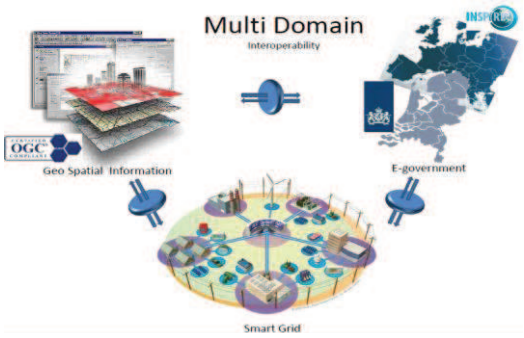


CERISE

Combining Energy and Spatial Information Standards as Enabler for Smart Grids

TKI Smart Grid Project: TKISG01010

D5.1 Cookbook for Standardization and Harmonization
D5.2 State–of-the-art and tools harmonization
D6.4 Evaluation test-bed

Work package – 50 & 60

Lead partner: TNO

23 September 2015

Versie 1.0 - Final

DOCUMENT INFORMATION	
ID	D5.1 Cookbook – D5.2 State-of-the-art – D6.1 Evaluation test-bed
Work package	WP50 State-of-the-art & WP60 Testbed
Type	Report
Dissemination	Public
Version	1.0 - Final
Date	23 September 2015
Author(s)	Maarten Steen (TNO), Frans Knibbe (Geodan), Wilko Quak (TUD), Jasper Roes (TNO), Laura Daniele (TNO)
Reviewer(s)	Roel Stap (Alliander)

Table of contents

Summary

The CERISE-SG project (Combining Energy and Geo information standards as enabler for Smart Grids) focuses on with respect to information exchange between smart grids and their surroundings. The project focusses on information exchange to and from smart grids, the government domain and the geo domain. Within the fast changing smart grid world acquiring reliable information from different sources is invaluable. The information required comes from different sources that all use their own (often different) definitions for the data they control. The national registration of buildings for instance contains different data with different definitions from the data source of energy consumption. To exchange information between these sources connections need to be made between the different areas that make sure that correct and reliable data is available.

This report is a combination of three deliverables:

- D5.1 Cookbook for Standardization and Harmonization
- D5.2 State–of-the-art and tools harmonization
- D6.4 Evaluation test-bed

In this report we give an overview of the state-of-the-art in information exchange by describing the concept of Linked Data and we present a number of recipes that can be used by the reader to:

- Publish data as Linked Data
- To generate a profile from UML to OWL
- To use Linked Data in web applications

The recipes defined assume that the reader has a good knowledge of information technology and is accustomed to using a variety of different IT-tools.

Finally this report contains our experiences with using Linked Data in the test-bed that was developed by the project. Next to providing guidelines to the reader on how to use Linked Data in web applications it also serves as an evaluation of the testbed.

1 Introduction

The overall goal of CERISE-SG is to support future-proof information exchange between the energy, e-government and geography domains in order to enable the realization and management of smart energy grids. Each domain has its own set of standards and information models for exchanging information within that domain, and even within domains there still are interoperability problems. This document contains practical descriptions of how to achieve interoperability between various data sets and describes the state of the art on Linked Data.

The contents of chapters 2, 3 and 4 in this report are identical to the chapters 2, 3 and 4 in deliverable D4.1 to make sure that both reports are separately readable. A reader who has already read deliverable D4.1 can therefore start reading in chapter 5.

This document contains a description of the harmonization problem (chapter 2) and an overview of different approaches to solving that problem (chapter 3). Within the project, we opted for one particular approach: Linked Data. It is explained in section 4. The rest of this deliverable consists of recipes for various activities that are required to be performed when harmonizing data exchanges using Linked Data. Section 5 contains recipes for the data published, section 6 a recipe for formalizing mappings and section 7 contains recipes for using Linked Data in applications. Section 8 contains the conclusion and some recommendations.

The target audience for this deliverable are people that are interested in applying Linked Data to solve a harmonization problem and that need guidelines on the various activities involved.

This deliverable is a combination of the Cookbook deliverable of WP50 (see chapters 4, 5 and 6), the State of the art deliverable of WP50 (see chapters 1, 2 and 3) and the Evaluation of the testbed of WP60 (see chapter 6). We combined these three deliverables into this one deliverable as they are very much related due to the fact the we used Linked Data to integrate different sources of data. Linked Data is currently state-of-the-art in the information integration domain and as we already applied it, there is no need to write a separate State of the art report. The evaluation of the test bed also fit this deliverable as we applied Linked Data principles in the test bed.

2 The harmonization challenge

Smart energy grids are a relatively recent phenomenon. Enabling them requires data, and those data cannot be found solely in one domain. Instead, the required data should come from different domains, i.e. also from outside the grid. The operation of smart grids depends on a combination utilities data, governmental data and spatial data.

Although existing structures often are based on standards, they are not always set up with external interoperability in mind as a primary design objective. And existing structures often stem from before the web era, the first time in history when things could be interconnected on a global scale. In the following paragraphs we describe how the three major data domains for CERISE-SG are set up.

2.1 Governmental data

Many data that are important to make Smart Grids work are government data, although the situation will be different in different countries. In the Netherlands, the government recognizes the need for making governmental data interoperable, as a means of providing better service to citizens and as a means of improving efficiency within the government itself. Also the Dutch government is aware of growing needs for automation and digitization of information flows, eGovernment. There are several initiatives for standardizing data exchange and information gathering within the Dutch government. An overarching framework is the Dutch Government Reference Architecture NORA[1] (*Nederlandse Overheid Referentie Architectuur*). It mostly describes quality criteria. For specific levels of government (e.g. provincial, municipal) more specific frameworks have been derived from NORA (e.g. EAR, GEMMA, PETRA). Most, probably all data exchange specifications are based on XML.

A national model for facilitating data exchange not only serves as a basis for regional models, it also has to comply with international models. These primarily come from the European Union, with its strong drive towards cooperation between its member states. Notable European frameworks for data exchange are the European Information Framework (EIF) and the Infrastructure for Spatial Information in the European Community (INSPIRE). In INSPIRE one of the data themes is ‘Energy resources’. This theme mainly deals with data on primary energy sources like hydrocarbons, wind and solar irradiation.

A national framework that is of particular importance to Smart Grids is the System of Base Registries (*Stelsel van Basisregistraties*)[2]. An outline of the framework is given in figure 1. Base Registries are important mainly because they contain data for many relevant topics such as persons, buildings, vehicles, addresses and topography. Those kinds of data are essential for many applications of governmental data. Work to harmonize the base registries is ongoing. A result of that work is a common catalogue of definitions, the *Stelselcatalogus* (system catalogue).

[1] http://www.noraonline.nl/wiki/NORA_online
[2] http://www.digitaleoverheid.nl/onderwerpen/stelselinformatiepunt

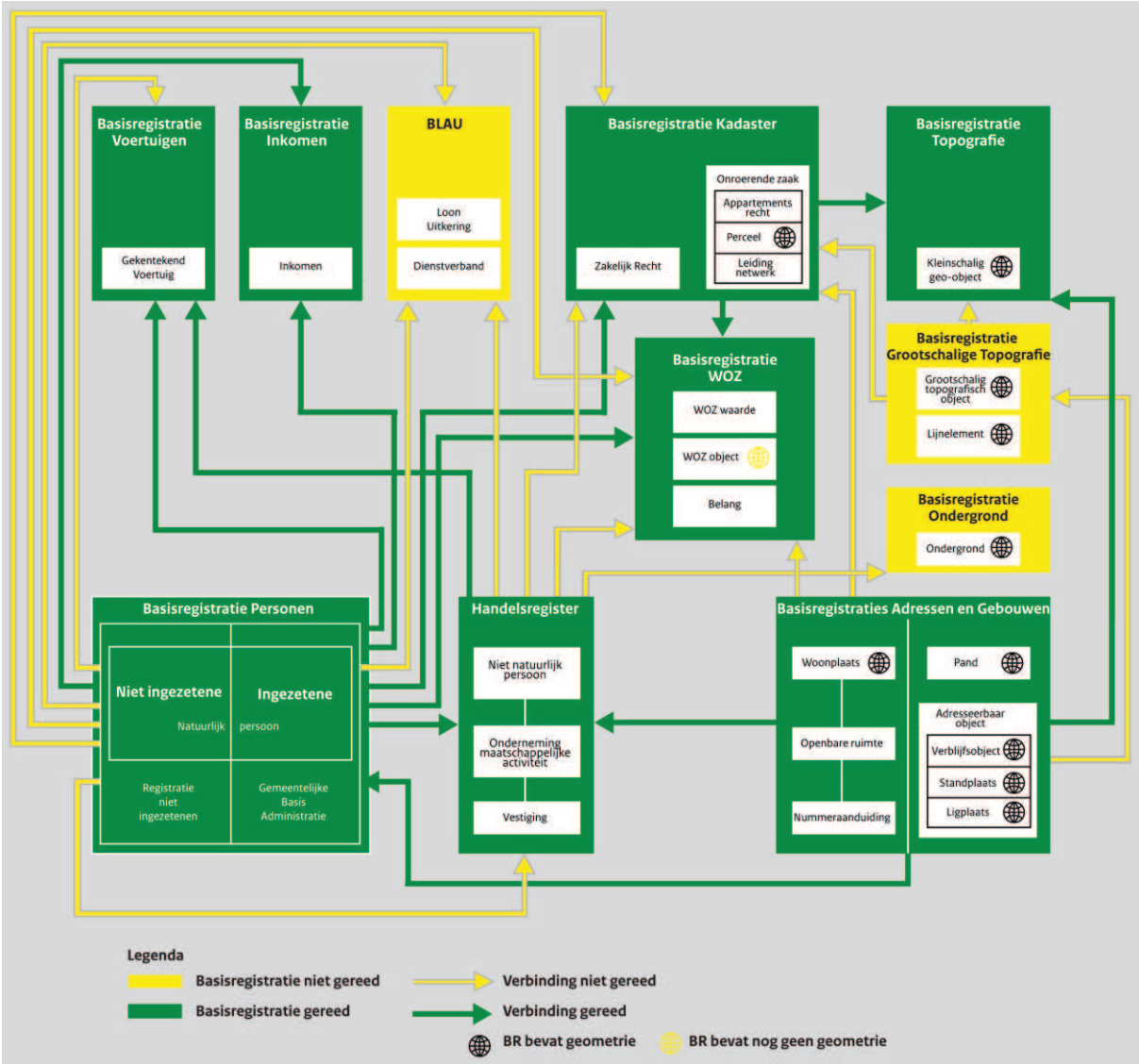


Figure 1 Diagram showing the different base registries and their relationships (in Dutch). Arrows indicate links between registries. The yellow part of the diagram is under construction, the green part has already been established.

2.2 Utilities data

The intrinsic drive for ITC standardization of data exchange in the utilities sector traditionally has been less than in the public sector. Nonetheless, recent global and national developments have caused increased activity in this area. This has resulted in the development of the Common Information Model (CIM)[3], a global standard adopted by the International Electrotechnical Commission (IEC). CIM information model is developed as a UML model for among other transmission and distribution of electric power[4]. It is foreseen that electric power companies will make increased use of CIM for exchanging information between applications with other parties, or within their own organisations.

[3] http://www.dmtf.org/standards/cim
[4] IEC is currently working on an extension of CIM for natural gas and water

In light of harmonization is should be mentioned that the CIM model is self-contained: It does not reuse elements from other models but has its own definitions of model elements (e.g. classes, properties, relationships). The CIM model is an extensive model that has a lot of detail. For specific applications it is possible to derive and use simpler subsets of the model. These subsets are called CIM profiles.

2.3 Geographic data

The domain of geography is somewhat different than the government and utilities domains. Rather than being concerned with an aspect of society, it is concerned with a special kind of data: geographical data. Like the two domains described above, it also has its heritage of standardisation, and it too has its own way of achieving standardisation.

The important standards body in the domain of geographic data is the Open Geospatial Consortium. It has developed a general model for geographic features, on which various standards are based. Information models are described in UML and encoded in XML. There are standards for various web services for the exchange of geographic data or geographic information. The OGC has a Domain Working Group for the energy and utilities domain (see http://www.opengeospatial.org/projects/groups/energyutilities), but work in that group has not lead to standards or recommendations yet.

Like CIM for the utilities domain, OGC standards are also largely self-contained. Other than basic XML data types (e.g. text string, date, number) no external data definitions are used.

2.4 Conclusions

The descriptions above show that within the three domains that CERISE-SG is concerned with there is a clear drive towards standardization, but the resulting standards are mainly useable within their respective domains, not so much outside of it. With existing domains being set up in different ways, efficiently combining data, for example to enable Smart Grids, requires a harmonization effort.

A benefit of current domain standards being based on common practices for information[5] modelling and information exchange like UML and XML, is that syntactic interoperability is not much of a problem. Semantic interoperability becomes a problem, as soon as information concepts surpass basic XSD datatypes. Something has to be done to make information from domains with different designs interoperable. Possible harmonization strategies will be discussed in the next section.

[5] Information is that which informs, i.e. an answer to a question, as well as that from which knowledge and data can be derived (as data represents values attributed to parameters, and knowledge signifies understanding of real things or abstract concepts) (https://en.wikipedia.org/wiki/Information)

3 Possibilities for semantic interoperability between domains

As explained in chapter 2, interoperability is needed between different domains that each have their own way of describing the world, or that part of the world that is of interest to the domain. This is a general problem, for which an optimal general solution should be found. The problem is mainly one of semantic interoperability. Should it become possible for a party with an interest in obtaining data from multiple domains to express a data query using a single semantic model, then actually performing the query and getting a meaningful set of data in response should be straightforward.

Semantic harmonization involves two basic types of problem. The first is the case of the same concepts being defined in different ways in domain models. This happens for common concepts, like ‘person’, ‘address’ or ‘location’. For example, the way a utilities information model defines a person should be interoperable with the way a governmental model defines a person. The second type of problem is definitions of specialized concepts that only exist in one of the domain models. It should be noted that the second case occurs less often than one might expect, because in most domain models class hierarchies are used, in which specialized concept definitions are derived from more abstract definitions. The more abstract a concept, the higher the likelihood of it having some semantic overlap with a concept from another model.

