street, postal code, consumption (utilization), utility (energy), identifier, date, time, value. Furthermore, since the name of each council site reveals its type, we have extracted the terms representing those types (e.g., library, museum, park, countryside).

Based on the previously extracted terms, we have defined the ontology conceptualization. Due to space restrictions the final model will be shown below instead of the conceptualization.

In order to search for existing ontologies that describe the extracted terms,

we have used Linked Open Vocabularies[8], Google, and the smart cities ontology catalogue[9]. Several ontologies were found: the schema.org ontology provides a class for describing public sites, which can be used for council site concept, and some additional classes and properties that can be used for this concept (e.g., *PostalAddress*, *CityHall* and *Park*, among others); because of this, this ontology was selected for reuse. The concept of metropolitan district was found in the Ordnance Survey ontology so it has also been selected for reuse. Furthermore, this ontology also provides a concept for describing places (i.e., *NamedPlace*).

In order to capture energy consumption, we have reused the Semantic Sensor Network (SSN) ontology. The key class in this ontology is the *ssn:Observation* class. The one-year time intervals of the observation are represented with the *time:Interval* class from the W3C's Time ontology, while the observed value of the consumption is modelled with the *ssn:SensorOutput* and *ssn:ObservationValue* classes. To capture the specific indicator for which the consumption is related to, the *ssn:Property* class and the *ero:FinalEnergy* class from the Energy Resource Ontology have been used. Measurement units are captured with the *om:Unit_of_measure* class from the Units of Measure ontology.

The ontology developed for the LCC example has been implemented in OWL using Protégé[10] as the ontology editor. The final implementation of the ontology is shown on Figure 1. Due to space reasons, the first level of the *schema:CivicStructure* hierarchy is not complete and the second level is not shown on the figure.

The integration of the reused elements has been done by referencing such terms, instead of by importing the reused ontologies as a whole. For example, the class *ssn:FeatureOfInterest* has been included in the ontology and extended by means of the *schema:CivicStructure* class.

Since the search for existing ontologies did not provide results for all extracted terms and their synonyms, it was necessary to complete the ontology with several properties and classes introduced in our namespace (lcc). For example, a new property *lcc:hasQuantityValue* has been introduced to the *ssn:ObservationValue* class. Furthermore, a complete hierarchy for the

[8]http://lov.okfn.org/

[9]http://smartcity.linkeddata.es/

[10]http://protege.stanford.edu/

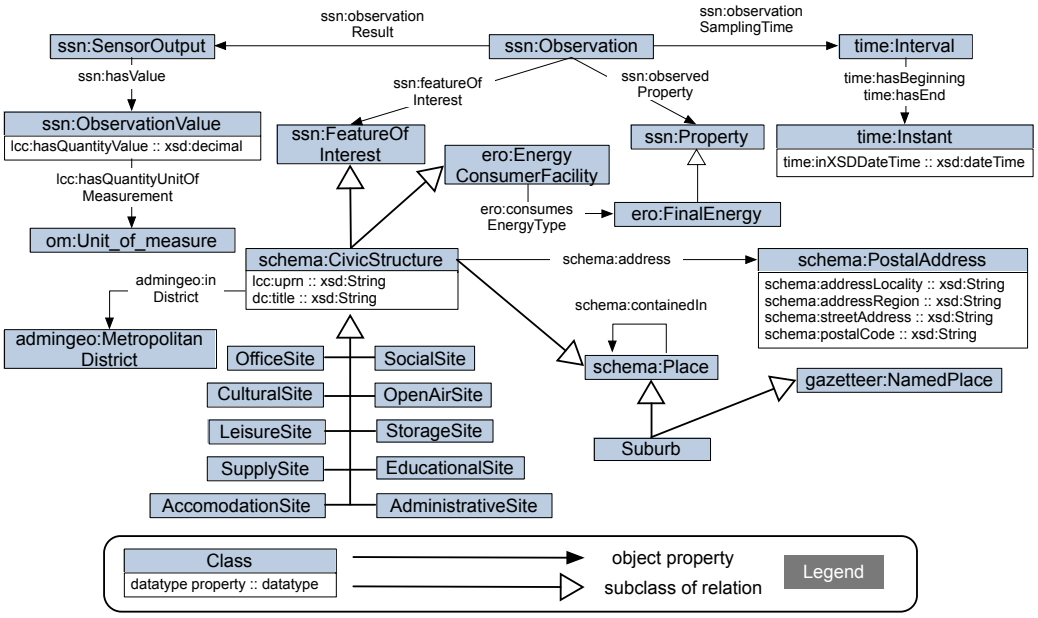

Figure 1: EnergyConsumption ontology for the LCC example.

