Linked Open Data as the fuel for Smarter Cities

Mikel Emaldi, Jon Lázaro, Oscar Peña, Diego López-de-Ipiña, Sacha Vanhecke

No Institute Given

Abstract. In the last decade big efforts have been carried out in order to move towards the Smart City concept, from both the academic and industrial points of view, encouraging researchers and data stakeholders to find new solutions on how to cope with the huge amount of generated data.

Open Data has arisen as a way to share data in order to be consumed freely without restrictions from copyright, patents or other mechanisms of control. Nowadays Open Data is an achievable concept thanks to the World Wide Web, and has been re-defined for its application in different domains.

Regarding public administrations, the concept of Open Government has found an ally in Open Data concepts, defending citizens' right to access data, documentation and proceedings of the governments.

We propose the use of Linked Open Data, a set of best practices to publish data on the Web proposed by the W3C, in a new data life-cycle management model, allowing governments and individuals to handle better their data, easing the consumption by anybody, including both companies and third parties interested in the exploitation of the data, and citizens as end users receiving relevant curated information and reports about their city.

In summary, Linked Open Data uses the previous Openness concepts to evolve from an infrastructure thought for humans, to an architecture for the automatic consumption of big amounts of data, providing relevant and high quality data to end users with low maintenance costs. Smart data can now be achievable in smart cities.

1 Introduction

In the last decade, cities have been sensorized, protocols are constantly refined to deal with the possibilities that new hardware offers, communication networks are offered in all flavours, etc., generating lots of data that needs to be dealt with. Public administrations are not usually able to process all the data they have in an efficient way, resulting in large amounts of data going un-analized and affecting the end-users that could benefit from them.

Citizens are also being encouraged to adopt the role of linked open data providers. User-friendly Linked Data apps should allow citizens to easily contribute with new trustable data that can be linked to already existing published (more static generally) Linked Open Data provided by city councils. In addition, people-centric mobile sensing, empowered by the technology inside actual smartphones, should progress into continous people-centric enriched Linked Data. Linked Open Data also encourages the linkage to other resources described formally through structured vocabularies, allowing the discovery of related information and the possibility to make inferences, resulting in higher quality data.

Data management is becoming one of the greatest challenges of the 21^{st} century. Regarding urban growth, experts predict that global urban population will double by the year 2050, meaning that nearly 70% of the whole planet's inhabitants will be living in a major town or city.

This prediction arises the need to deal with the huge amounts of data generated by cities, enabling the possibility to manage their resources in an efficient way. The *Smart Data* term has been coined to address the data that makes itself understandable, by extracting relevant information and insights from big data and presenting the conclusions as human-friendly visualizations.

The problems of managing data are moving to a new level. It's not only a matter of caring about *more* data, but how we can use it efficiently in our processes. It's about how we can deal with increasing volumes of data (from standalone databases to real *Big Data*) and integrate them to our advantage, making it useful and digestible in order to make better decisions.

In the last few years, the *Smart city* concept has been adopted to refer to those cities aware of their citizens' life quality, worried about the efficiency and trustworthiness of the services provided by governing entities and businesses.

Smart data can help cities reach a *Smart City* status, analysing the generated data streams and providing useful information to their users: citizens, council managers, third parties, etc.

Although, efficient data lifecycle management processes need to be adopted as best practices, avoiding to convert input data in non-sense noise that can not be used to improve council's services.

Thus, our approach relies is based on an actual review of the state of the art regarding data lifecycle management, proposing our own model as a more refined approach to the existing ones. We also encourage the adoption of Linked Open Data principles to publish both the whole generated data and the processed data, in order to allow further research on the area by third parties and the development of new business models relying on public access data. Similar proposals regarding Linked Data are defended by [1].

2 Background and definitions

3 Data Life Cycle

Throughout the literature, a variety of different definitions of data life cycle models can be found. Although they have been developed for different actuation domains, we describe here some of them which we think that can be applied for generic data, independently of its original domain.