In the following sections two different methods for achieving semantic interoperability are described.

3.1 Define relationships between elements from different models

One way of achieving semantic interoperability is to define mappings between entities in the domain models. This should only have to happen for those concepts that are shared between models. Concepts that are uniquely defined within a single domain model do not have to be mapped to another model, their original definitions can be used.

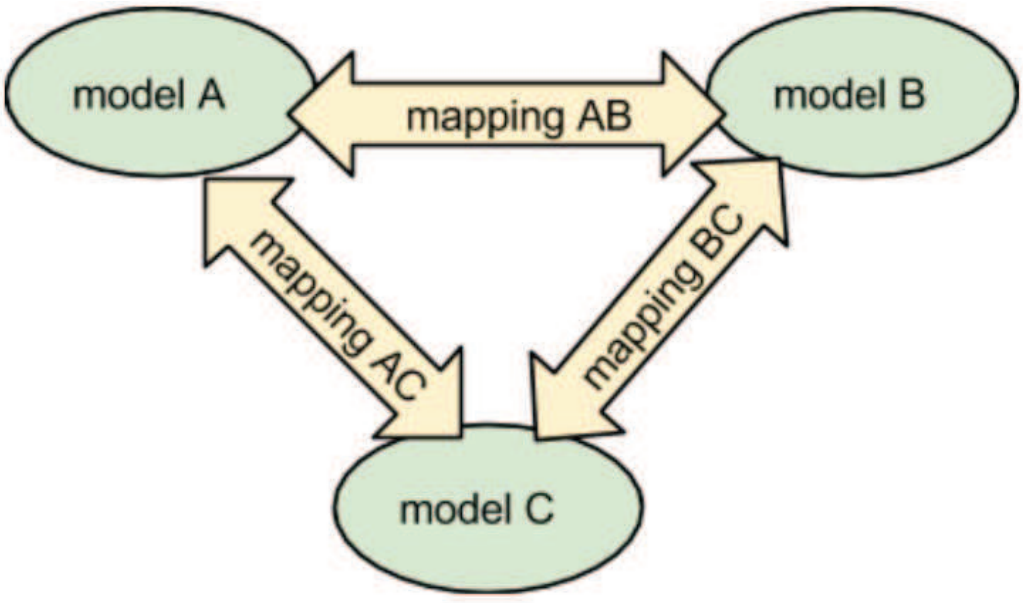


Figure 2 Establishing semantic interoperarability by defining mappings between the various domains.

This method of semantic mapping is especially suited for use from within one of the domains. For example, someone working in the utilities domain could make use of the

utilities-government mapping to obtain data from a data source that uses a governmental information model. For use outside of the domains this method seems less suited. Within the context of CERISE-SG there are examples of this kind of use: a neighbourhood energy collective does not have its own domain model, but will need to obtain data from different domains with existing models.

An important disadvantage of this approach is that when the number of domain models to map increases, the number of required mappings increases drastically. For instance, interoperability between three domain models requires three mappings, but with five domain models twenty domain-domain mappings can be made. Complexity increases even more when domain models change over time, which means that multiple mappings will have to be updated.

3.2 Express model elements from different models in a common model

A different approach is to map concepts from a domain model to concepts from a shared information model. The general model can then be used to express all domain data.

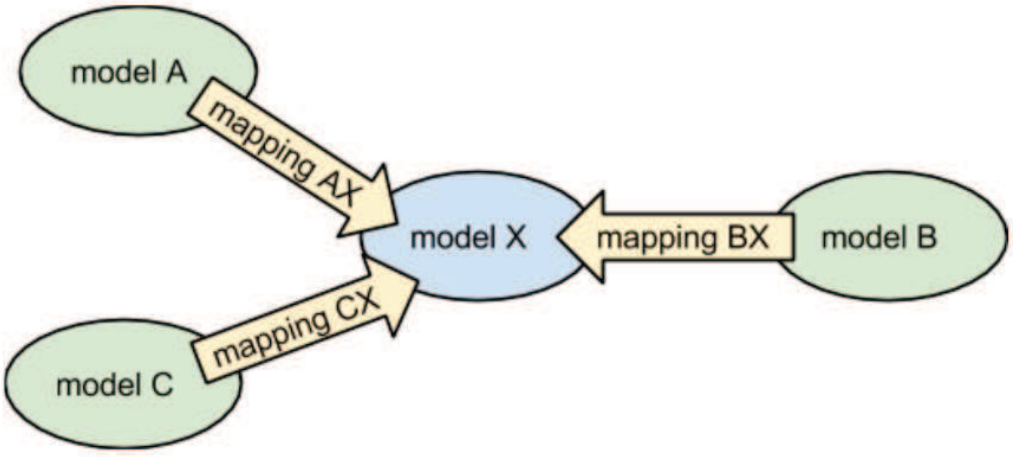


Figure 3 Establishing interoperability by defining mappings to a common shared model.

Care should be taken to make this approach extensible. When new system requirements call for interoperability with yet another domain model, it has to be possible to add another mapping without having to change existing mappings. This means that the shared model should be sufficiently general.

A useful property of this method is that a data consumer only needs to know the general model in order to make sense of data from the domain models. Such use would require *all* concepts in the domain models to be mapped to the general model. For specialized concepts this means that mapping should take place at a sufficiently high abstraction level (e.g. parent class), which in turn means that data consumers could incur a loss of semantic accuracy.

3.3 Conclusion

To make domain data interoperable some sort of semantic mapping needs to be done. Such a mapping can be expressed in a modelling language like OWL or a rule language like SPIN. From these formal mappings automatic transformation procedures can be derived.

4 Practical interoperability

4.1 Linked Data

Given the project mission - achieve interoperability between different domains for users within and outside those domains - we have the Linked Data paradigm[6] as offering the required capabilities for investigating the problem.

In short, Linked Data is a way of sharing raw data on the world wide web. Linked Data is strongly related to the Semantic Web, the idea of annotating information on web pages with semantic tags so that those web pages can easily be interpreted by automatic procedures (machines versus humans). Those automatic procedures could improve the information available to humans, for example by creating and maintaining indexes that make data discoverable, or by enriching data with derived data.

Key principles of Linked Data are:

1. All data (including metadata and semantics) are web resources, identified by HTTP(S) URIs (Uniform Resource Identifiers).
2. Looking up a URI returns data describing the resource.
3. The Semantic Web stack family of standards (RDF, RDFS, SPARQL, OWL, SKOS, SPIN, ...) is used to model and query data.
4. Content negotiation is used to request data in a specified format (e.g. HTML is nice for humans, XML is nice for processing, JSON-LD is nice for web developers).
5. Data providers are encouraged to link their data to other data sources on the web. This way, all data on the web become interconnected and form one global database (or one global graph, since RDF models data as graphs).

Fully explaining the concept of Linked Data falls outside the scope of this document, but there is much information available on the web. A starting point could be http://www.w3.org/standards/semanticweb/data. Nevertheless, some aspects of Linked Data that make it especially suitable for CERISE-SG can be listed:

- It uses modular semantics - small data sets with data definitions (called vocabularies or ontologies) are published on the web and can be used, mixed and matched by data providers.
- It is adopted by many different domains, especially those that want to achieve better inter domain and cross-domain interoperability. Domains working with Linked Data include the three domains that CERISE-SG is concerned with: geography, energy and government.
- It builds on existing web and existing web architecture: much of the system and infrastructure for data exchange is already in place.
- It allows advanced data analysis, e.g. reasoning/inference (see http://www.w3.org/standards/semanticweb/inference)

4.2 Existing Linked Data semantics in the geography, government and utilities domains

Linked Data principles have found their way into the three domains that CERISE-SG is concerned with, to different extents. This section describes the existing semantics in the three domains.

[6] http://linkeddata.org/

4.2.1 Geography

Semantics for geography in the Semantic Web come from both the web and the geography communities. Recognition of geographic geometry as a basic type of data took place early in the development of the Semantic Web. A notable result was the publication of the Basic Geo vocabulary[7] in 2003. It provides semantics for expressing point geometry as latitude/longitude coordinates. A more recent specification is schema.org, a vocabulary developed by major web search engines. Among other things it contains classes for expressing geography, e.g. http://schema.org/GeoShape. Many more vocabularies that can be used to express geographic data have come into existence, both from communities with a web background and with a geographic background. A vocabulary that is recognized as authoritative by people and organizations with a geographic background is GeoSPARQL[8] . It is a standard from the Open Geospatial Consortium (OGC), the main standards body for the geography domain. The GeoSPARQL specification is based on the foundations of the OGC (or ISO/TC 211) theoretical framework for geography that is documented as UML class diagrams. Next to definitions of geometry in RDF, GeoSPARQL defines topological functions for SPARQL (an RDF query language).

Lastly, a specification that is worth mentioning is the ISA Programme Location Core Vocabulary[9] , which is a product of INSPIRE-related research. The vocabulary defines concepts for locations and addresses in a general way, making it easy to apply these semantics as umbrella terms.

4.2.2 Government

The Dutch government, like most national governments, is a large and heavily segmented organization. For that reason it fully understands the need for frictionless data exchange within and between its many subdivisions, and that is why it is looking at what Linked Data can offer. Also there is the understanding that opening up governmental data to the general public has important societal and economic benefits, something that other countries have also realized and have adjusted their policies to that effect (e.g. the UK and the USA). When looking at the best way to provide open data, Linked Data is a consideration (see the five star open data concept: http://5stardata.info/).

Recently a member of the RDF family of standards, SKOS (Simple Knowledge Organisation System) has been put on the comply-or-explain list of the Dutch Standardisation Forum. But with a national government being big and complex, it is understandable that most changes in data exchange techniques and procedures can't be made overnight. At the moment, experiments and pilots are undertaken to get an idea of costs and benefits.

Of immediate interest to CERISE-SG is governmental participation the Platform Linked Open Data Nederland (PLDN), a continuation of the Pilot Linked Open Data Nederland (PiLOD). The system of base registrations plays an important role there, as well as the Dutch Cadastre, an important source of national geographic data. Among the results of the platform is a national strategy for minting URIs, and experimental publication of two important datasets as Linked Data, the BAG (buildings and addresses) and the BGT (large scale base topography).

[7] http://www.w3.org/2003/01/geo/
[8] http://www.opengeospatial.org/standards/geosparql
[9] http://www.w3.org/ns/locn

4.2.3 Utilities

Of the three domains under consideration, until now the utilities domain shows the least inclination to move towards web based data exchange. It is likely that this is because the utilities world I(before the emergence of smart grids was more self-contained than the other domains and had less need for sharing data with the outside world. Nevertheless, the global standard for energy data exchange, CIM, is published in RDF, although not with persistent URIs.

4.3 Possibilities for Semantic mapping in Linked Data

The Linked Data paradigm, with its emphasis on linking data (resources) to common semantics is well suited for solving the problem of semantic interoperability.

A general truth is that the more a data set is linked, the more usable it is. That goes especially for links to vocabularies, which provide meaning to data, and provide automated procedures with the means to combine data from different sources. The key to improving semantic interoperability is to provide linkage to common semantics. That way, a data consumer does not need to know about domain specific semantics, but can use general expressions to discover and filter data. For example, a data set containing some address data (e.g. street name and number, postal code and city) could be published on the web using semantic annotation that is specific to the data set. The semantics could be derived from the names of the columns that are used in the relational database where the data are internally stored. According to the Five Star Data scheme, this would count as four star data. To make the data more useful for consumers, the address data could be linked to additional semantics from a general domain model. Both the utilities domain and the national government domain have their own semantics for address data. That would make that part of the data set interoperable with other data sets from the same domain. A further improvement can be made if the address data are also linked to global semantics (for example the Location Core Vocabulary). When that happens, the data are usable by user agents from any domain.

It should be stressed that common semantics do not need to replace local semantics. In an RDF dataset it is possible and perfectly acceptable to model data using different models (vocabularies). It will be up to the requirements of the data consumers which semantics that are provided will be used. Semantics with a narrow scope could carry over details that have been abstracted away in more general models, while general semantics provide the means of data harmonization and interoperability.

Two different strategies for providing access to common semantics (and through that, achieving semantic interoperability) can be distinguished. These two strategies could be viewed as exclusive ways of achieving interoperability, but they can also be applied both, mutually supporting each other.

4.3.1 Use an external mapping and reasoner

In this strategy, links from locally defined concepts to more general concepts are not included in the published data set, but are defined externally and optionally augmented with a smart reasoner. A mapping between the local semantics and the more general semantics can be published as a separate dataset, where local semantic resources (identified by URIs) are related to general semantic resources (also identified by URIs). This mapping could be straightforward (stating that two classes are equivalent), or more complex, using rules. Formalizations like OWL and SPIN are well suited for expressing the latter kind of mapping.

One can imagine a specialized service on the web that contains these mappings, and also provides the means to use the mappings to infer implicit relationships. Because of

the way RDF is set up, automated procedures can use deductive reasoning to find relationships between resources that have not been explicitly published. Servers with high reasoning capabilities are not required by any standard and are not common when RDF data are published, so this would be an added value.

This method could relieve data publishers of having to add extra semantics to the data they publish, but more is required from data consumers. For one thing, they need to know that there is an external mapping and reasoner available somewhere. So at least there should be a standardised way of linking from the data set to the reasoner. Secondly, the data consumer will have to interact with two web servers to work with the data, instead of one. Thirdly, a single reasoner could be considered a potential single point of failure in an otherwise distributed architecture.

Drawbacks of this strategy could largely be negated if the output of the central service is fed back to the source dataset. The service housing the mappings and reasoner could be made to produce RDF data that could be added to the source data as an enrichment.

4.3.2 Provide general semantics at the source

Instead of having an external service provide the data needed for semantic interoperability, those data can be added to a data set by the data provider. For instance, a data set that is based on CIM could have addresses that are stated to be instance of a CIM address class. A data provider could add extra data to the dataset for the addresses, stating that the addresses are also instances of the address class that is defined in the Location Core Vocabulary. That way the address data would be discoverable and queryable by consumers that only know about general web semantics.