schema:CivicStructure class has been introduced. For example, a new class *lcc:CulturalSite* has been introduced for representing council sites related to culture, together with related sub-classes (*schema:Museum* and *lcc:Library*). Also, a new class has been introduced to represent suburbs.

The ontology developed for this example has been evaluated using the OOPS! pitfall scanner[11], the syntax of the ontology was also validated, and the Pellet[12] reasoner was used in order to evaluate the logical consistency of the ontology.

3.7. Transform Data Source

The ontology and the resource naming strategy are used in the transformation of the data into the RDF format.

The first step of this task is to select the RDF serialization. While several serializations of RDF exist[13] (the W3C recommendations are RDF/XML,

[11] http://oeg-upm.net/oops/
[12] http://clarkparsia.com/pellet/
[13] http://www.w3.org/TR/rdf11-concepts/#rdf-documents

Turtle, N-Triples and JSON-LD), no serialization is better than other and the benefits of using a specific serialization may include simplicity, speed of processing, and readability by humans.

The second step is to select a tool for data transformation, depending on the format of the data (database, spreadsheets, etc.) and on concrete needs of the transformation process (e.g., dynamicity).

The third step is to use the selected tool in order to obtain the RDF data. This usually requires to define a mapping between the data and the ontology, which also specifies the naming of all instances in a data set according to the resource naming strategy defined.

Furthermore, in this step the compliance with the Linked Data principles and best practices should be ensured, in order to facilitate data reuse and discovery. The most relevant aspects to check in this task are [28]: to avoid blank nodes, to use HTTP URIs, to use external URIs, to provide *owl:sameAs* links (addressed in the next task), and to reuse existing terms.

The fourth step is to evaluate the obtained RDF data set. Several approaches for the evaluation of Linked Data exist (some of them supported by tools): validation of the syntax of the RDF produced; licensing evaluation, which includes checking whether the data set contains machine-processable and human-readable indications of the license; checking for literals that are incompatible with the data type ranges; checking whether the data set contains redundant objects (i.e., if it contains any pair of two equivalent objects with different identifiers) and checking whether the data set contains duplicate entries; checking whether the data set uses existing established ontologies to represent its entities; or determining whether a data set provides possibilities for obtaining the necessary information (e.g., in terms of SPARQL [29] queries).

Different file formats and the dynamicity of the data that can be transformed into RDF are addressed through a set of tools that can be used to perform the transformation task. There are tools available for transforming data from databases (e.g., morph-RDB, D2R Server, TopBraid Composer), XML files (e.g., XML2RDF, OpenRefine, TopBraid Composer), spreadsheets (e.g., Excel2rdf, RDF123, XLWrap, TopBraid Composer, OpenRefine), or data streams (e.g., morph-streams D2R server).

LCC example. Since the data set is small and the speed of processing is not an issue, the Turtle serialization was selected because it is easy to read by humans.

Besides, since the data are available in the CSV format, we have selected

OpenRefine[14] with the RDF extension for transforming the data into RDF. This tool was selected because it is easy to use and it is widely-known in the community.

We have used the selected tool to generate RDF Data from the CSV file, having in mind the Linked Data principles and best practices. Using OpenRefine, this can be achieved in several steps: i) making initial transformations to the data in order to correct errors; ii) creating mappings between the columns and rows in the table and the ontology and specifying the pattern for naming instances according to the resource naming strategy; and iii) choosing the RDF syntax and generating the Linked Data.

We evaluated the RDF data by executing several SPARQL queries and by observing the correctness of the obtained results. All the results obtained with SPARQL queries are correct and in line with the original data. Furthermore, we have also validated the representational consistency, i.e., using the Pellet reasoner we have checked whether the RDF data are consistent with the used ontologies.