3.1 Data Documentation Initiative

The first model to be analysed is the model proposed by Data Documentation Initiative (DDI). The DDI introduced a Combined Life Cycle Model for data managing [2]. As Figure 1 shows, this model has eight elements or steps which can be summarized as follows, according to [3]:

- **Study concept.** At this stage, apart from choosing the research question and the methodology for collecting the data, the processing and analysis step of the needed data to answer the question is planned.
- **Data collection.** This model proposes different methods to collect data, like surveys, health records, statistics or Web-based collections.
- **Data processing.** At this stage, the collected data is processed to answer the proposed research question. The data may be recorded in both machine-readable and human-readable form.
- **Data archiving.** Both data and metadata should be archived to ensure long-term access to them, guaranteeing confidentiality.
- **Data distribution.** This stage involves the different ways in which data is distributed, as well as questions related to the terms of use of the used data or citation of the original sources.
- **Data discovery.** Data may be published in different manners, through publications, web-indexes, etc.
- **Data analysis.** Data can be used by others to achieve different goals.
- **Repurposing.** Data can be used outside of their original framework, restructuring or combining it to satisfy diverse purposes.

3.2 Australian National Data Service

In late 2007, the *Australian National Data Service* (ANDS) was founded with the objective of create a national data management environment. ANDS established a set of verbs, denominated *Data Sharing Verbs*, that describe the entire life cycle of the data [4]:

- **Create.** *Create* (or *collect* for disciplines with an observational focus) is about the kinds of metadata that could be collected and the tools to fulfill this collection task.
- **Store.** This *Data Sharing Verb* remarks the need for stable and web-accessible storage, taking care about the appropriate storing of data.
- **Describe.** The more information inside the storage, the more difficult its discovery, access and exploit is. Annotating the data with the proper metadata solves this issue.

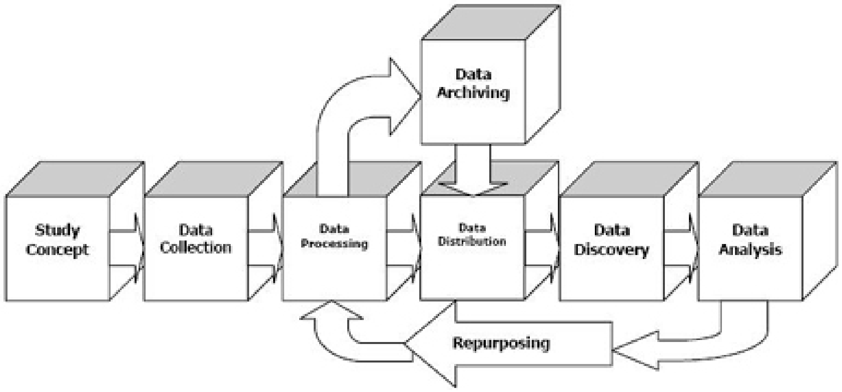


Fig. 1. Combined Life Cycle Model (ownership: DDI Alliance).

- **Identify.** The application of this verb implies the proper identification of each data resource, assigning a persistent identifier to each of them.
- **Register.** This Verb pertains to record the descriptions of the different data collections with one or more public catalogues.
- **Discover.** To improve data-reusing, ANDS suggests to enable different discovery services.
- **Access.** To guarantee the appropriate access to data, ANDS advises to provide a suitable search engine to retrieve these data. If data is not electronically available, ANDS recommends to provide contact details to get the data in conventional formats.
- **Exploit.** *Exploit*, the final *Data Sharing Verb*, comprises the tools, methodologies and support actions to enable reutilisation of data.

3.3 Ecoinformatics data life cycle

Michener and Jones define in [5] the concept of "ecoinformatics": *a framework that enables scientists to generate new knowledge through innovative tools and approaches for discovering, managing, integrating, analysing, visualizing and preserving relevant biological, environmental, and socioeconomic data and information.* To manage these data, the following data life cycle has been defined, as can be seen at Figure 2:

- **Plan.** This step involves the confection of a data management planning.
- **Collect.** This step considers both manual (hand-written data sheets) and automatic (sensor networks) data-gathering methods.
- **Assure.** Quality assurance and quality control (QA/QC), an issue addressed in previously mentioned models is not taken into account. Michener and Jones proposal is based on developing methods to guarantee the integrity of data. Quality assurance can also include the definition of standards for formats, codes, measurement units, metadata, etc.