This strategy is more demanding on the data publisher, he or she has to understand common web semantics in order to link to them in the right way. Moreover, sometimes complex rule-based relationships need to be defined if there is no one on one match between local concepts and general concepts.

An important advantage of this method is that interoperability does not rely on the functioning and availability of a single network node (the server that has the mapping and the reasoning capabilities).

4.4 Conclusion

In this chapter it was argued that Linked Data is a very suitable paradigm for achieving the kind of data harmonization that is sought after in CERISE-SG

Two different strategies for adding semantics to data were described. Both these strategies need the same groundwork to be done: mappings between information models need to be made, in order to make the data available with common semantics on the worldwide web of data.

5 Publishing Linked Data

In this chapter we describe recipes for publishing Linked Data. It provides a step-by-step guide for data publishers, illustrated with concrete examples from the energy domain. This guide was produced by members of the CERISE-SG project in close collaboration with another project at TNO and shared and discussed within the Platform Linked Data Netherlands community of practice. We would particularly like to thank Erwin Folmer, Silja Eckartz and Laura Daniele for their contributions to this guide, and Alliander for providing the example dataset. An earlier version was published on the website of the Platform Linked Data Netherlands[10].

5.1 A Step-by-Step Guide for Publishing Linked Data

There are already many guides, textbooks, tutorials and best practices available about linked (open) data. As part of our investigation we have reviewed several of these, but found none of them practical, concise and concrete enough for data publishers to apply directly. In this cookbook we have attempted to collect several of these best practices and compose them into a practical guide for publishing linked (open) data. That being said, our steps are largely based on the best practices from the W3C Linked Data Cookbook[11] and Heath and Bizer's Linked Data book[12].

Our guidelines are divided in nine consecutive activities that should be taken into consideration when publishing linked data:

1. Select data
2. Prepare the data
3. Model the data
4. Define a naming scheme
5. Convert the data
6. Organize Governance
7. Add metadata
8. Publish the data
9. Link the data

In order to illustrate these guidelines we apply them to an example dataset. The example concerns an existing, non-governmental open dataset from Liander, one of the Dutch regional energy distributers.

Liander manages the energy distribution network in a large part of The Netherlands. They transport gas, electricity and heath from energy producers to households and other users. In order to support their operations, Liander has lots of data, but is not allowed to use this data in applications due to legal limitations. Nevertheless, this data could be used by third parties in applications, for instance to facilitate the transition towards a more sustainable energy future. Therefore, Liander would like to open some of their data to support such innovations.

In the following sections we describe the nine steps identified above in detail. The Liander dataset serves as a running example to illustrate each of the steps. In this way we show how Liander, or any other data owner, can turn their data into linked (open) data and publish them as a starting point for integration with other data sources.

Note that we generally only describe one way of performing the given steps, i.e., we give one recipe. There are often several alternatives possible, using different tools or different methods to achieve the same result.

[10] See http://www.pilod.nl/wiki/BoekTNO/stappenplan

[11] http://www.w3.org/2011/gld/wiki/Linked_Data_Cookbook

[12] Heath & Bizer. Linked Data: Evolving the Web into a Global Data Space. Retrieved from: http://linkeddatabook.com/book

5.2 Step 1: Select data

The first step is to select the data that you want to publish and determine if any restrictions apply that could prohibit you to publish the data. The reasons for publishing data as open or linked open data can be very diverse: from compliance to data laws to following competitors to realizing new unexpected value from data. Once an organization has decided to open up some of its datasets, either to a specified community or the general public, a data manager or other responsible person needs to decided which datasets they actually want to publish. This can be done by setting up a data strategy or by inventorying the datasets of an organization and deciding, based on the goals to be reached by open data, which datasets are interesting to be published. Hereby it is important not to be too selective, as others might be able to use the data for new innovative applications that one does not think of in the first place.

Once datasets have been selected for publication one needs to analyze if and how the datasets can actually be opened up or if publication restrictions apply for (parts of) the data. The following aspects should be taken into account when making a decision about opening data: ownership, privacy, economic, data quality and technical format. The open data decision tree[13] shown in Figure 4 can be used to structurally analyze datasets for possible constraints. The decision model works as follows. If a certain constraint to data sharing is present in a given situation, the next step is to analyze if the constraint can be overcome by an intervention (the light green curved arrow in Figure 4). For example, when a privacy constraint occurs, anonymizing by filtering or aggregation by combining a dataset into a single record, are potential interventions. Interventions are usually of a technical nature, but also include organizational mechanisms. When no suitable intervention can be identified the dataset cannot be shared. This means that the five constraints can be interpreted as knock-out criteria. The data can only be opened if all identified constraints in all categories can be overcome by interventions. This is shown by the arrow on the right-hand side of Figure 4.

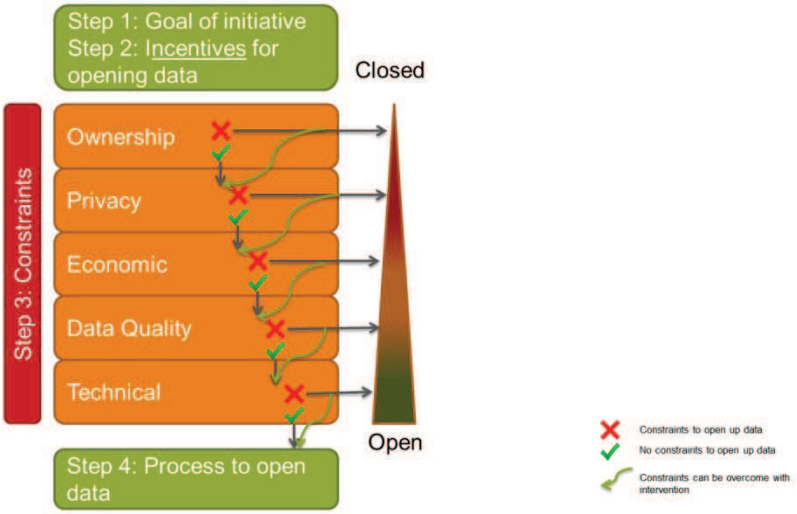


Figure 4 – ***Decision model for opening up data***

[13] Eckartz, Hofman, Van Veenstra, A Decision Model for Data Sharing, 13th international IFIP EGOV conference 2014

We will now provide some exemplary questions per category to give a bit more information on the level of detail of the analysis:

- Ownership: Is the person entitled to decide about opening the data positive about it?
- Privacy: Does the data source contain information that can be traced to individual persons or companies?
- Economic: Is the business case of opening the data positive? (here several business case options can be compared, e.g. the costs and benefits of several technical opening formats)
- Data Quality: Is the data validated to be correct?
- Technical: Is the data published as raw data? Can the data be published in an open format?

In many cases raw data is appreciated and also might overcome some responsibility issues. The decision model should be applied both on a dataset level as well as on individual data properties and even data values of a dataset. It should be noted that the decision model that is presented in this section, often serves as an example rather than a definite set of issues that needs to be addressed. While the categories remain more or less the same, for every use case new issues can be added to the categories.
Once the datasets to be published and necessary interventions have been identified, the data publisher can use this information to formulate his data publication strategy and continue the process of preparing the data for publication (Step 2).

5.2.1 Running example – Step 1: Select data

Liander collects data on energy consumption and (local) production, e.g. through the use of energy meters. Liander would like to publish this data in order to:

- Be transparent as a public utility company
- Stimulate open innovation
- Gain insight into data needs
- Improve data quality by receiving feedback

The table below shows a snapshot of the raw metering data. It contains the electricity consumption at 15 minute intervals of a number of households with smart meters.

datetime	Klant 1	Klant 2	Klant 3	Klant 4	Klant 5	Klant 6
1-5-2012 0:00	105	80		57	44	23
1-5-2012 0:15	92	67		58	48	37
1-5-2012 0:30	86	33		58	24	34
1-5-2012 0:45	100	50		129	16	21
1-5-2012 1:00	100	33		503	15	27
1-5-2012 1:15	82	46		87	16	18
1-5-2012 1:30	59	40		63	19	27
1-5-2012 1:45	84	58		57	50	24
1-5-2012 2:00	61	80		65	40	40
1-5-2012 2:15	60	61		19	17	17
1-5-2012 2:30	74	75		20	15	29
1-5-2012 2:45	69	51		19	15	22
1-5-2012 3:00	56	42		19	14	26
1-5-2012 3:15	84	55		19	38	18

Although this data can be interesting for data consumers, e.g., to visualize energy consumption for individual households at different periods during a day, there are several issues with this data and publication will have to be restricted in a number of ways.

Firstly, this data is subject to data protection laws. It is personal data and publication will violate the privacy of the households concerned. Therefore, it cannot be published as is.

Secondly, Liander is providing a commercial service based on this data to large energy consumers. Publishing the data as open data would cannibalize one of their own revenue streams.

Thirdly, the quality of the data varies a lot. Households with smart meters may provide measurements at 15 minute intervals, but not all households have smart meters yet. In the worst case, for households without smart meters, meter readings are only validated once every three years. And even for households with smart meters readings are sometimes received only once a quarter.

In order to deal with these issues, the data is restricted in the following ways:

- The data quality is standardized. Rather than publishing actual meter readings at regular intervals, Liander only publishes the estimated, standardized annual usage. This value is recalibrated once a quarter using recent readings, but will be published only once a year.
- Commercially sensitive data is removed from the dataset, i.e., only energy usage of private households, the so called small users, is published.
- The data is anonymized. Rather than publishing the annual usage for each individual household, the annual usage is aggregated for all households in the geographical area determined by the 6-digit postcode. If there are less than ten households in one postcode area, the annual usage of two or more consecutive postcode areas are aggregated.

5.3 Step 2: Prepare the data

Once the data has been selected, the next step is to prepare this data for publication. The following sub-steps have to be considered:

a) Obtain access to the data source or data extracts, or create a new dataset in a way that can be replicated.
b) Obtain a copy of the logical model of the database to be used in the data modelling in Step 3.
c) Perform a data quality assessment to get insights into the data quality of the dataset.
d) Use data cleansing were needed to improve the data quality, e.g., by removing outdated, obsolete and irrelevant data.
e) Implement technical interventions, such anonymizing sensitive data elements or the integration of datasets identified when selecting the data.

Different tools can be used for these steps ranging from general purpose spreadsheet and database tools to dedicated data cleansing tools (see 5.3.2).

5.3.1 Data quality assessment

It is of utmost importance to check the quality of the dataset as a whole as well as the individual data elements before moving on in the process of opening data. Based on an extensive literature review done earlier we propose to check the following quality aspects identified by Nousak and Phelps[14] and Knight and Burn[15] for each data instance:

- Validity, the extent to which information is correct and reliable.
- Completeness, the extent to which information is not missing (e.g. all required data elements are given).
- Consistency, the extent to which information is presented in the same format and compatible with previous data, and free from variation and contradiction based on the condition of another data element.
- Uniqueness, if the data element is unique, meaning that there are no duplicate values.
- Timeliness, the extent to which the information is sufficiently up-to-date.
- Accuracy, if the data element values are properly assigned and free of error. And describing the closeness between a value v and a value v' considered as the correct representation of the reality that v aims to portray.
- Preciseness, if the data element is used only for its intended purpose, i.e., the degree to which the data characteristics are well understood and correctly utilized.

The data owner might decide to improve the data quality of data elements that show low quality with respect to one or more of the quality aspects. However, this is not required. No matter if the quality of a dataset is high or low, it is always valuable to describe the actual data quality of the dataset in the metadata, e.g. in terms of the data quality aspects described above. This allows users of the dataset to judge if the quality is good enough for their purpose.

5.3.2 Data Cleansing

Where needed the data quality of data elements can be improved by data cleansing. Datasets are similar to raw material: they first have to be refined before they become useful. Data cleaning (also referred to as cleansing or scrubbing) describes the process of: fixing errors, transforming and homogenizing formats, aligning inconsistencies in data and metadata, removing duplicate and redundant information, adding lacking information, and making sure the information is up-to-date. One concrete example is the deletion of white spaces and empty cells in a dataset and the identification of missing data. In the data mining literature quite some research has been done on data cleansing, especially in the field of anomaly detection. We will not dive into this field of research in this report but only mention some practical tips: the tools to actually do data cleansing.
A wide range of cleansing tools (both commercial as well as open source) can be found on the web. These are a few examples:

1. Open Source:
    - Spreadsheet software such as Calc from Libre Office: http://schoolofdata.org/handbook/recipes/cleaning-data-with-spreadsheets/
    - Open Refine (formerly Google Refine) with LOD extensions: https://github.com/sparkica/LODRefine

[14] Nousak, P., & Phelps, R. (2002). A Scorecard approach to improving Data Quality. Paper presented at the SUGI27. Retrieved from http://www2.sas.com/proceedings/sugi27/p158-27.pdf
[15] Knight, S. A., & Burn, J. (2005). Developing a framework for assessing information quality on the World Wide Web. Informing Science, 8, 159-172.

- Data Cleaner: http://datacleaner.org/

2. Commercial Tools:
    - Trifacta.com based on Wrangler: http://vis.stanford.edu/wrangler/
    - Data Ladder: http://dataladder.com/

An illustrative usage of OpenRefine can be found in chapter C3 “How to use LODrefine?” by Paul Hermans in the book “Pilot Linked Open Data: Deel 2 – De Verdieping”[16].

5.3.3 Running example – Step 2: Prepare the data

In this step the data is extracted from the core information systems of Liander, filtered, anonymized, aggregated and documented.
The selected dataset is a combination of data from different database tables in Liander’s information systems. There is a table containing the standardized annual usage of gas and electricity per household and tables with metadata about the connections, the installed meters and the customers. This data cannot be published as is, because we need to apply the restrictions defined above. Therefore, we create a new database (table) with a copy of the required data using SQL. When copying the data we can already filter for small users with a SQL WHERE-clause. Below is a snapshot of the resulting table. Note that the column names and the data itself are in Dutch. However, even if the data was in English, it contains all sorts of codes, abbreviations and special terminology. Documentation is required to understand the data.