3.8. Link with Other Data Sets

The task of data linking has the goal of creating links in the RDF data [30] and it can be achieved in several consecutive steps by using the RDF data set and the ontology as inputs. The first step is to identify classes whose instances can be the subject of linking, while the second step is to identify data sets that may contain instances for the previously-identified classes.

The third step is to select the tools for performing the task. Different tools for data linking exist, and each tool has its advantages and provides different functionalities for certain matching tasks. However, in some cases, the linking can be performed manually (e.g., when the generated data set is small, or when the number of instances to link is low), and the next step is not necessarily performed. The fourth step is to use the tool in order to obtain links. Different tools are used differently and each tool requires configuration from the user in a specific form.

Tools that can be used for data linking include LN2R, LD mapper, Silk, LIMES, RDF-AI, Serimi, and OpenRefine with the reconciliation service of the RDF extension.

LCC example. The classes whose instances can be subjects of linking

[14]http://openrefine.org/

were identified first. In the case of the ontology developed for the LCC example, the identified classes include *lcc:Suburb* and *admingeo:MetropolitanDistrict*.

DBpedia is a database containing the structured content from the Wikipedia pages in RDF format; it was identified as the data set that might contain relevant instances of the previously mentioned classes.

Since in the LCC RDF data set there is a significant number of instances that can be linked, and since OpenRefine has been used for transforming the data into RDF, we have selected OpenRefine for performing the linking task.

The RDF extension of OpenRefine can perform the search task and find the links which can be represented in a separate column in OpenRefine and mapped with the related RDF instances using the *owl:sameAs* property. When such mapping is created, the links appear in the RDF data exported by OpenRefine.

In total, there were 120 different instances that the tool suggested for creating links with instances in DBpedia. After performing the linking process, 110 of these instances have been linked.

4. Linked Data Publication Process

The goal of the Linked Data publication process is to make available and discoverable on the Web the generated linked data set and its associated ontology. This section describes the tasks that compose this process.

4.1. Publish the Data Set and the Ontology on the Web

The goal of this step is to make accessible through the Web the main products of the generation process, that is, the ontology and the RDF data set. This step should carefully follow existing principles and best practices in order to achieve the desired added value for the publisher. In particular, both the ontology and the RDF data set should be published in a way that adheres to the Linked Data principles. Moreover, the publication process must be aligned with the desired access policies; to this end, both the HTTP stack and Linked Data technologies provide the access control mechanisms to do so. For instance, the publisher could decide to enable access exclusively within a particular local network, to require credentials, etc.

The first step in this task is to store the RDF data into a persistent repository where they can be then accessed and queried. A natural choice is to use an RDF repository, i.e, a graph-oriented repository whose main advantage is that it offers the possibility of querying the RDF data set using

the standard SPARQL query language. Nevertheless, there are other ways to store RDF data that could integrate better with existing infrastructures or architectures of the organization publishing the data. Furthermore, in this step the ontology must also be published online in a file.

The second step is to enable resolvable HTTP URIs and content negotiation, i.e., the mechanisms for accessing the data through the Web. The second principle of Linked Data recommends to use HTTP URIs, which allow the retrieval, creation, update and deletion of RDF data using the standard and generic methods provided by the HTTP protocol (mainly GET, POST, PUT, and DELETE). Additionally, other common recommendation is to provide content negotiation for clients requesting different representations of the data, meaning the data could be served in different formats and serializations such as HTML, JSON or Turtle, depending on the request made by a certain user agent (a browser, a semantic application, etc.). Although the publisher can implement a service layer that provides HTTP access and content negotiation to the repository, there are several out-of-the-box solutions (frequently called linked data front-ends) that enable the publisher to easily set up the HTTP access to the data set in way that is compliant with standards and best practices.

The third step is to enable a SPARQL HTTP endpoint. One of the advantages of using RDF to model the data set is that it can be queried in a standard query language, namely SPARQL. Once the publisher has set up the RDF store and loaded the data, access through HTTP using SPARQL can be configured. This configuration should take care of security and performance issues because having an open repository on the Web comes with potential problems such as very expensive or harmful queries that could slow-down or even completely halt the service.