- **Describe.** As other data life cycle models, this model remarks the value of the metadata to answer questions about *who, when, where, how* and *why*.
- **Preserve.** Data preservation implies the storage of the data and metadata, ensuring that these data can be verified, replicated and actively curated over time.
- **Discover.** The authors describe the data discovering process as *one of the greatest challenges*, as many data are not immediately available because they are stored in individual laptops. The main challenges to publish the data in a proper way are related to the creation of catalogues and indexes, and about the implementation of the proper search engines.
- **Integrate.** Integrating data from different and heterogeneous sources can become a difficult task, as it requires *understanding methodological differences, transforming data into a common representation, and manually converting and recording data to compatible semantics before analysis can begin.*
- **Analyze.** As well as the importance of a clear analysis step, this models remarks the importance of documenting this analysis with sufficient detail to enable its reproduction in different research frameworks.

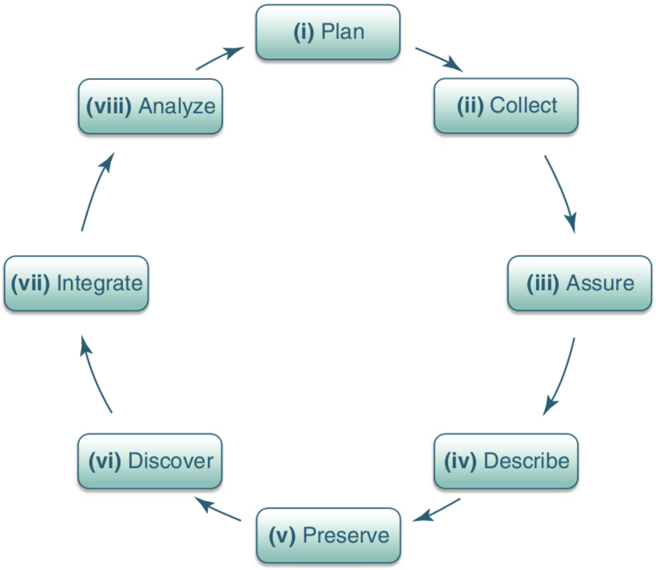


Fig. 2. Data life cycle in ecoinformatics. Taken from [5].

3.4 UK Data Archive

Another data life cycle model is the one proposed by *UK Data Archive*[1]. This model is oriented to help researchers publish their data in a manner that allows other researchers to continue their work independently. In Figure 3, the following stages can be observed:

- **Creating data.** Creating the data involves the design of the research question, planning how data are going to be managed and their sharing strategy. If we want to reuse existing data, we have to locate existing data and collect them. Whether data is new or existing, at this stage the metadata has to be created.
- **Processing data.** Like in other models, at this stage the data is translated, checked, validated and cleaned. In the case of confidential data, data needs to be "anonymized". The UK Data Archive recommends the creation of metadata at this stage too.
- **Analysing data.** At this stage data are interpreted and derived into visualizations or reports. In addition, the data are prepared for preservation, as mentioned in the following stage.
- **Preserving data.** To preserve data properly, they are migrated to the best format and stored in a suitable medium. In addition to the previously created metadata, the creating, processing, analysis and preserving processes are documented.
- **Giving access to data.** Once the data is stored, we have to distribute our data. Data distribution may involve controlling the access to them and establish a sharing license.
- **Re-using data.** At last, the data can be re-used enabling new research topics.

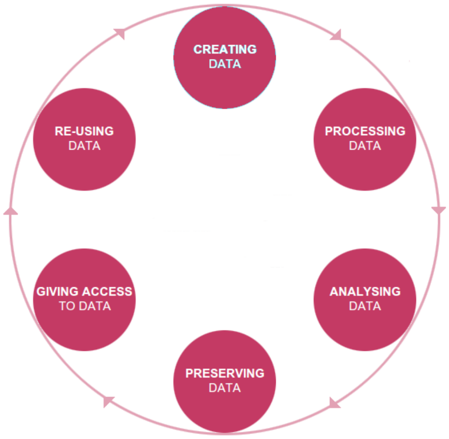


Fig. 3. Data life cycle proposed by UK Data Archive.

[1] http://www.data-archive.ac.uk/create-manage/life-cycle

3.5 The LOD2 Stack Data Life Cycle

The last analyzed data life cycle has been developed under the LOD2[2] project. This project proposes a technological and methodological stack which supports the entire life cycle of Linked Data [6]. As Figure 4 shows, the proposed life cycle phases are the following:

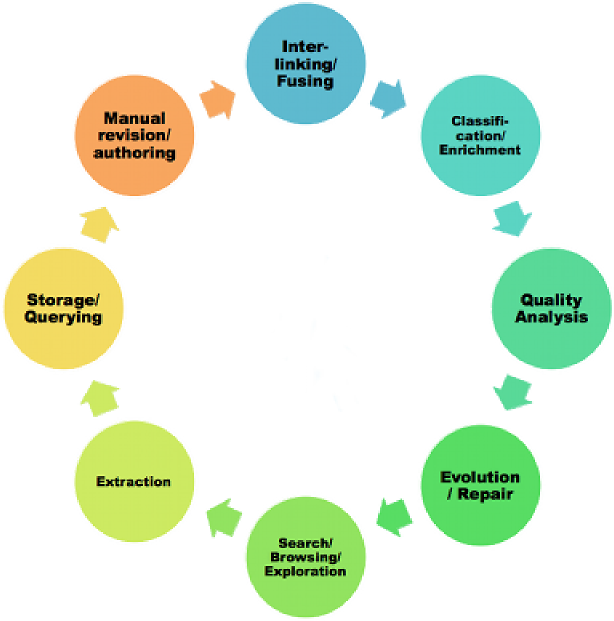

Inter-linking/ Fusing
Classifi-cation/ Enrichment
Quality Analysis
Evolution / Repair
Search/ Browsing/ Exploration
Extraction
Storage/ Querying
Manual revision/ authoring

Fig. 4. Linked Data life cycle from LOD2 stack.

- **Storage.** As RDF data presents more challenges than relational data, they propose the collaboration between known and new technologies, like columnstore technology, dynamic query optimization, adaptive caching of joins, optimized graph processing and cluster/-cloud scalability.
- **Authoring.** LOD2 provides provenance about data collected through distributed social, semantic collaboration and networking techniques.
- **Interlinking.** At this phase, LOD2 offers approaches to manage the links between different data sources.
- **Classification.** This stage deals with the transformation of raw data into Linked Data. This transformation implies the linkage and integration of data with upper level ontologies.
- **Quality.** Like other models, LOD2 develops techniques for assesing quality based on different metrics.

[2] http://lod2.eu/

- **Evolution/Repair.** At this stage, LOD2 deals with the dynamism of the data from the Web, managing changes and modifications over the data.
- **Search/Browsing/Exploration.** This stage is focused on offering Linked Data to final users through different search, browsing, exploration and visualization techniques.

3.6 A common data life cycle for smart cities

Based on these data life cycle models, we proposed a common data life cycle for managing data in smart cities. As can be seen at Figure 5, the different stages of mentioned models have been aggregated, forming our proposed model.

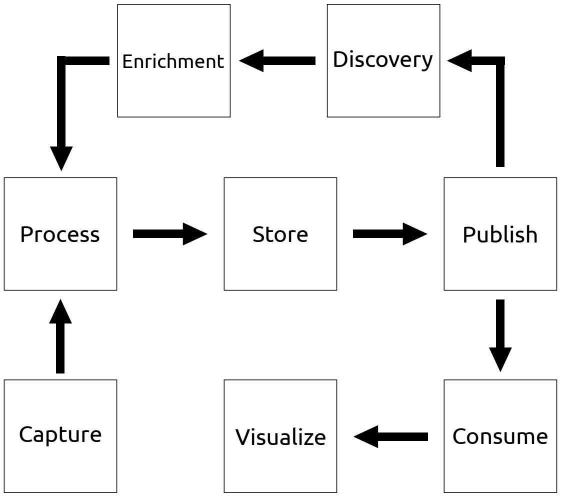


Fig. 5. Proposed model.

The different stages of this model, which are going to be explained widely in following sections, are:

- **Discovery:** The first step in our model consists of discovering where data can be taken from, identifying the available datasets which contain the necessary data to accomplish our task. Datasources can either be maintained by us, or by external entities, so the more metadata we can gather from datasets, the easier further steps will become.
- **Capture:** Once datasources are identified, data needs to be collected. In a smart city environment, there are a lot of alternatives to capture data, like sensors, data published by public administration, social networks or in more traditional way like surveys.
- **Process:** After the required data are captured, they are prepared to be stored and in need of proper methods to explore them. This

processing involves the analyzing, refining, cleaning, formatting and transformation of the data.