EAN	POST-CODE	HUIS-NUMMER	STAAT-NAAM	WOON-PLAATS	LAND	PRODUCT	RICHTING	TYPE	SJV_ NORMAAL	SJV_ LAAG	TYPE_ METER
55581503	7231JT	24	't Spiker	WARNSVELD	NL	ELK	CMB	3x25	3586		DUN
17866103	7231JT	24	't Spiker	WARNSVELD	NL	GAS	CMB	G4	1574		DUN
8662423	7522AV	27	Minister Kuyperplein	ENSCHEDE	NL	ELK	LVR	3x25	1399	1499	CVN
15126093	7522AV	27	Minister Kuyperplein	ENSCHEDE	NL	GAS	LVR	G4	2662		CVN

Now, the data still needs to be anonymized by aggregating and averaging the energy usage for both electricity (ELK) and gas (GAS) per postcode area and removing the EAN codes and house numbers that identify individual consumers. Because we aggregate we cannot simply copy the values in each column. The service direction (RICHTING), for example, can have any of three values: LVR (Levering = consumption), TLV (Teruglevering = production) or CMB (Combination). In this case, it is decided to replace this by the percentage of entries with value “LVR”. For the connection TYPE, we copy the value that occurs most within an area and add a column indicating the percentage of households with this type of connection. The energy usage values, SJV_NORMAAL and DJV_LAAG are added and then averaged over the postcode area. Finally, a new column is added with the number of connections within an area. All these operations should be clearly documented to enable users to interpret the data correctly. A snapshot of the resulting dataset is provided in the table below.

STRAAT-NAAM	POST-CODE VAN	POST-CODE TOT	WOON-PLAATS	LAND	PRODUCT	Aantal	%Richting	%TYPE	TYPE	SJV	%Laag	%Slimme Meter
Rijksweg A44	1000AA	1011AB	NIEUW VENNEP	NL	ELK	31	100	29	3x25	16245	38,71	16,13
De Ruyterkade	1011AC	1011AC	AMSTERDAM	NL	ELK	32	100	31	3x25	11433	28,13	15,63
't Spiker	7231JS	7231JT	WARNSVELD	NL	ELK	24	75	54	3x25	3764	41,67	0
't Spiker	7231JS	7231JT	WARNSVELD	NL	GAS	20	100	100	G4	2615	0	0
't Spiker	7231JV	7231JV	WARNSVELD	NL	ELK	16	100	88	1x25	2425	0	0
't Spiker	7231JV	7231JV	WARNSVELD	NL	GAS	16	100	100	G4	1626	0	0

[16] http://www.pilod.nl/wiki/Boek/Hermans

Finally, it is useful to export the data from the database table to a more open format, such as a comma separated (CSV) file.

5.4 Step 3: Model the data

Once access to the data has been ensured and the data quality has been described and improved where necessary, the next step is to model the data. Modeling linked data is often a very time consuming task, but it makes the data more widely understandable and usable both within and across organizations. When creating linked data, one should employ proper engineering practices in order to create datasets of high quality that possibly make use of existing resources on the Web rather than creating them from scratch, and express the intended semantics correctly so that others (both machines and humans) can properly understand and reuse the datasets being built to extend the Web of data[17]. In this respect, the following process should be followed for producing high quality linked datasets.

The term linking data is sometimes confusingly used, particularly because one can create "links" in multiple ways. It is also important to notice that "links" between datasets can be done at several steps in the process of data modeling. Different types of "links" can be made: ontology links and data links. We will highlight three different options to link datasets during the process of modeling data using *italics*.

1. Make a conceptual model of the data by defining concepts and their relationships and properties. You can use the logical data model obtained when preparing the data as input for this step.
   1.1. Sketch or draw the objects on a white board (or similar) and draw lines to express how they are related to each other. Assign one or more data elements to each object. This kind of *data element linking (Option 1)* will be discussed in more detail in Step 9.
   1.2. Look for real world objects of interest such as people, places, things and locations.
   Use common sense to decide whether or not to make links.
2. Investigate how others are already describing similar or related data in vocabularies.
   2.1. *Reuse existing, standardized and widely adopted vocabularies (Option 2)* as much as possible to facilitate data merging and reuse. Since others use the same vocabularies, your dataset will be linked to the dataset of others with the vocabulary as bridge. This is very important to increase the usability of the dataset (see section 5.4.1 for more in depth information).
   2.2. If reuse is not possible *use your own or create a new vocabulary (Option 3)* according to the best practices for modelling linked data. Linked data is created by linking your own vocabulary via ontology-links to existing vocabularies (see section 5.4.2 for more information).
3. Formalize the model and your vocabulary, preferably in the Web Ontology Language OWL, alternatively in RDFS or SKOS.

While modelling you should put aside immediate needs of any application and be sure to test the assumptions in the schema with subject matter experts familiar with the data.

It is not necessary to define the ultimate model of the data at once. More the contrary; the philosophy of linked data offers you the possibility to start without modelling the data, do it later or not, or go for a step-by step approach. Tools that help you model the data include Topbraid Composer and Protégé.

[17] www.w3.org/2011/gld/wiki/Linked_Data_Cookbook and www.w3.org/TR/ld-bp/

We will now elaborate in more detail on two types of ontology linking: the reuse of standard vocabularies and the creation of new vocabularies.

5.4.1 Reuse of standard vocabularies

The underlying idea of adopting existing vocabularies is to enable an optimal reuse of the work that has already been done and acknowledged on linked data. In this way, it is not only possible to create datasets based on proved solutions easier and faster than starting from scratch, but also contribute to a proper expansion of the Web of data, by clearly linking new datasets to existing and commonly adopted resources using the same semantics across the different datasets.
When reusing existing vocabularies, it is important to first take the time to look for what is currently available. For example, there are several vocabularies for linked data suggested by the W3C[18]. An overview of vocabularies is provided in Table 1.

Table 1 – Overview of standard vocabularies

Name	Prefix	Namespace URI	Describes
Basic Geo	geo:	http://www.w3.org/2003/01/geo/	geo positioning
Bibliographic Ontology	bibo:	http://purl.org/ontology/bibo/	Bibliographies
Creative Commons Rights Expression Language	cc:	http://creativecommons.org/ns	Licenses
Data Catalog Vocabulary	dcat:	http://www.w3.org/TR/vocab-dcat/	Datasets
Data Cube Vocabulary	qb:	http://www.w3.org/TR/vocab-data-cube/	Multi-dimensional data
Description of a Project	doap:	http://usefulinc.com/ns/?doap	Projects
Dublin Core Metadata Initiative	dct:	http://dublincore.org/documents/dcmi-terms/	Publications
Friend-of-a-Friend	foaf:	http://xmlns.com/foaf/spec/	People
GeoNames Ontology	gn:	http://www.geonames.org/ontology/ontology_v2.2.1.rdf	Locations
Good Relations	gr:	http://purl.org/goodrelations/v1	Products
Object Reuse and Exchange	ore:	http://www.openarchives.org/ore/	Resource maps
Organization Ontology	org:	http://www.w3.org/TR/vocab-org/	Organizations
Semantically-Interlinked Online Communities	sioc:	http://rdfs.org/sioc/spec/	Online Communities
vCard	vcard:	http://w3.org/TR/vcard-rdf/	Business cards
Vocabulary of Interlinked Datasets	void:	http://www.w3.org/2001/sw/interest/void/	Vocabularies

[18] www.w3.org/2011/gld/wiki/Linked_Data_Cookbook#Step_3_Re-use_Vocabularies_Whenever_Possible

- The WGS84 for geopositioning defines terms for latitude, longitude and other information about spatially-located things, using WGS84 as a reference datum[19].
- The Bibliographic Ontology (BIBO) provides the main concepts and properties for describing citations and bibliographic references, such as quotes, books, articles, etc.
- The Creative Commons Rights Expression Language defines terms for describing copyright licenses in RDF.
- The Data Catalog Vocabulary (DCAT) facilitates interoperability between data catalogs published on the Web. By using DCAT to describe datasets in data catalogs, publishers increase discoverability and enable applications easily to consume metadata from multiple catalogs.
- The Data Cube Vocabulary provides a means to publish multi-dimensional data, such as statistics, on the web.
- The Description of a Project (DOAP) vocabulary describes software projects with particular emphasis on Open Source projects.
- The Dublin Core Metadata Initiative (DCMI) Metadata Terms defines general metadata attributes for published works including title, creator, date, subject and publisher.
- The Friend-of-a-Friend (FOAF) vocabulary defines terms for describing people, their activities (collaboration) and their relations to other people and objects.
- The GeoNames Ontology is a geographical database containing over 10 million geographical names.
- The Good Relations is an ontology for E-commerce that defines terms for describing products, price, and company data.
- The Object Reuse and Exchange vocabulary is used by libraries and media publishers for the description and exchange of aggregations of Web resources that may combine distributed resources with multiple media types including text, images, data, and video.
- The Organization Ontology supports the publishing of organizational information across a number of domains, as Linked Data.
- The Semantically-Interlinked Online Communities vocabulary (SIOC) is designed for developers to describe information about an online community sites, such as users, posts and forums.
- The vCard vocabulary is an older but popular address book format that has since been ported to RDF and includes the basics of what is needed for representing addresses internationally.
- The Vocabulary of Interlinked Datasets (VoID) defines key metadata about RDF datasets. It is intended as a bridge between the publishers and users of RDF data, with applications ranging from data discovery to cataloging and archiving of datasets. One should always publish a VoID description of your vocabulary so others can reuse it.

In addition, it is possible to find existing vocabularies using dedicated search engines for the Semantic Web (e.g., Watson, Sindice, Semantic Web Search Engine, Swoogle, and Schemapedia), and other platforms, such as the LOV directory, Prefix.cc, Bioportal for the biological domain, and the European Commission's Joinup platform.

[19] http://en.wikipedia.org/wiki/World_Geodetic_System

Once a potential vocabulary has been identified, one should critically review this vocabulary according to the following criteria in order to decide whether to adopt it or not:

- *The vocabulary must be well documented* by using label and comment annotations. Moreover, a human-readable page should be available to describe classes and properties, possibly including use cases that show the vocabulary's applicability.
- *The vocabulary should be self-descriptive* by using at least a label, a definition and a comment for each class or property.
- *The vocabulary should be described in several native human languages*, especially when publishing government data, by using labels, definitions and comments in the government's official language(s) and at least in English.
- *The vocabulary should be used by other datasets* to guarantee acknowledgement in the LOD community and promote reuse of high quality contributions.
- *The vocabulary should be accessible for a long period* by providing some guarantee of maintenance over a specified period, ideally indefinitely.
- *The vocabulary should be published by a trusted group or organization* since anyone can create a vocabulary, it is always better to check whether there is a person, group or authoritative organization that is responsible for publishing and maintaining the vocabulary.
- *The vocabulary should have a persistent URL* to guarantee persistent access to the server hosting the vocabulary.
- *The vocabulary should provide a versioning policy* to guarantee that the publisher will address compatibility of versions over time. Major changes to the vocabularies should be reflected in the documentation.

Another good source before starting with defining new vocabularies is the http://sameas.org website, which contains a collection of triples that contain the "owl:sameas" construct. The "owl:sameas" construct is also useful when you find out that a term in someone else's vocabulary means the same thing in your own vocabulary. In that situation it is not necessary to change your vocabulary but only to publish a triple with owl:sameas. Note: owl:sameas should only be used for relating two pure synonyms, i.e., two terms that refer to exactly the same concept. If the relationship is less precise, rdfs:subClassOf may be used to relate one concept to a slightly more general concept.

5.4.2 Creation of new vocabularies

Sometimes there are no existing vocabularies available for a specific domain, or they do not comply with the review criteria described above, therefore, one may decide to create a new vocabulary. In this case, it is necessary to use best engineering practices for modelling linked data in order to guarantee quality by design, and use proper advertising strategies to stimulate the adoption of the vocabulary in the LOD community.
The main guidelines for creating a new vocabulary can be summarized in the following criteria[20]:

- *Define a clean and stable URI* using a careful URI naming strategy. More details on these strategies can be found in the Step 4 and the Linked Data Cookbook[21].
- *Choose the proper language to model your vocabulary* depending on your purpose. For example, SKOS[22] is suitable to model lists of terms, such as controlled vocabularies, taxonomies or thesauri. RDF[23] allows to represent data

[20] www.w3.org/TR/ld-bp/#VOCABULARIES
[21] www.w3.org/2011/gld/wiki/Linked_Data_Cookbook#Step_2_Name_Things_with_URIs and www.w3.org/TR/ld-bp/#HTTP-URIS
[22] http://www.w3.org/TR/2009/NOTE-skos-primer-20090818/
[23] http://www.w3.org/TR/2004/REC-rdf-primer-20040210/

models as objects (web resources) and relations in terms of (subject, predicate, object) triples, while RDF Schema[24] extends RDF for describing properties and classes of RDF-based resources. OWL[25] provides more primitives to describe properties and classes , and axioms to constrain the usage of these properties and classes, allowing a higher degree of semantic reasoning.

- *Make your vocabulary self-descriptive* using at least a label, a definition and a comment for each class or property that is defined.
- *Provide documentation*, not only machine readable, but also human readable, together with basic metadata that allow others to correctly understand and properly reuse your vocabulary. In this respect, a best practice consists in publishing a VoID description to describe key metadata of the schema or dataset being created, as described by W3C[26].
- *Provide a versioning policy* to show commitment to possible users that you as publisher will take care of changes in the vocabulary and adapt both human and machine readable versions of the vocabulary accordingly.
- *Publish the vocabulary at a stable URI using an open license* following best practices for publishing and advertising, as described in the Linked Data Cookbook[27].

More guidelines on the process of creating a new vocabulary can be found in this Blog[28]. Setting up a new domain vocabulary has much in common with what traditionally was called defining a new semantic data standard for an industry domain. Both are a group process, and both results, the vocabulary and the semantic standard need to be maintained and updated. See BOMOS[29] for an overview and detailed description of all activities needed for the management and maintenance of open standards. One might even argue that some semantic standards will be published as vocabularies in the future.
Ontology links can be specified using rdfs:subClassOf or owl:equivalentClass relations in the ontology itself, or in a separate mapping ontology that imports both the ontology of the original dataset and the ontologies one wants to map to. Such mappings can be exploited by a reasoner attached to the triple store to derive additional links between the data and the more general ontologies. In this way, a user that does not know the original ontology can query the dataset using the more general ontologies.