Besides specialized RDF repositories, there are other options for storing RDF, such as using a relational database system or a so-called NoSQL database system (see [31] for an empirical evaluation of existing solutions).

LCC example. We have chosen to store the RDF data set into a specialized RDF repository; in particular, the data have been stored into Openlink's Virtuoso Open Source repository[15]. It is important to have in mind that in this step, the data are not yet available on the Web. Besides, the ontology

[15]https://github.com/openlink/virtuoso-opensource

developed for the LCC example has been published online[16].

In order to enable HTTP access to the data, a Linked Data front-end has been selected and configured. In particular, we have chosen the Elda implementation of the Linked Data API specification. This front-end guarantees access via HTTP to our data and enables content-negotiation to allow consumers to request the data in different formats.

The last step in the process has been to enable access to the RDF store settled up in the first step. For this, we have configured our Virtuoso store to be accessible through the SPARQL HTTP protocol and have enabled public access[17]. This public access allows anyone to query our repository using the SPARQL language, but it is important to note that this access could be restricted using standard HTTP security mechanisms and a more specialized configuration of the repository.

4.2. Publish Metadata and Online Documentation

Once the RDF data have been stored, the next task is to create and publish the documentation of the RDF data set and the ontology. This documentation is oriented to both human and machine users and its purpose is to facilitate the usage of the data set that is being made available.

The first step in this task is to create and publish machine-readable metadata descriptions. In recent years two vocabularies published by the W3C allow describing data sets and data catalogs in RDF: VoID (Vocabulary of Interlinked Datasets) and DCAT (Data Catalog Vocabulary)[18]. The VoID vocabulary [32] focuses exclusively on linked data sets with metadata elements to describe general aspects (e.g., name, authors, license), access methods (addresses of SPARQL endpoints and data dumps), structural characteristics (e.g., URI patterns, vocabularies used, statistics), and links (e.g., target data sets linked to by the data set). DCAT, on the other hand, is oriented to any type of data set/catalog (including Linked Data data sets) and provides a richer set of metadata elements to describe the data in terms of versions, composition of catalogs, or maintenance. Both vocabularies are complementary and the recommendation is to describe the data set first using DCAT and to provide further descriptions of exclusive Linked Data aspects using

[16]http://smartcity.linkeddata.es/lcc/ontology/EnergyConsumption
[17]http://smartcity.linkeddata.es/sparql
[18]http://www.w3.org/TR/vocab-dcat/

VoID. Regarding the publication of these descriptions, they should be published in the same way as the RDF data set and the ontology by following the Linked Data principles and best practices.

The metadata description must include licensing information. If the data set has to be used in an industry setting, where litigation around intellectual property is a real possibility, the data set must be perceived as a trustable asset with clear licensing terms. Data publishers should always publish a license along with the published data set. In the case when the publisher is allowed to publish the data set, in order to define and publicize the license of the data set to be released, the first step is to choose the right license. If the publisher is also the data rightsholder, any license can be chosen; if the data to be published includes (or is based on) data from other parties, the license must encompass possible restrictions imposed by other parties. Finally, an appropriate method to publish the license has to be chosen, ensuring that the license is visible both to humans and to machines. To achieve this, a Dublin Core *license* element is the most recommended choice. If using HTML, introducing RDFa annotations [33] is a good practice.

The second step is to create and publish a human-readable documentation of the data set and ontology. Providing documentation about the data set and the ontology can ease data usage to consumers. As with APIs, a good documentation helps developers to understand the available access mechanisms and the underlying data model (i.e., the ontology). Regarding the ontology, there are existing tools that can help the publisher to semi-automatically generate a human-readable documentation based on the machine-readable descriptions and axioms available in the ontology. Regarding the data set, the documentation could be generated out of VoID and DCAT descriptions, but it is recommended to use an online data catalog such as datahub.io[19] to save time and effort and benefit from the visibility provided by this kind of repositories.