- **Store:** The storage of data is, probably, the most delicate action in the life cycle. Above the storage all the analysis tools are build, and is the "final endpoint" when someone requests our data. A suitable storage should have indexing, replication, distribution and backup capabilities, among other services.
- **Publish:** Most of the previously mentioned models prioritize the analysing stage over the publication stage. In our model, we defend the opposite approach for a very simple reason: when you consume your data before the publication of them, and using different processes as the rest of the people who is going to consume them, you are not making enough emphasis on publishing these data correctly. Everybody has ever met a research paper or an application in which accessing the data was difficult, or, when once the data was collected it became totally incomprehensible. To avoid this issue, we propose to publish the data before consuming them, and consume them through the same way as the rest of the people does.
- **Linkage:** Before consuming data, we suggest to search for links and relationships with other datasets found in the discovery step. Actual solutions do not allow the linkage with unknown datasets, but tools are developed to ease link discovery processes between two or more given datasources.
- **Consume:** Once the data is published, we use the provided methods to consume the data. This data consumption involves the data mining, analytics or reasoning.
- **Visualize:** To understand the data properly, designing suitable visualizations is essential to show correlations between data and the conclusions of the data analysis in a human-understandable way.

4 Identified challenges

Taking into consideration the large amounts of data present at smart cities, data management's difficulty can be described in terms of:

- Volume
- Variety
- Velocity

This three variables can also be found in *Big Data*-related articles (also known as the *Big Data's Vs*) [7, 8], so it's not surprising at all that smart cities are going to deal with Big Data problems in the near future (if they are not dealing with them right now).

Data scientist need to take into account the following three variables, which could overlap in certain enviroments. Should this happen, each scenario will determine the most relevant factors of the process, generating un-desired drawbacks on the other ones.

4.1 Volume

The high amount of data used and generated by cities nowadays needs to be properly analysed, processed, stored and eventually accessible. This means conventional IT structures need to evolve, enabling scalable storage technologies, distributed querying approaches and massively parallel processing algorithms and architectures.

However, big amounts of data should not be seen as a drawback attached to smart cities. The larger the datasets, the better analysis algorithms can perform, so deeper insights and conclusions should be expected as an outcome. These could ease the decission making stage.

As management consultant Peter Drucker once said: *"If you can't measure it, you can't manage it"*, thus leaving no way to improve it either. This adage manifests that if you want to take care of some process, but you are not able to measure it or you can't access the data, you will not be able to manage that process. That being said, the higher amounts of data available, the greater the opportunities of obtaining useful knowledge will become.

4.2 Variety

Data is rarely found in a perfectly ordered and ready for processing format. Data scientist are used to work with diverse sources, which seldom fall into neat relational structures: embedded sensor data, documents, media content, social data, etc. Variety in data sources, in storage systems, in data-types to get together in a unified analytic...

There is also an increasing concern on data trustworthiness. As pointed out by [9], *data provenance is fundamental to understanding data quality*. They also highlight that established information storage systems may not be adecuate to keep semantic sense of data.

In a previous research [10], we introduced a provenance data model to be used in user-generated Linked Data datasets, which follow W3C's PROV-O ontology[3].

[3] http://www.w3.org/TR/prov-o/

Several efforts are trying to convert existing data in high quality data, providing an extra confidence layer in which data analysts can rely.

4.3 Velocity

Finally, we must assume that data generation is experiencing an exponential growth. That forces our IT structure to not only tackle with volume issues, but with processing rates. A widely spread concept among data businesses is that sometimes you can not rely on five-minute-old data for your business logic.

That's why *streaming data* has moved from academic fields to industry to solve velocity problems. There are two main reasons to consider streaming processing:

- Sometimes, input data is too fast to store in their entirety without rocketing costs.
- If applications mandate immediate response to the data, batch processes are not suitable. Due to the rise of smartphone applications, this trend is increasingly becoming a common scenario.

5 Open Linked data as a viable approach

In the previous section, we identified some of the challenges smart cities will need to face in the following years. The data lifecycle model proposed at Figure 5 relies on Linked Open Data principles to try to solve these issues, reducing costs and enabling third parties to develop new business models on top of Linked Open Data.
Next we describe how Linked Open Data principles could help in the model's stages:

5.1 Capture

Data are the basis of smart cities, undoubtedly: services offered to citizens, decisions offered to city rulers by Decision Support Systems, all of them work thanks to big amounts of data. These data are captured from a wide variety of sources, like sensor networks installed along city, social networks or publicly available government data. In most cases, these sources publish data in a wide set of heterogeneous formats, forcing data consumers to develop different connectors for each source. As can be seen at section 5.2, there are a lot of different and widely extended ontologies which can represent data acquired from sources found in smart cities, easing the capture, integration and publication of data from heterogeneous domains. In this section, different sources of data which can be found in smart cities are shown, while in section 5.2 the transformation process from their raw data to Linked Data is exposed.