[24] http://www.w3.org/TR/2004/REC-rdf-schema-20040210/
[25] http://www.w3.org/TR/2009/REC-owl2-primer-20091027/
[26] www.w3.org/2001/sw/interest/void/
[27] www.w3.org/2011/gld/wiki/Linked_Data_Cookbook#Step_6_Specify_an_Appropriate_License, www.w3.org/2011/gld/wiki/Linked_Data_Cookbook#Step_7_Host_Linked_Data_Publicly_and_Announce_it.21 and www.w3.org/TR/ld-bp/#ANNOUNCE
[28] http://richard.cyganiak.de/blog/2011/03/creating-an-rdf-vocabulary/
[29] NoiV and TNO, Beheer- en OntwikkelModel voor Open Standaarden (BOMOS), 2010 (https://www.forumstandaardisatie.nl/fileadmin/os/publicaties/Bomos_-_deel_1.pdf and https://www.forumstandaardisatie.nl/fileadmin/os/publicaties/Handreiking_BOMOS_deel_2.pdf)

5.4.3 Running example – Step 3: Model the data

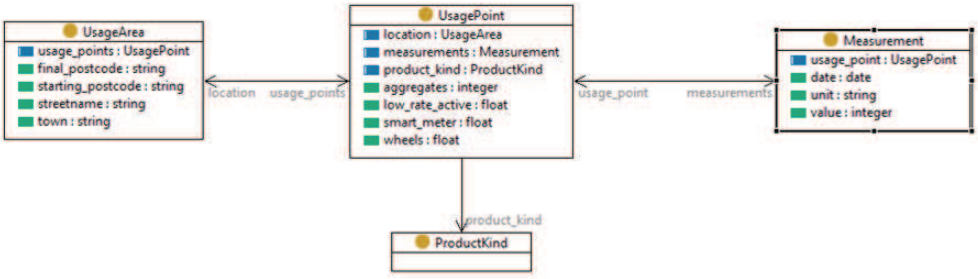
In this step we create a vocabulary that describes the Liander dataset.
The starting point for this step is the documentation of the database. Part of the documentation (in Dutch) is presented in the table below. This gives us an idea about the meaning of the data and is our starting point for a conceptual model of the data.

Variabele/veld	Mogelijke waarden	Toelichting
[A.07] Postcode	4 cijfers 2 letters	Twee kolommen: van en naar
[A.10] Straatnaam	Naam	Bij verschillende postcodes bij “van” en” naar”: de straatnaam van de eerste postcode
[A.11] Woonplaats	Naam	Naam van de woonplaats
[A.17] Productsoort	ELK of GAS	De energiesoort waarover het SJV gegeven wordt: ELK= elektriciteit, GAS = aardgas
Aantal aansluitingen	getal	Het aantal aansluitingen in het betreffende postcodegebied voor het betreffende product.
[A.33] SJV	kWh of m3	Gemiddeld Standaardjaarverbruik, waarbij voor aansluitingen met een normaal én laagtarief SJV het totale SJV wordt meegenomen in de middeling, zonder decimalen. Het standaardjaarverbruik is het verwachte jaarverbruik van een afnemer op een netaansluiting bij gestandaardiseerde condities en op basis van een genormaliseerd jaar.
		Wanneer een aansluiting uit bedrijf is blijft het laatste SJV staan tot het moment waarop de aansluiting weer in bedrijf genomen wordt.
[A.34] SJV laag tarief	%	Percentage van de aansluitingen dat een laagtarief SJV heeft, oftewel een dag/nachttarief geactiveerd heeft.
[M.102] Type meter	%	Percentage slimme meters. Het betreft alle typen slimme meters, zowel de op afstand schakelbare als de niet op afstand schakelbare meters (resp codes DUS en DUN)
[M.115] Aantal telwielen	getal	Gemiddeld aantal telwielen van de meters

We identify the following concepts:

- Usage Area: a geographical area defined by a range of consecutive postcodes in which energy is consumed and/or produced.
- Usage Point: a (possibly virtual) connection point at which energy is transferred from the network to and from (a set of) energy prosumer(s).
- Measurement: amount of energy consumption or production measured or predicted for a certain date/time interval in a certain unit of measure.
- Product Kind: kind of energy product being delivered and consumed at a certain usage point. Currently either electricity or gas.

The figure below shows how these concepts, their properties and relations could be modelled as an ontology.



In this case we have opted to create our own basic ontology for this dataset. This has the advantage that it can be created to closely match the structure of the dataset, and that it does not require knowledge of other external ontologies (faster time to publication). The disadvantage is that the data is less accessible, because it is unlikely that data consumers will be familiar with this ontology. That is the reason why best practice prescribes to reuse existing vocabularies as much as possible to model your data. The nice thing about the Semantic Web, however, is that such links to other ontologies can always be added at a later stage as we will show later in Step 9 (linking the data).
Once you have modelled your data by either re-using existing vocabularies or by creating new vocabularies the next step is to define a naming structure for your dataset which makes it uniquely identifiable.

5.5 Step 4: Defining a naming structure – Name things with URIs

This step will provide guidelines on how to use URIs (Unified Resource Identifiers) in order to identify your data. One of the principles of linked data is that each object and relation is uniquely identifiable with a URI, both on set and element level. The use of persistent and unique identifiers, such as URIs, URLs and DOIs is an important quality aspect. As a linked data publisher you should therefore give careful consideration to the selection and consistent application of your URI strategy, i.e., the scheme used for assigning URIs to data elements. We propose to use national and international best practices whenever possible.
The W3C lists the following as best-practice[30]:

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

Based on more practical experience the SEMIC project identified rules that should be taken into account[31].

- Take data changes over time into account
- Use clean, stable URIs
- Use natural keys
- Follow the pattern
- Re-use existing identifiers
- Link multiple representations
- Implement 303 redirects for real-world objects
- Use a dedicated service
- Avoid stating ownership
- Avoid version numbers
- Avoid using auto-increment

[30] http://www.w3.org/DesignIssues/LinkedData.html
[31] https://joinup.ec.europa.eu/community/semic/document/10-rules-persistent-uris

- Avoid query strings
- Avoid file extensions

In the Netherlands the working group: “URI-strategy” (as part of the PiLOD project[32]) has formulated a number of starting points that should be observed upon drawing up a URI strategy[33]:

- Link up with international best-practices. You can go faster on your own, but you will go farther by working together. By linking up with international developments, you benefit from solutions that are devised on a global scale. In addition, European regulations are becoming increasingly important to the Dutch government.
- Link up with existing developments. The strategy concerns many parties and systems and cannot be implemented all at once as something new. And so it is wise to assess what is already taking place in the sphere of standardization and authentic registrations and to reuse that as much as possible.
- Anticipate deviating systems. Even if systems are developed that, for whatever reason, do not observe the national strategy, it must still be possible to link to these systems.
- Keep it as simple as possible, but not simpler. If the approach is too complex, then the strategy will not be adequately applied, or not applied at all. If the approach is too simple, then the strategy will not yield sufficient results.

The “URI Strategy” working group is working towards a Dutch national URI strategy. They currently propose the following structure:
http://{domain}/{type}/{concept}/{reference}
where

- {domain} should be an internet domain (URL) that the data owner controls where the data will be published and the URIs can be dereferenced. Optionally, this includes a path within that domain: {domain} = {internet domain}/{path}.
- {type} is either ‘id’ if the URI is an identifier of an object (individual/instance), ‘doc’ if it refers to the metadata about an object, or ‘def’ if it refers to the definition of a concept in an ontology.
- {concept} is the name of the concept to which the object identified by the URI refers.
- {reference} should be a unique number or code identifying the object within the namespace. It can be a name or a number, as long as they are unique and not too long.

5.5.1 Running example – Step 4: Define a naming scheme

In this step we define a scheme for assigning URIs to the Liander dataset. We deviate slightly from the proposed Dutch national URI strategy. Our URIs have the following structure:

http://{domain}/{type}/{dataset|ontology}/{concept}/{reference}

where

- {domain} should be an internet domain (URL) that the data owner controls where the data will be published and the URIs can be dereferenced. In this case, ‘data.liander.nl’ seems appropriate.
- {type} is either ‘id’ if the URI is an identifier of an object (individual/instance), ‘doc’ if it refers to the metadata about an object, or ‘def’ if it refers to the definition of a concept in an ontology.

[32] www.pilod.nl
[33] http://www.pilod.nl/wiki/Boek/URI-strategie

- {dataset|ontology} is either the short name of the dataset or of the ontology. In our case, we use the name 'liander for both. We have made this addition for pragmatic reasons. We intend to publish this dataset alongside other datasets and ontologies on the same server. Therefore, we need a way to distinguish datasets and ontologies.
- {concept} is the name of the concept to which the object identified by the URI refers.
- {reference} should be a unique number or code identifying the object within the namespace. It can be a name or a number, as long as they are unique and not too long. For usage areas, we will use the concatenation of the starting and final postcodes as reference; for usage points, the postcodes followed by 'E' (for Electricity) or 'G' (for Gas); for Measurements, the date and the reference of the usage point.

Concepts in an ontology are referenced by their name. Therefore, the {reference} is left empty in this case. And rather than a '/' the hash (#) is used to separate the {concept} from the rest of the URI. This is a best practice for naming ontology concepts that is supported by most ontology editors.
Following this scheme we get for example the following URIs:
<http://data.liander.nl/def/liander> for the Liander ontology.
<http://data.liander.nl/def/liander#UsageArea> for the concept of Usage Area within the Liander ontology.
<http://data.liander.nl/id/liander/UsageArea/7231JS7231JT> for the Usage Area that starts at postcode 7231JS and ends at postcode 7231JT.

5.6 Step 5: Convert the data to RDF

Once you have a schema that you are satisfied with, the next step is to convert the source data into a Linked Data representation or serialization. In this step the data is converted to RDF triples while applying the naming scheme defined in Step 4. RDF triples may be stored in a file or in a specialized database called a triple store.
Before converting your data to RDF you need to decide how you want to publish the data. Do you want to publish the data as a web service that can be queried? We advise that you do not choose one single serialization but multiple. Potential serializations for RDF are turtle (human readable), RDF/XML, N3, RDFa (in HTML), and the lately becoming very popular JSON-LD (for JavaScript Developers).
Conversion approaches fall into three categories:

- Automatic conversion, sometimes called triplication
- Partial scripted conversion
- Modeling by human and subject matter experts, followed by scripted conversion

There are different tools that can be used to do this conversion:

- LODRefine, an extension of OpenRefine
- RDF Translator: http://rdf-translator.appspot.com/
- Spyder (http://www.revelytix.com/content/spyder)
- Ontop (http://ontop.inf.unibz.it/)

The Simile project[34] made an attempt in proving a directory of tools for converting various data formats into RDF. A similar list can be found by W3C[35]. A tutorial on converting

[34] http://simile.mit.edu/wiki/RDFizers
[35] http://www.w3.org/wiki/ConverterToRdf

relational data into RDF format is provided by Oracle[36]. For an easy approach to transform spreadsheets to RDF, have a look at one of the books[37] published by the PiLOD project containing a step by step approach based on the OpenRefine tool, including screenshots of the tool. If you are interested in expressing customized mappings from relational databases to RDF datasets you should check out the R2RML language[38]. Once you have converted your data to RDF the next step is to make sure that you have some governance structure in place to maintain and manage your data.

5.6.1 Running example – Step 5: Convert the data

In step 2 we have created a table with the data. In this step we convert the data to RDF triples and apply the naming scheme defined in step 4. There are different tools that can be used to do this conversion. In this case we have used LODRefine, an extension of OpenRefine. We load the table with the data in LODRefine. Then we use the RDF plugin to define RDF skeletons based on the ontology we defined in step 3. For example, the following expression is used to create URIs for Usage Areas:

```
"http://data.liander.nl/id/liander/UsageArea/" + cells['POSTCODE_VAN'].value + cells['POSTCODE_TOT].value
```

We can also specify the rdf:type to be liander:UsageArea. And the various datatype properties to take their value from the appropriate columns in the table, e.g., that the liander:town property should get the value from the WOONPLAATS cell.
An extract of the resulting triples is shown below.

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix liander: <http://data.liander.nl/def/liander> .


<http://data.liander.nl/id/liander/UsageArea/1011AC1011AC>                    a
<http://data.liander.nl/def/liander#UsageArea> ;
       <http://data.liander.nl/def/liander#streetname> "De Ruyterkade" ;
       <http://data.liander.nl/def/liander#starting_postcode> "1011AC" ;
       <http://data.liander.nl/def/liander#final_postcode> "1011AC" ;
       <http://data.liander.nl/def/liander#town> "AMSTERDAM" .

<http://data.liander.nl/id/liander/UsagePoint/1011AC1011ACE>                  a
<http://data.liander.nl/def/liander#UsagePoint> ;
       <http://data.liander.nl/def/liander#low_rate_active> "28.13" ;
       <http://data.liander.nl/def/liander#smart_meter> "15.63" ;
       <http://data.liander.nl/def/liander#wheels> "1.1" ;
       <http://data.liander.nl/def/liander#location>
<http://data.liander.nl/id/liander/UsageArea/1011AC1011AC> ;
       <http://data.liander.nl/def/liander#product_kind>
<http://data.liander.nl/def/liander#Electricity> .