LCC example. For our running example, we have created a data set description using DCAT and VoID in combination. This machine-oriented description has been made available online[20]. Besides, we have decided to use a data catalog to provide information about the data as we will describe in the next section. The human-oriented documentation of the ontology has

[19]http://datahub.io/

[20]http://smartcity.linkeddata.es/lcc/dcat.ttl

been semi-automatically produced using Widoco and is available online[21].

4.3. Enable Data Set Discovery

The goal of this step is to enable the mechanisms to complement the efforts from the previous step and to allow both human and machines to discover and better use the data set. These mechanisms include traditional ways oriented to search engines, such as the so-called sitemaps, and other more oriented to the new trend of publishing data on the Web such as data catalogs. Given the high number of choices, in this step we focus on three simple steps that can offer visibility and discover-ability while minimizing the invested effort.

The first step is to create a sitemap, either manually or using some tool. A sitemap[22] is a mechanism to inform search engines about the page structure of a certain web site in order to allow for a more efficient crawling. It is widely used and adopted by major search engines and it is therefore recommended for any type of web site including data sets. Once created, the sitemap should be uploaded to the major search engines.

The second step is to register the data set in datahub.io. Currently, there are available several online data catalogs, that range from general open data catalogs like datacatalogs.org to corporate initiatives like Google Public Data[23]. Although these catalogs are interesting and could help discover-ability, in a Linked Data scenario we recommend to register the data set into the datahub.io catalog, given that it is cross-domain, widely-used and allows for automatic crawling by the system that creates the LOD cloud registry, as we will seen in the next point.

The third step is to ensure the fulfillment of requirements to be added to the LOD cloud. As mentioned above, registering the data set into datahub.io can enable our data set to be promoted within the LOD cloud initiative which can boost its visibility and, ultimately, its reuse and connection to other data sets. In order to ensure that, it is necessary to follow a set of recommendations[24] that basically consist on adding the proper metadata

[21] http://smartcity.linkeddata.es/lcc/ontology/EnergyConsumption

[22] http://www.sitemaps.org/

[23] https://www.google.com/publicdata/admin/

[24] http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/DataSets/CKANmetainformation

and ensuring that the published data set conforms to a set of Linked Data best practices.

LCC example. We have used the sitemap4rdf tool[25] to generate a (semantic) sitemap of our data set[26]. This sitemap is automatically generated by extracting data directly from the SPARQL endpoint.

The data set has been registered into datahub.io[27], precisely describing its characteristics and focusing on easing the access and reuse of the data set. We have made sure to fulfill the criteria to be included in the LOD cloud using the datahub.io catalog entry to do so. In particular, we have followed the guidelines provided by the LOD cloud community and checked the result with the record validator[28] provided for this in the LOD cloud website.

5. Discussion and Conclusions

Although some general guidelines for Linked Data generation and publication exist, experience has shown that such general guidelines are not always sufficient in order to be applied to every domain. In order to overcome this issue, guidelines that are more domain-oriented need to be developed. Such guidelines tend to address domain-specific characteristics and provide domain-related examples, which help the community to better understand Linked Data technologies and might lead to their faster adoption.

This paper presents a set of guidelines for Linked Data generation and publication, together with one complete example in the domain of energy consumption in buildings. By providing detailed descriptions of each task in the generation and publication processes, these guidelines help both private and public organizations that work with data about energy consumption in buildings in generating Linked Data from already-existing data and in publishing the generated data according to the latest standards.

This paper also presents a complete example of how to use the guidelines in order to generate and publish energy consumption data as Linked Data, in particular the energy consumption data from the Leeds City Council. This example helps the audience from different organizations to gain better insight

[25]http://lab.linkeddata.deri.ie/2010/sitemap4rdf/
[26]http://smartcity.linkeddata.es/lcc/sitemap.xml
[27]http://datahub.io/dataset/lcc-leeds-city-council-energy-consumption-linked-data
[28]http://validator.lod-cloud.net/

into the processes of Linked Data generation and publication, thus ensuring the highest quality of the outputs of these processes.

The guidelines presented in this paper are aimed to help researchers and practitioners interested in energy consumption in buildings in exploiting Linked Data technologies. Since it is reasonable to expect that such technologies are new to target practitioners, future work will deal with creating a set of services for facilitating the usage of Linked Data technologies. Such services will help practitioners in adopting these technologies, and thus create benefits for their organizations.