Sensor Networks A sensor network is composed by low-cost, low-power, small sized and multifunctional sensor nodes which are densely deployed either inside the phenomenon or very close to it [11]. In a smart city, these sensor networks are used for a wide range of applications, from the simple analysis of air quality[4] to the complex representation of public transport services[5], through the sensors embedded in citizens smartphones. For example, the SmartSantander project envisions the deployment of 20,000 sensors in four European cities [12]. Nowadays due the existence of open-source and cheap hardware devices like Arduino[6] or Raspberry Pi[7], the amount of collaborative and social sensor networks is growing faster and faster. Furthermore, there are software platforms like Xively[8] or Linked Sensor Middleware [13], which allow users to share the captured data from their own sensors networks in an easy way.

Social Networks Since the adoption of the Web 2.0 paradigm [14], users have become more and more active when interacting with the Web.

[4] `http://helheim.deusto.es/bizkaisense/`
[5] `http://traintimes.org.uk/map/tube/`
[6] `http://www.arduino.cc/`
[7] `http://www.raspberrypi.org/`
[8] `https://xively.com`

The clearest example of this transformation of the Web can be found in social networks and the high growth of their users. For example, at the end of the second quarter of 2013, Facebook has almost 1.2 billion users[9], while at the end of 2012, Twitter reached more than 200 million monthly active users[10]. Although users of social networks generate a lot of data, it is hard to manipulate them because users write in a language not understood by machines. To solve this issue many authors have worked with different Natural Language Processing techniques. For example, NLP and Named Entity Recognition (NER) systems [15] can be used to detect tweets which talk about some emergency situation like a car crash, an earthquake and so on; and to recognize different properties about the emergency situation like the place or the magnitude of this situation [16, 17]. Extracting data from relevant tweets could help emergency teams when planning their response to different types of situations as can be seen at [18–20].

Government Open Data Government Open Data has gained a lot of value in recent years, thanks to the proliferation of Open Data portals from different administrations of the entire World. In these portals, the governments publish relevant data for the citizens, in a heterogeneous set of formats like CSV, XML or RDF. Usually, data from these portals can be consumed by developers in an easy way thanks to the provided APIs, so there are a lot of applications developed over these data. As citizens are the most important part of smart cities, these applications make them an active part in the governance of the city.
To illustrate the importance of Government Open Data, in Table 1 some Open Data portals are shown.

Name	**Public Administration**	**No. of datasets (Sept. 2013)**	**API**
Data.gov	Government of USA	97,536	REST, SOAP, WMS
Data.gov.uk	Government of UK	10,114	REST
Data.gc.ca	Government of Canada	197,805	REST
Open Data Euskadi	Government of Basque Country	2,127	RSS, Java API, REST
Datos Abiertos de Zaragoza	Council of Zaragoza	112	SPARQL

Table 1. Open Data portals around the World.

As have been shown, in a smart city a lot of data sources can be found, publishing an abundant stream of interesting data in a different and

[9] http://techcrunch.com/2013/07/24/facebook-growth-2/

[10] https://twitter.com/twitter/status/281051652235087872

heterogeneous manner. In section 5.2, how transform these data in a standard formats is shown.

5.2 Process

As can be seen at section 1, Linked Data paradigm proposed the Resource Description Framework (RDF) as the best format to publish data and the reuse of widely extended ontologies. In this section we explain **what** is an ontology, **which** are the most popular ontologies and **how** we can map previously captured raw data to a proper ontology. At the end of this section a set of best practices to construct suitable URIs for Lined Data are shown.

As defined by [21], an ontology is *a formal explicit description of concepts in a domain of discourse, properties of each concept describing various features and attributes of the concept, and restrictions on slots.* According to this definition, an ontology has *Classes* which represent concepts, *Properties* which represent different characteristic of *Classes* and *Restrictions* on the values of these properties and relationships among different *Classes*. An ontology allows modelling data avoiding most of ambiguities originated when fusing data from different sources, stimulating the interoperability among different sources. As we see in section **??** in a smart city, data came from a wide variety of sources, whereby the ontologies seem to be a suitable option to model these data.