<http://data.liander.nl/id/liander/UsagePoint/1011AC1011ACG>                  a
<http://data.liander.nl/def/liander#UsagePoint> ;
       <http://data.liander.nl/def/liander#low_rate_active> "0" ;
       <http://data.liander.nl/def/liander#smart_meter> "18.18" ;
       <http://data.liander.nl/def/liander#location>
<http://data.liander.nl/id/liander/UsageArea/1011AC1011AC> ;
       <http://data.liander.nl/def/liander#product_kind>
<http://data.liander.nl/def/liander#Gas> .
```

36

http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/11g/r1/prod/datamgmt/relational_into_rdf/relational_data_into_rdf_format_otn.htm

37 http://www.pilod.nl/wiki/Boek/Hermans

38 http://www.w3.org/TR/r2rml/

5.7 Step 6: Organize Governance

When publishing Linked (open) data it is of utmost importance to not only think of the technical aspects described in this cookbook but also take governance aspects into account. Things like a licensing structure, how maintenance will be deployed and other governance aspects need to be thought of and agreed upon before the data is published. We will not cover these issues in this cookbook but refer to BOMOD[39], a method for governance of open data. Organizing governance will be a step that takes a lot of throughput time, as it involves agreement with the involved stakeholders. Therefore, it is important to start early on in the data publication process and do this step in parallel with the other steps.

5.7.1 Running example – Step 6: Organize Governance

In this step, data governance has to be organized for the Liander dataset. As said before, Liander would like to publish this data in order to:

- Be transparent as a public utility company
- Stimulate open innovation
- Gain insight into data needs
- Improve data quality by receiving feedback

This is the vision from which the governance structure and data publishing strategy are derived. At the moment, the governance structure consists of one person who is responsible for the open data initiative at Liander. The data is published with a liberal license, i.e. Creative Commons with Attribution. The dataset is actively promoted through a dedicated website and at events and challenges, in order to build a user community. Users can contact Liander with questions about the dataset through a dedicated e-mail address. However, there is no official support available for users.

5.8 Step 7: Add metadata

While following this guide, and especially in the previous step, the organization of governance, you will realize that metadata about your dataset is of crucial importance. In this step we will introduce three levels of metadata that you can use when describing your dataset.

In order to make the dataset self-describing and thus support the re-usage of data, extra information about the data needs to be added to the data by the data supplier. Self-describing data suggests that information about the encodings used for each representation is provided explicitly within the representation. Such data about data is called metadata and includes information about the data origin, the data production date and for which applications the data can be used. Metadata that describes the process of data development is also referred to as provenance[40]. Provenance gives an indication of the reliability of the data. Another metadata aspect interesting for reusing data is information about the usability of the data. It might be interesting for data users to learn about successful applications of other data users. Information about data usability is also very valuable for Linked Data. It can provide a good indication of the potential success of similar applications in the future. Metadata can be added by simply adding triples to the RDF version of the dataset obtained in Step 5 describing facts about the dataset.

Linked Data published on the Web should be as self-describing as possible in order to make it easier for clients to understand and use the data. Important aspects of self-descriptiveness are making vocabulary terms de-referenceable according to the best

[39] http://publications.tno.nl/publication/34616703/ATAycW/eckartz-2015-bomod.pdf

[40] Freire, J., Koop, D., & Moreau, L. (2008). Second International Provenance and Annotation Workshop. Paper presented at the IPAW 2008, Salt Lake City, Utah.

practices described in Publishing RDF Vocabularies, using terms from common vocabularies and providing vocabulary mappings for proprietary vocabulary terms[41].
We structure this section using the three levels of metadata described by CKAN[42]:

- Level 1, basic metadata,
- Level 2 minimal metadata and
- Level 3 complete metadata.

We extend the aspects mentioned in that classification with aspects from our quality model developed during earlier research. The Dutch government has published a list[43] with elements that metadata of datasets published at data.overheid.nl should include. Most of the elements are compulsory. The elements fall in four categories: context, data source, characteristics, involved organizations. We add these elements to the tables provided for the three levels of metadata using their original identifier from data.overheid.nl.

5.8.1 Level 1: Basic Metadata

The basic aspects that should be included in all metadata descriptions are shown in Table 2.

Table 2 – Level 1 Basic Metadata aspects

Dimension	Definition	Source	Metrics
A1. Name/ Title	Unique name or ID for the dataset	ODI[44], Dutch Government (DS_02)	Text
A3. Publisher/ Author	Name of the publishing organization and/or person. Including contact information. (e.g. email address)	ODI, Dutch Government (B_03 & B_07)	# 5 from Zaveri et al[45]
DS_01 Identifier	Unique name of the dataset that is used in URLs and for identification	Dutch Government	Uniqueness of the identifier
A4. Location/ URL	Unique link to the (online) place/ website where the dataset can be accessed or downloaded. This might also include Links that enable alternative access to the data set.	ODI, Dutch Government (DB_01)	Availability of the location link
License	Information about the license structure of the dataset		Link to license

5.8.2 Level 2: Minimal Metadata

Level 2 metadata should include the basic aspects from Level 1 and in addition the aspects shown in Table 3 which are based on our extensive literature review performed during earlier research. In order to measure these aspects we make, where possible use of the metrics defined by Zaveri et al.

[41] CKAN: http://validator.lod-cloud.net/levels.html
[42] http://validator.lod-cloud.net/levels.html
[43] https://data.overheid.nl/node/609
[44] https://certificates.theodi.org
[45] Zaveri, Amrapali, et al. "Quality assessment methodologies for linked open data." Submitted to Semantic Web Journal (2013).

Table 3 – Level 2 Minimal Metadata aspects

Dimension	Definition	Source	Metrics
A5. Release	One-off vs. ongoing release. Single vs. a set or series of related datasets. Is it a service or API for accessing data?	ODI	Frequency of release
A6. Potential Use	What can users do with it? What sort of question can it answer? Topic tags can be used to structure this aspect.	LATC[46]	Topic Tags
A7. Compliance	To which regulations/rules does it comply?	ISO 9126[47]	Link to regulations
A8. Production Date	The date the dataset has been created. This might also include information about the last modification date or version of the data set.	Ehling & Körner[48]	Date and time
B3. Format: Open Format	Information about the format in which the dataset is provided, especially focusing on if the data is available in a standard open format.	ODI, Dutch Governmen t (DB_03)	Format used (JSON, XML, RDF, CSV etc.)
B4. Kind of data (Type of data)	Unstructured (human readable data), statistical data (counts, percentages), Geo data (points, boundaries), other structured data.	ODI, Dutch Governmen t (DS_03)	# 15-17 from Zaveri et al
DS_04 Language	The language that is used in the dataset	Dutch Governmen t	Natural language used
DS_07 Spatial	Describes the area/ territory covered by the dataset	Dutch Governmen t	Province, national, international
B7. Semantics	Understandability: Extend to which data are clear without ambiguity and easily comprehended	ODI, Knight & Burn[49]	# 20-21 from Zaveri et al
B8. Data model	Is there a data model describing the objects represented by a computer system together with		Availability of data model

[46] https://docs.google.com/document/d/150dJSMZk5W5ucF23hGj62DaoKtTk9qeaEPBN_VCCihI/edit?pli=1

[47] ISO/IEC. (2003). ISO/IEC 9126-2 Software engineering - Product quality - Part 2: External metrics.

[48] Ehling, M., & Körner, T. (Eds.). (2007). Handbook on Data Quality Assessment Methods and Tools. Wiesbaden.

[49] Knight, S. A., & Burn, J. (2005). Developing a framework for assessing information quality on the World Wide Web. Informing Science, 8, 159-172.

	their properties and relationships.		
B9. Links	Coherent links to other datasets.	Ehling & Körner	# 22-23 from Zaveri et al
B10. Size	The size of the datasets, e.g. the amount of triples, or megabytes.	Dutch Government (DB_04)	# 24-25 from Zaveri et al
B11. Concise	Extend to which information is compactly represented without being overwhelming.	Knight & Burn	#26-29 from Zaveri et al
B12. Complete	Is the datasets complete or are there certain parts missing?	Knight & Burn	# 30-33 from Zaveri et al
B13. Believability	Extent to which dataset is regarded as true and credible.	Knight & Burn	meta-information about the identity of information provider, # 34 from Zaveri et al
B14. Reputation	Extent to which dataset is highly regarded in terms of source or content.	Knight & Burn	# 35-36 from Zaveri et al

The last four metadata dimensions are less objective than the other dimensions. Data publishers might first need to get input from users, such as subjective judgments, before they are able to provide metadata information about these aspects for a specific dataset.

5.8.3 Level 3: Complete Metadata

Level 3 metadata should include the aspects from Level 1 and 2 and in addition the aspects discussed in this section. Table 4 shows the aspects which are based on our extensive literature review performed during earlier research. In order to measure these aspects we again make use of the metrics defined by Zaveri et al.

Table 4 – Complete Metadata aspects

Dimension	Definition	Source	Metrics
A2. Description	A short description of the dataset, including its full name and original intended usage	ODI, Dutch Government (DS_08 & DS_10)	Text; presence of a tag to identify LOD
DS_11 Website with explanation	Website that gives explanation about the dataset and provides guidelines on how to use the dataset	Dutch Government	Link to website
A9. Provenance	Understand the issues of data creation, transformation, and copying.	Freire et al[50]	# 1-11 from Zaveri et al
B6. Use of vocabularies / codelists/ schemas	Are custom vocabularies or schemas which say what columns or properties the data contains used?	ODI	# 18-19 from Zaveri et al

[50] Freire, J., Koop, D., & Moreau, L. (2008). Second International Provenance and Annotation Workshop. Paper presented at the IPAW 2008, Salt Lake City, Utah.

DS_12 LOD stars	Degree to which the dataset fulfills the linked open data criteria measured in stars (Tim Berners-Lee)	Dutch Government	5 star system (Tim Berners-Lee)
DS_06 Temporal	Describes the time period covered by the dataset	Dutch Government	Timespan

5.8.4 Provenance

Provenance is one kind of metadata which tracks the steps by which the data was derived and can provide significant value addition in data intensive scenarios[51]. Data provenance, describes the derivation history of a data product starting from its original sources. It is a collective term for all aspects related to traceability, responsibility, auditability, accountability and accuracy of data. Provenance gives an important indication about the reliability of the data and is very important for the re-use of Linked Data. The linking and combination of different data sets, which might even result in editing data sets, has huge effects on the reliability of the new data sets. Recently, the PROV vocabulary got standardized for Linked Data by W3C.[52] The vocabulary can be used to express provenance metadata in Linked Data.

- *W3C PROV standard*: Provenance is information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness. The goal of PROV is to enable the wide publication and interchange of provenance on the Web and other information systems. PROV enables one to represent and interchange provenance information using widely available formats such as RDF and XML. In addition, it provides definitions for accessing provenance information, validating it, and mapping to Dublin Core[53].

Dublin Core defines provenance as: “A statement of any changes in ownership and custody of the resource since its creation that are significant for its authenticity, integrity, and interpretation”[54]. A collection of literature about provenance, structured according to three dimensions (content, management and use) is provided by W3C[55]. Open Provenance Vision[56] is a vision of a set of architectural guidelines to support provenance inter-operability, consisting of controlled vocabulary, serialization formats and APIs.
The simplest way to use PROV is through one of the many applications, such as ProvStore[57], that support it. Questions that one needs to answer when describing provenance include the following:

- Who created that content (author/attribution)?
- Was the content ever manipulated, if so by what processes/entities?
- Who is providing that content (repository)?
- What is the timeliness of that content?
- Can any of the answers to these questions be verified (for example by e-signatures)?

[51] Simmhan, Yogesh L., Beth Plale, and Dennis Gannon. "A survey of data provenance techniques." Computer Science Department, Indiana University, Bloomington IN 47405 (2005).
[52] http://www.w3.org/TR/prov-overview
[53] http://www.w3.org/TR/2013/NOTE-prov-overview-20130430/
[54] http://dublincore.org/documents/2012/06/14/dcmi-terms/#terms-provenance
[55] http://www.w3.org/2005/Incubator/prov/wiki/Mendeley_%26_BibBase_Collection
[56] http://openprovenance.org/
[57] https://provenance.ecs.soton.ac.uk/store/

Metadata and especially provenance are essential when publishing datasets to ensure re-usability and value creation.

5.8.5 Running example – Step 7: Add metadata

In this step we make the dataset self-describing by adding metadata. We can do this by simply adding triples to the RDF version of the dataset obtained in step 5 that describe facts about the dataset itself. Below are some examples of basic metadata for the Liander dataset.