We expect the building modelling community to actively take part and to exploit the benefits of Linked Data technologies by generating and publishing their data as Linked Data. To that extent, the guidelines presented in this paper are a valuable resource to achieve this goal.

6. Acknowledgements

This work has been supported by the READY4SmartCities European project (FP7-608711) and by the FPU grant (FPU2012/04084) of the Spanish Ministry of Education, Culture and Sport. We would like to thank the members of READY4SmartCities for their feedback about the guidelines, specially Thanasis Tryferidis and Matthias Weise, and the people from the Leeds City Council Open Data office for providing us with additional information about their data set.

References

[1] T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web, Scientific American 284 (5) (2001) 34–43.

[2] W3C OWL Working Group, OWL 2 Web Ontology Language. Available online: http://www.w3.org/TR/owl2-overview/. Last retrieved on 25.09.2014., Tech. rep., World Wide Web Consortium (2012).

[3] D. Brickley, R. Guha, RDF Schema 1.1. Available online: http://www.w3.org/TR/2014/REC-rdf-schema-20140225/. Last retrieved on 24.10.2014., Tech. rep., World Wide Web Consortium (2014).

[4] F. Radulovic, R. García-Castro, M. Poveda-Villalón, M. Weise, T. Tryferidis, D4.1: Requirements and guidelines for energy data generation, Tech. rep., READY4SmartCities Consortium (May 2014).

[5] P. Pauwels, R. De Meyer, J. Van Campenhout, Interoperability for the design and construction industry through Semantic Web technology, in: T. Declerck, M. Granitzer, M. Grzegorzek, M. Romanelli, S. Rger, M. Sintek (Eds.), Semantic Multimedia, Springer, 2011, pp. 143–158.

[6] F. Abanda, W. Zhou, J. Tah, F. Cheung, Exploring the relationships between Linked Open Data and Building Information Modelling, in: Proceedings of the Sustainable Building Conference, Coventry University, UK, July, 2013, pp. 176–185.

[7] L. Madrazo, G. Costa, Open product modelling and interoperability in the AEC sector, in: Proceedings of the 1st International Workshop on Linked Data in Architecture and Construction (LDAC 2012), Ghent, Belgium, March, 2012, pp. 4–6.

[8] S. Törmä, Web of building data–integrating IFC with the web of data, in: Proceedings of the 10th European Conference on Product and Process Modelling – eWork and eBusiness in Architecture, Engineering and Construction, Vienna, Austria, September, 2014, p. 141.

[9] S. Törmä, J. Oraskari, N. V. Hoang, Distributed transactional building information management, in: Proceedings of the 1st International Workshop on Linked Data in Architecture and Construction (LDAC 2012), Ghent, Belgium, March, 2012, pp. 9–11.

[10] P. Pauwels, D. Van Deursen, IFC/RDF: Adaptation, aggregation and enrichment, in: Proceedings of the 1st International Workshop on Linked Data in Architecture and Construction (LDAC 2012), Ghent, Belgium, March, 2012, pp. 2–4.

[11] ISO, ISO 16739:2013 - Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries. International Standardization Organization (2013).

[12] E. Curry, J. ODonnell, E. Corry, Building optimisation using scenario modeling and Linked Data, in: Proceedings of the 1st International

Workshop on Linked Data in Architecture and Construction (LDAC 2012), Ghent, Belgium, March, 2012, pp. 6–8.

[13] E. Curry, J. O'Donnell, E. Corry, S. Hasan, M. Keane, S. ORiain, Linking building data in the cloud: Integrating cross-domain building data using Linked Data, Advanced Engineering Informatics 27 (2) (2013) 206–219.

[14] J. O'Donnell, E. Corry, S. Hasan, M. Keane, E. Curry, Building performance optimization using cross-domain scenario modeling, Linked Data, and complex event processing, Building and Environment 62 (2013) 102–111.

[15] F. Bauer, D. Recheis, M. Kaltenböck, data.reegle.info – a new key portal for Open Energy Data, in: J. Hebek, G. Schimak, R. Denzer (Eds.), Environmental Software Systems. Frameworks of eEnvironment, Springer, 2011, pp. 189–194.

[16] T. Heath, C. Bizer, Linked Data: Evolving the Web into a Global Data Space, 1st Edition, Morgan & Claypool, 2011.

[17] S. Auer, V. Bryl, S. Tramp, Linked Open Data – Creating knowledge out of Interlinked Data: Results of the LOD2 Project, Vol. 8661, Springer, 2014.

[18] L. Sauermann, R. Cyganiak, Cool URIs for the Semantic Web. Available online: http://www.w3.org/TR/cooluris. Last retrieved on 15.08.2014., Tech. rep., World Wide Web Consortium (2008).

[19] UK Government Cabinet Office, Designing URI sets for the UK public sector, Tech. rep., UK Government Cabinet Office (2010).

[20] SEMIC, 10 Rules for persistent URIs, Tech. rep., Semantic Interoperability Community, European Commission (2012).

[21] L. Dodds, I. Davis, Linked Data patterns. Available online: http://patterns.dataincubator.org/book/. Last retrieved on 15.08.2014.

[22] M. Poveda-Villalón, A reuse-based lightweight method for developing Linked Data ontologies and vocabularies, in: Proceedings of the 9th Extended Semantic Web Conference (ESWC2012). Heraklion, Crete, May, 2012, pp. 833–837.

[23] M.C. Suárez-Figueroa, NeOn methodology for building ontology networks: Specification, scheduling and reuse, Ph.D. thesis, Universidad Politécnica de Madrid (2010).

[24] M. Poveda-Villalón and M.C. Suárez-Figueroa and A. Gómez-Pérez, The landscape of ontology reuse in Linked Data, in: Proceedings of the 1st Ontology Engineering in a Data-driven World Workshop (OEDW 2012), Galway, Ireland, October, 2012.

[25] SEMIC, Cookbook for translating Data Models to RDF Schemas, Tech. rep., Semantic Interoperability Community, European Commission (2013).

[26] M. Suárez-Figueroa, G. A. de Cea, A. Gómez-Pérez, Lights and shadows in creating a glossary about ontology engineering, Terminology 19(2) (2013) 202–236.

[27] M. Poveda-Villalón, A. Gómez-Pérez, M. C. Suárez-Figueroa, Oops!(ontology pitfall scanner!): An on-line tool for ontology evaluation, International Journal on Semantic Web and Information Systems 10 (2) (2014) 7–34.

[28] A. Hogan, J. Umbrich, A. Harth, R. Cyganiak, A. Polleres, S. Decker, An empirical survey of Linked Data conformance, Journal of Web Semantics 14 (2012) 14–44.

[29] E. Prud'hommeaux, A. Seaborne, H.-P. L. Bristol, SPARQL query language for RDF. Available online: http://www.w3.org/TR/rdf-sparql-query/. Last retrieved on 17.10.2014., Tech. rep., World Wide Web Consortium (2008).

[30] A. Nikolov, A. Ferrara, F. Scharffe, Data linking for the Semantic Web, International Journal on Semantic Web & Information Systems 7 (2011) 46–76.

[31] P. Cudré-Mauroux, I. Enchev, S. Fundatureanu, P. T. Groth, A. Haque, A. Harth, F. L. Keppmann, D. P. Miranker, J. Sequeda, M. Wylot, NoSQL databases for RDF: An empirical evaluation, in: Proceedings of the 12th International Semantic Web Conference, Sydney, NSW, Australia, October, 2013, pp. 310–325.

[32] K. Alexander, R. Cyganiak, M. Hausenblas, J. Zhao, Describing Linked Datasets with the voID Vocabulary. W3C Interest Group Note. Available online: http://www.w3.org/TR/void/. Last retrieved on 17.10.2014., Tech. rep., World Wide Web Consortium (2010).

[33] B. Adida, M. Birbeck, M. Shane, I. Herman, RDFa Core 1.1 - Second Edition. Available online: http://www.w3.org/TR/rdfa-core/. Last retrieved on 23.10.2014., Tech. rep., World Wide Web Consortium (2013).