```
<http://data.liander.nl/id/liander> rdf:type rdfs:DataSet
<http://data.liander.nl/id/liander> dcterms:modified "2014-05-27"
<http://data.liander.nl/id/liander> rdfs:label "Liander energy usage dataset."
<http://data.liander.nl/id/liander>  rdfs:comment  "Standardized  annual  energy
usage of small users in the Liander domain aggregated per postcode area."
<http://data.liander.nl/id/liander>                                  dcterms:creator
<http://nl.dbpedia.org/resource/Alliander>
<http://data.liander.nl/id/liander> dcterms:date "2014-03-08"
<http://data.liander.nl/id/liander> dcterms:publisher <http://www.liander.nl/>
<http://data.liander.nl/id/liander>                                  rdfs:vocabulary
<http://data.liander.nl/def/liander>
```

5.9 Step 8: Publish the data – Announce it!

In this step the dataset is made available on the Internet. There are different options for publishing the dataset. A good practice is to make use of several options, so that data users have a choice and can select the method that best suits their purposes.

One option is to publish the dataset as a flat file. Often used syntaxes are: RDF/XML (.rdf) and Turtle (.ttl). LODRefine, the tool recommended in Step 5 to convert the data to RDF can export to both formats. The resulting files can simply be put on a webserver.

Another, more advanced, way to make the data available is to store it in a triple store and serve it through a SPARQL-endpoint. If you provide a SPARQL Endpoint you allow others to query your linked data/ metadata. You can provide links to the dataset download files (dumps) or the SPARQL endpoint[58]. Download files relieve your server from strong crawling/querying activity for people interested in bulk loading (e.g. indexing) your dataset. SPARQL endpoints allow people to select a subset of their interest through a query.

If you have a SPARQL endpoint please provide information, such as the location of the SPARQL endpoint in the metadata of your dataset. It is also important that you publish your metadata on a central data broker to give it more visibility and increase the reuse of your dataset. The metadata quality dimension important for this step of the guideline is accessibility, as defined in Table 5.

Table 5 – Metadata quality dimensions for publishing data

Dimension	Definition	Source	Metrics
B1. Accessibility	Extent to which information is available or easily retrievable. Extent to which data are easily found and linked to (API).	Knight & Burn, ODI	# 12-14 from Zaveri et al
B2. Format: Machine-readable	If the data is machine readable.	ODI	

[58] http://validator.lod-cloud.net/levels.html

Consumers of Linked Data do not have the luxury of talking to a database administrator who could help them understand a schema. Therefore, a best practice for publishing a Linked Dataset is to make it "self-describing" e.g. by adding metadata as described in Step 7. Self-describing data suggests that information about the encodings used for each representation is provided explicitly within the representation.
Several frameworks/ tools are available for hosting RDF data. One of them is Sesame[59], an open source framework for storing and querying RDF data. Sesame can be installed on any appropriate server. A web interface, the OpenRDF Workbench, enables you to create a new RDF repository and upload the RDF triples created in Step 5 from a file. Once the data is uploaded to Sesame, users can query the dataset with SPARQL, the standard query language for linked data.
Other options to publish your data include the following platforms:

- Swirrl (http://www.swirrl.com/publishmydata): Commercial software as a service publishing platform.
- LOD Cloud (http://datahub.io/group/lodcloud): This group catalogs datasets that are available on the Web as Linked Data and contain data links pointing to other Linked Datasets.
- Open Data overheid (https://data.overheid.nl/): The Dutch National Open Data platform where governmental organizations can register their open datasets.
- City-SDK (http://citysdk.waag.org/data): A web service offering unified and direct access to open data from government, commercial and crowd sources alike. Cities can open up their data using CitySDK.
- Platform Linked Data Nederland (http://www.platformlinkeddata.nl): Platform that offers organizations to publish their linked open data
- Open data Nederland (http://opendatanederland.org/): A registry listing all the open datasets of the Netherlands on one single website.
- CKAN (Ckan.org): A powerful data management system that makes data accessible by providing tools to streamline publishing, sharing, finding and using data.

5.9.1 Running example – Step 8: Publish the data

In this step we make the Liander dataset available on the Internet. We have different options for publishing the dataset. A good practice is to use multiple ways, so that data users have a choice and can select the method that best suits their purposes.
Firstly, we publish the dataset as a flat file. We do this in two often used syntaxes, i.e., RDF/XML (.rdf) and Turtle (.ttl). LODRefine, the tool we used to convert the data to RDF in step 5 can export to both formats. The resulting files can simply be put on a webserver at data.liander.nl.
A better way to make the data available and to store it in a triple store and make it accessible through a SPARQL-endpoint. In this case we use Sesame, an open source framework for storing and querying RDF data (see http://openrdf.org). Sesame can be installed on an appropriate server, e.g., data.liander.nl. A web interface, the OpenRDF Workbench (shown in Figure 5), enables us to create a new RDF repository and upload the RDF triples we created in step 5 from a file.

[59] http://openrdf.org

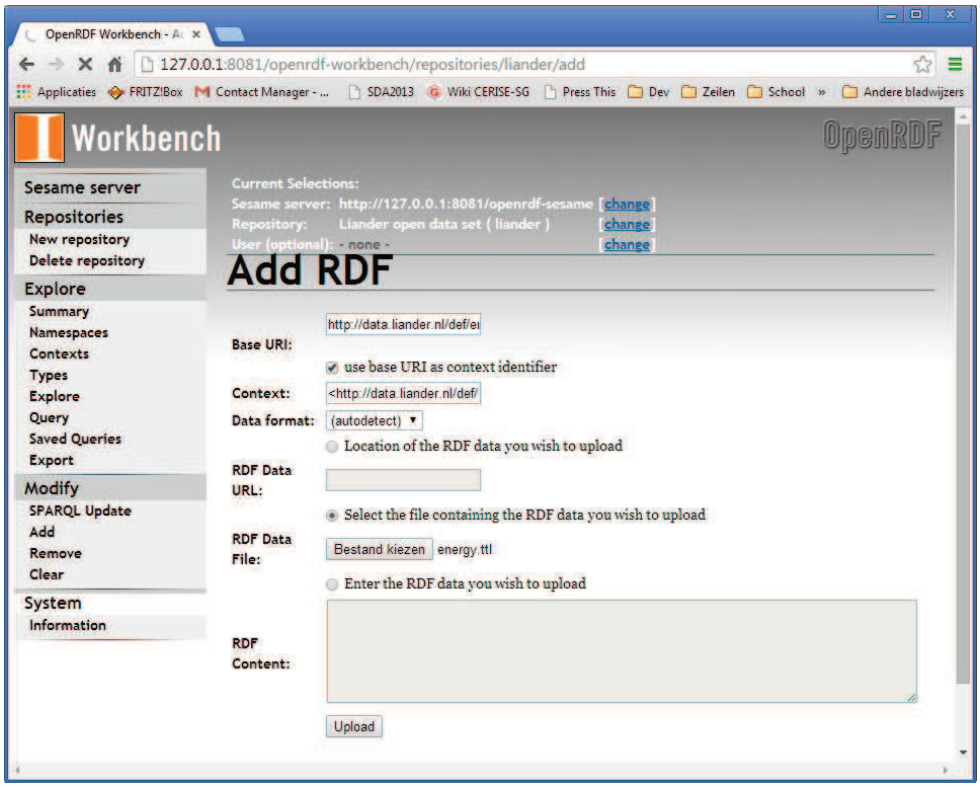


Figure 5 – Screenshot of Sesame workbench to upload data to the triple store

Once the data is uploaded to Sesame, users can query the dataset with SPARQL, the standard query language for linked data.

5.10 Step 9: Link the data

A final and optional step is to link the dataset to other datasets to be able to provide more context to the data. Linked Data, unlike other data formatting and publication approaches, provides a simple mechanism for combining data from multiple sources across the Web.

Several guidelines on how to combine datasets can be found on the web providing step-by-step guidelines[60]. Different types of links can be made: ontology links and data links. While links to ontologies have been already made in Step 3, we will now describe data links. *Data links:* The data itself can also be linked to other available linked datasets. This may be useful to provide more context to the data. Consider for example a dataset that includes addresses which contain a reference to a town. It is likely that more information about these towns is already available on the web. DBpedia, for example, the linked data version of Wikipedia, usually has an entry for each town. One could add triples to the original dataset to link the addresses to the DBpedia entry providing more details about the town.

Another option is to create a new dataset that contains the links between your and other datasets. This way it can be done afterwards, but it can also be done by others that link your data to other datasets.

[60] http://wifo5-03.informatik.uni-mannheim.de/bizer/pub/LinkedDataTutorial/#whichvocabs

5.10.1 Running example – Step 9: Link the data

A final and optional step is to link the Liander data to other datasets. The data is already linked to a vocabulary, in this case to our own Liander ontology, i.e., each object has a type. This ontology can be linked to other, better known ontologies. For example, our ontology talks about addresses, postcodes and units of measure, some concepts that have been described elsewhere. Addresses and postcodes, for example, appear in the W3C Location vocabulary[61]. Such links can be specified using rdfs:subClassOf or owl:equivalentClass relations in the Liander ontology itself, or in a separate mapping ontology that imports the Liander ontology and the ontologies we map to. Such mappings can be exploited by a reasoner attached to the triple store to derive additional links between the data and the more general ontologies. In this way, a user that does not know the Liander ontology can query the dataset using the more general ontologies.

For example, we could assert that a UsageArea is a Location according to the W3C Location vocabulary as follows:

liander:UsageArea rdfs:subClassOf dcterms:Location.

The data itself can also be linked to other available linked datasets. This may be useful to provide more context to the data. Consider for example the addresses in our dataset. They contain a reference to a town. We could add triples to our dataset to link our Usage Areas to the DBpedia entry providing more information about the town in which the Usage Area lies.

The link from a Usage Area to the DBpedia entry for the city could be made as follows:

<http://data.liander.nl/id/liander/UsageArea/1012CM1012CN> dbpedia-owl:isPartOf dbpedia:Amsterdam

This is just one triple relating a specific Usage Area to the DBpedia entry for Amsterdam. Of course, it is impractical to add such links by hand because our dataset contains tens of thousands of Usage Areas. A semantic link tool, such as SiLK, is useful to semi-automate the linking of data[62].

[61] See http://www.w3.org/ns/locn.html

[62] http://www.pilod.nl/wiki/Boek/Gueret-Linking

6 Generation of profile from UML to OWL

The EBIF vocabulary as developed by the CERISE-SG project and described in deliverable D4.1 is expressed in RDF and specified by the IEC CIM model in UML. In order to formalize the EBIF2CIM conceptual mappings, such that they could be used for automating data integration, it was necessary to generate a RDF/OWL version of the IEC CIM model that was originally expressed in UML/XMI (and the Enterprise Architect proprietary format). Since the project was interested only in some parts of the IEC CIM model, we first specified a IEC CIM-based CERISE profile for metering and location using the CIMTool, and then transformed this profile from UML/XMI to RDF/OWL, also using CIMTool. Section 6.1 elaborates on the transformation process using the CIMTool, while Section 6.2 presents our findings during this process.

6.1 Generation of CERISE-CIM metering profile from UML to OWL

In order to transform a UML model to OWL using the CIMTool, it is necessary to use the corresponding XMI (XML Metadata Interchange) serialization of the UML model as the source model for the transformation. The CIMTool translates XMI to OWL by first extracting UML information, then creating analogous OWL definitions. The analogy between UML and OWL is close, especially in the fundamental concepts of classes and associations (properties in OWL). The CIMTool strategy is to glean UML from the XMI. An XML parser recognises constructs of interest while ignoring surrounding syntax. When a construct is recognised, corresponding statements are inserted into an OWL/RDF model[63]. The main UML concepts are translated to OWL as follows:

- A UML class translates as an OWL Class.
- A UML association translates as two ObjectProperty's in OWL, each the inverse of another. In other words, an association end or role translates as a single an ObjectProperty.
- A UML enumeration translates to either an OWL Class plus individuals or and OWL Class enumerated by a oneOf definition. The former creates an open set that can be extended. The latter a closed set.

For the specification of the cerise-metering profile we have used CIMTool and followed these steps:

1. Create a new “CIMTool Project” with name “cim-cerise” and select the “CIM16.xmi”as the file to import as initial schema,
2. Create a new “CIMTool Profile”, assign a namespace URI (in our case “http://ontology.tno.nl/cerise/cim-profile”) and give a profile name (in our case “cim-profile”)
3. Move the classes and properties of interest from the source “CIM model” to the target “cim-profile”
4. Select the type of profile that should be built, in our case we use the builder for simple-owl
5. Save the file and the “cim-profile.owl” file will appear in the Profile folder of the project workspace, including the selected ontology types (in our case the “simple-owl”.

[63] See http://wiki.cimtool.org/UMLOWL.html for more details on the transformation

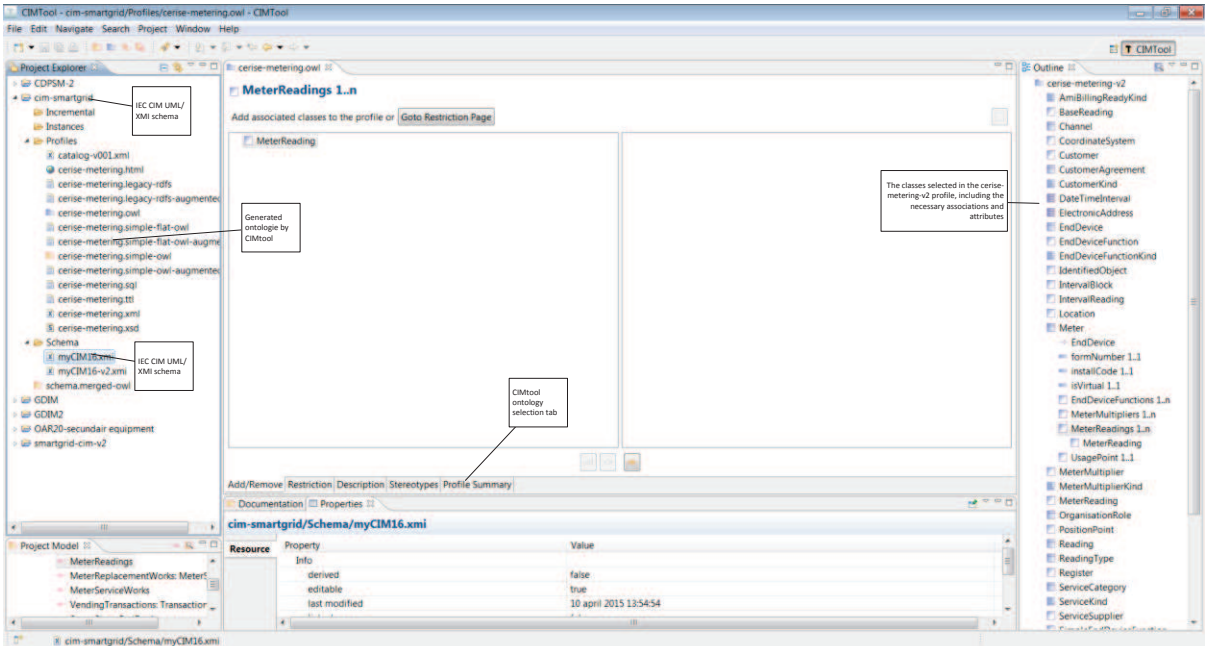


Figure 6 CIMTool GUI generating cerise-metering ontology

Notice that several profiles can be generated by the CIMTool:

- Builder for simple-flat-owl
- Builder for simple-flat-owl-augmented
- Builder for simple-owl
- Builder for simple-owl-augmented
- Builder for ttl

6.2 Findings and remarks during the IEC CIM ontology processing

We first tried to build a turtle (ttl) profile, which is our preferred format (more compact and clear than RDF/XML) and also used by TopBraid Composer that is our semantic modelling environment of choice. However, the result was not a suitable turtle ontology, nor compliant with the UML to OWL transformation rules mentioned above that the CIMTool is supposed to use (e.g., no OWL object properties were created corresponding to UML associations). Therefore, we tried to build a simple-owl profile and this time we obtained a suitable ontology with UML associations properly mapped to OWL object properties. However, we noticed an incorrect mapping of cardinalities from UML to OWL. For example:

- the UML association *Meter [0..1]* was mapped into the OWL property *MeterReading.Meter exactly 1*, while we would expect it to be mapped to *MeterReading.Meter* ***max 1***
- the UML association *Readings [0..*]* was mapped into the OWL property *MeterReading.Readings min 1*, while while we would expect it to be mapped to *MeterReading.Readings* ***min 0***

We checked whether building one of the other profiles (i.e., simple-flat-owl, simple-owl-augmented and simple-flat-owl-augmented) would produce a correct result concerning the cardinality translation, but the same OWL cardinality was generated also when building these profiles. We then concluded that this is a limitation of the CIMTool. The

way we could overcome the limitation was to edit afterwards by hand the cardinalities in the generated OWL profile[64].
As an additional remark, it was not clear to us nor explained in the CIMTool documentation what are the differences between the different OWL profiles that can be generated (i.e., simple-owl, simple-flat-owl, simple-owl-augmented and simple-flat-owl-augmented). We noticed that the simpe-owl profile was suitable for our needs and therefore used this as profile of choice when generating OWL with the CIMTool. Notice that the generated "cerise-cim-metering-profile.owl" can be directly opened with Protégé but not with TopBraid Composer, which no longer supports the ".owl" extension. Therefore, the "cerise-cim-metering-profile.owl" first needs to be saved as turtle (.ttl) or RDF/XML (.rdf) in Protégé and only afterwards can be opened with TopBraid Composer.
The resulting CERISE CIM metering profile is available online as follows:

- http://ontology.tno.nl/cerise/cim-metering (html documentation)
- http://ontology.tno.nl/cerise/cim-metering.ttl (turtle version)

[64] A request for solution was posted on the CIMtool support wiki.

7 Using Linked Data in web applications

This chapter provides the evaluation of the test bed that was developed by the project to be able to test Linked Data for the exchange of information.

7.1 Introduction

This chapter deals with consumption of harmonized data that is made available using the recommendations that resulted from the project. It is possible to envisage different possibilities of data consumption: Data could serve as input for isolated (not shared on the web) data silos, or it could serve as input for desktop applications (applications that run on an operating system like Windows of Linux). But the most obvious and rewarding type of end user application will be web applications, applications that run in a web browser. In that case both application and data run on the same platform, the world wide web. And there is a distinct overall trend in application development to move from desktop to web. A trend that is supported by increasing capabilities of web browsers and an economy of sharing ideas and software.

In the CERISE-SG project several web applications have been developed, with the purpose of demonstrating benefits of provision of harmonized data to end users. These web applications are accessible from the project website, http://www.cerise-project.nl/. The (javascript) source code of the application is made available as Open Source software. The applications are not meant to serve as parts of any production system, but parts or ideas could be used to create more robust applications.

The following paragraphs contain general advice on Linked Data based web application development, based on experiences in CERISE-SG.

7.2 Degrees of freedom

An important difference between a Linked Data based web application and a traditional web application is that the former can and should make less assumptions about the data it is going to use. A traditional web application that works with data is usually part of an isolated stack and usually linked to a single dataset that may contain changing data, but which has a fixed format, fixed semantics and a fixed location.

A Linked Data based web application *could* work like that, but because Linked Data are self-descriptive and interlinked there are possibilities for allowing more unknowns to exist in the application. Which could make the application more versatile and powerful.

For example, a Linked Data web application does not need to be preconfigured to work with a particular dataset. It could make use of metadata and data catalogues on the web to discover data sets that it could work with, or which match the needs of the user. The application could then try to find out more about the data set, for example its size, its data types, its semantics, before user interaction and the actual retrieval of data.

However, this does not mean that a Linked Data application should be designed to work with any data it can find on the web. That would effectively mean building a web browser for data, a daunting task because there would be a lot of unknowns. In the case of CERISE-SG the demonstrators were constrained to work with energy data in the Netherlands, so geographical and thematic constrains were used. Other constraints can be applied according to the purpose of an application.

7.3 Data formats

A Linked Data web application works with data that use the RDF model. But that kind of data can come in many formats. When the application requests data as an RDF graph, JSON-LD is the preferred format. JSON (Javascript Object Notation) is a data format that is easy to use in javascript web applications. JSON-LD is a type of JSON that can be used to convey intact RDF graphs. It has several forms. Of these, *flattened form* is the easiest to process in a web application. By means of content negotiation it is possible to request JSON-LD format as response to a data request, but it is not possible to request a certain form. Therefore the jsonld.js library (https://github.com/digitalbazaar/jsonld.js) is used to transform each incoming JSON-LD dataset into flattened form.

Not all data that are gathered from the web are graph data, though. For retrieving selections of data from large datasets the demonstrators make use of SPARQL, a query language for RDF. A common query in SPARQL is the SELECT query. It does not return graph data, but tabular data. Data returned from a SPARQL SELECT query are not meant to be interpreted, they have known semantics and relationships. Instead, they are useful for direct visualization, for instance in a table or on a map.

Responses from a SPARQL SELECT query can also have different formats. In the CERISE-SG demonstrators the CSV (Comma Separated Values) format was used, because is it compact and straightforward.

7.4 Data retrieval interfaces (APIs)

Two basic ways of retrieving data are used in the CERISE-SG demonstrators. Firstly, direct dereferencing of URI's can be used to get the data that a URI identifies. This method is useful for obtaining relatively small sets of data, in particular vocabularies and metadata. Often, dereferencing indeed is the only way of obtaining a vocabulary or a set of metadata.

For selecting data from larger datasets, SPARQL is used. It is an expressive language, similar to SQL. But unlike SQL, it can be used directly on the web. SPARQL can be used to select just the data that are needed for the application, but it does come at a cost. SPARQL query can be hard to process on the server, and hard to compose by a client. In the CERISE-SG clients, a set of preconfigured SPARQL queries was used, with some variable elements (e.g. temporal or spatial constraints could be based on user input).

Between the straightforward method of dereferencing and the expressive but complex method of SPARQL other means of data retrieval can be envisaged, but were not used in the project. For continued work it would be good to look at interesting new initiatives for interaction with datasets: SPIN (SPARQL Inferencing Notation, see http://spinrdf.org/) can be used to define SPARQL functions, and so could take away some complexity in client side SPARQL queries. The Linked Data Platform (http://www.w3.org/TR/ldp/) defines simple read-write access to Linked Data using HTTP methods (GET, POST, PUT, PATCH, DELETE,..). The RDF Data Shapes Working Group (http://www.w3.org/2014/data-shapes/charter) is working on a specification that can be used to define structures in datasets that can enhance usability and allow easier data validation. Lastly, Linked Data Fragments (http://linkeddatafragments.org/) offers a way of relieving data servers of some of the burden of query processing.

7.5 Data discovery

As stated before, the Linked Data approach allows for some freedom with regard to binding data sets. Data sets to work with do not need to be preconfigured in the application, but can be found on-the-fly. A good starting point for finding appropriate data sets is a data catalog. The Data Catalog Vocabulary (DCAT, see http://www.w3.org/TR/vocab-dcat/) prescribes a way how data catalogs can be structured and interlinked. This recommendation was used for the data catalog of CERISE-SG (http://lod.geodan.nl/cerisesg/datasets/). This catalog can be used to find the datasets in the project, and to obtain the metadata describing each dataset. The metadata can and should contain all data that a client needs to work with the dataset. For instance, the metadata can describe the temporal and spatial extent of a dataset, it can make known when the dataset was updated and is can give the URI of a SPARQL endpoint that can be used to query the data.

7.6 Data retrieval

After data sets have been discovered, or have been bound in another way, at some point the application will need to download data. As explained above, data retrieval is done in two ways in the CERISE-SG clients.

Dereferencing is used for vocabularies and metadata. The preferred data format is JSON-LD, flattened form. In order to make this format always available, a server side component (based on node.js in this case) is used to transform data to JSON-LD before they are handled by the web application. The server side component assumes that at least RDF/XML format is available. Open Source libraries can be used to transform from RDF/XML to JSON-LD.

The server side component also solves another problem: Cross Origin Resource Sharing CORS). In some situations it is not possible to have a javascript application that is hosted in one domain request data from a host in another domain. This problem is circumvented by letting a single server (a CORS proxy) handle all requests for the web application. This server is hosted on the same platform as the web application.

The other type of data retrieval is SPARQL. In general it is used to get the data that are displayed to the end user. SPARQL has many possibilities of filtering the data to get just the subset that is needed, based on configuration of the application or on user settings. It is even possible to request data from different servers in one query, using a federated query. In one of the CERISE-SG clients a federated query is used to geocode addresses: for the addresses in a dataset with address based energy consumption data the addresses are used to look up coordinates in the BAG dataset. This operation can be executed in a single SPARQL query. But a drawback of using SPARQL is that it can put a heavy load on servers, especially when there are much data to process and when queries are not optimized. The CERISE-SG clients use a small set of SPARQL queries that were optimized during development time, so the developer has some idea of performance per query. Still, SPARQL queries can take some seconds to complete so it is recommendable to handle data retrieval asynchronously and not have a data request block the application. For simple tabular data streaming can be used, it allows displaying the data as they come in, without having to wait for the entire transaction to complete.

There are ways of reducing stress on servers and clients caused by high data volumes. One is caching of data. Data that have been requested once from a server can be temporarily stored at some intermediate location that can quickly be accessed by the web application. This intermediate location could be the memory of storage of the machine the

web application is running on, it could be the machine that hosts the web application or it could be some other platform that is easy to access for the web application and that has dedicated resources for caching data. Caching does require having an idea of the storage life of the data. For example, if a data set is known to be updated on the first Monday of each month, this means that it is known when the cache in no longer valid. Again, this shows that provision of extensive metadata is vital for data consumption.

Another way of reducing the burden of high data volumes is aggregating data to an appropriate level. Raw data could come at a resolution that is high compared to what a web application can visualize. For example, when looking at energy consumption in a city a client could request energy consumption data per address. But that would mean that a lot of data will have to be downloaded, and visualizing all those data will be challenging. It would make more sense to request the data grouped by postal code or neighbourhood. Fortunately, SPARQL has many possibilities for aggregating data.

7.7 Visualisation and user interaction

When data become available to the web application, the data can be displayed. Because data are available in raw un-interpreted form, there are many visualization opportunities in the web application. When turning data in to information for the application user, freely available libraries can be used that offer diverse types of visualization. For example, the same dataset could be presented in a table or in a diagram. And tables and diagrams themselves can be formatted or styled in different ways. Should the data contain a temporal component (e.g. timestamps or dates), a time slider could be offered to the user. Similarly, if the data contain a geographic component the data could be plotted on map.

A library for data visualization that is used in the CERISE-SG applications is D3.js. D3 stands for Data Driven Documents. It can be used to visualize datasets in many different and interactive ways, including maps. Google Charts is another library that can be used to visualize data interactively in various ways in a web application.

Visualization libraries and other libraries can be used to enable a user to interact with data. A user might want to change views (e.g. switch from map view to chart view) or reconfigure a view (e.g. change a classification, zoom in on a map, change axis scale in a chart). When the application is designed well, user interaction does not mean new data queries have to be issued each time, which means that such changes can be handled smoothly. But the need for new data could arise if the user wants to visualize another (related) data set or want to change the constraints used in the original query. In those cases it would be preferable to minimize wait times by using cached data or by streaming data.

A Linked Data based web application should be able to explain to the user what the data mean. The fact that the meaning of data is not necessarily known at the time the application is developed makes this an interesting challenge. Semantics do not need to be hard-coded, but can be retrieved online, from the vocabularies that are referenced in the data. A good vocabulary will offer short labels as well as more extensive description of the concepts it defines. The labels are good candidates for direct display in the web application, for instance as column headers or in map legends. Description can be used to provide some more context, for example as a tooltip. Ideally, human readable annotation in vocabularies comes in different languages, allowing semantic information to be displayed in the user's preferred language.

That said, it could be worthwhile to be prepared for the occurrence of very general semantics. For example, the predicate `rdf:type` indicates that a resource is an instance of a class. It is a relationship that is both meaningful and ubiquitous. An application developer could therefore choose to make provisions for making such information clear to the user, without relying on on-the-fly resolution of semantics.

7.8 Conclusion

In various ways the sections above show that the quality of a Linked Data driven web application depends highly on the quality of available data. The Linked Data paradigm offers ways of provisioning high quality data, containing everything that a web application needs to function well: extensive metadata, service descriptors, multiple API's, support for multiple formats (including JSON-LD), using common vocabularies, using multilingual annotation and providing services with high performance and high availability.

But the Linked Data paradigm only encourages data providers to publish high quality data, it does not mandate it. Data publishers should be aware that effective exploitation of their data is very much dependent on data quality, and they should be open for suggestions from web application developers on how their data, and data provisioning methods can be improved.

8 Conclusions and recommendations

Linked Data is a promising solution for solving harmonization issues between different domains and different datasets (in- or outside one domain). This deliverable described the harmonization problem, identified two methods for solving the issue and described the concept of Linked Data in more detail.

Next to that three recipes where defined that can be used by the reader to:

- Publish data as Linked Data
- To generate a profile from UML to OWL
- To use Linked Data in web applications

To improve the results the recipes describes should be tested and tried by different persons to find inconsistencies, possible errors and based on that should be improved. Furthermore it would be interesting to extend this document in the future with more recipes to get a more complete overview of all the possibilities and to help people that are just starting with Linked Data.

Appendix A: Overview of tools used

This appendix provides an overview of the tools used, including a link to the webpage of the tool.

Tool	Webpage
Calc (Libre Office)	http://schoolofdata.org/handbook/recipes/cleaning-data-with-spreadsheets/
CIMTool	http://wiki.cimtool.org/index.html
Data Cleaner	http://datacleaner.org/
Data Driven Documents	http://d3js.org/
Data Ladder	http://dataladder.com/
jsonld.js library	https://github.com/digitalbazaar/jsonld.js
Ontop	http://ontop.inf.unibz.it/
Open Refine with LOD extensions	https://github.com/sparkica/LODRefine
Protégé	http://protege.stanford.edu/
RDF Translator	http://rdf-translator.appspot.com/
Topbraid Composer	http://www.topquadrant.com/tools/modeling-topbraid-composer-standard-edition/
Sesame	http://rdf4j.org/
Silk	http://silk-framework.com/
Spyder	http://www.revelytix.com/content/spyder
Trifacta.com based on Wrangler	http://vis.stanford.edu/wrangler/
Virtuoso	https://github.com/openlink/virtuoso-opensource
Webkarma	http://usc-isi-i2.github.io/karma/