Following works use ontologies to model different data sources which can be found in a smart city. In Bizkaisense project [22], diverse ontologies like Semantic Sensor Network ontology (SSN) [23], Semantic Web for Earth and Environmental Terminology (SWEET) [24] or Unified Code for Units of Measure ontology (UCUM)[11] are used to model raw data from air quality stations from Basque Country. AEMET Linked Data project[12] has developed a network of ontologies composed by SSN ontology, OWL-Time ontology[13], wsg84_pos ontology[14], GeoBuddies ontology network[15] and its own AEMET ontology, to describe measurements taken by meteorological stations from AEMET (Spanish National Weather Service). In [25] authors extend SSN ontology to model and publish as Linked Data the data stream generated by the sensors of an Android powered smartphone.

Another example of semantic modelling of infrastructures from a city can be found in LinkedQR [26]. LinkedQR is an application that eases the managing task of an art gallery allowing the elaboration of interactive tourism guides through third parties Linked Data and manual curation. LinkedQR uses MusicOntology [27] to describe the audioguides and Dublin Core [28], DBpedia Ontology[16] and Yago [29] to describe other basic information.

[11] http://idi.fundacionctic.org/muo/ucum-instances.html

[12] http://aemet.linkeddata.es/models.html

[13] http://www.w3.org/TR/owl-time/

[14] http://www.w3.org/2003/01/geo/wgs84_pos

[15] http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/ontologies/83-geobuddies-ontologies

[16] http://dbpedia.org/ontology/

LinkedStats project[17] takes data about waste generation and population of Biscay to develop a statistical analysis about the correlation between these two dimensions of the data. It models these statistical data through the RDF Data Cube Vocabulary [30], an ontology developed for modelling multi-dimensional data in RDF. At last, in [31] the authors show how Linked Data enables the integration of data from different sensor networks.

The mapping between raw data and ontologies, usually is made by applications created *ad-hoc* to each case; Bizkaisense, AEMET Linked Data and LinkedStats have their own Python[18] scripts to generate proper RDF files from raw data. In the case of LinkedQR, it has a control panel where the manager can manually type data and map to a desired ontology. Instead, there are tools designed for transforming raw data into structured data. One of them is Open Refine[19] (formerly Google Refine). Open Refine is a web-tool which can apply different manipulations to data (facets, filters, splits, merges, etc.) and export data in different formats based on custom templates. Additionally, Google Refine RDF Extension[20] allows exporting data in RDF.

Another interesting tool is Virtuoso Sponger, a component of OpenLink Virtuoso[21]. Virtuoso Sponger generates Linked Data from different data sources, through a set of extractors called *Cartridges*. There are different Cartridges which support wide variety of input formats (CSV, Google KML, xHTML, XML, etc.) and vendor specific Cartridges too (Amazon, Ebay, BestBuy, Discogs, etc.).

After modelling data, one of most important concepts in Linked Data are the URIs or Unified Resource Identifiers. As shown in section 2, to publish a data resource as Linked Data it has to be identified by an HTTP URI which satisfies these conditions:

- A HTTP URI is unique and consistent.
- A HTTP URI can be accessed from everywhere by everyone.
- The URI and its hierarchy are auto-descriptive.

Designing valid URIs is a very important step into the publication of Linked Data: if you change URIs of your data, all the incoming links from external sources are going to be broken. To avoid this issue there is a set of good practices, proposed in [32]:

- **Be on the web.** Return RDF for machines and HTML for humans through standard HTTP protocol.
- **Don't be ambiguous.** Use a URL to describe a document and a different URI to identify real-world objects. A URI can't stand for both document and real-world object.

To apply these good practices correctly, authors propose two solutions, which nowadays have been widely adopted by Linked Data community:

- **303 URIs.** Use `303 See Other` status code for redirecting to the proper RDF description of the real world object or to the HTML document. For example:

[17] `http://helheim.deusto.es/linkedstats`

[18] `http://www.python.org/`

[19] `http://openrefine.org/`

[20] `http://refine.deri.ie/`

[21] `http://virtuoso.openlinksw.com/`

  - `http://helheim.deusto.es/hedatuz/resource/biblio/5112` - A URI identifying a bibliographic item.
  - `http://helheim.deusto.es/hedatuz/page/biblio/5112` - The HTML view of the item.
  - `http://helheim.deusto.es/hedatuz/data/biblio/5112` - A RDF document describing the item.
- **Hash URIs.** URIs contains a *fragment*, a special part separated by the symbol #. The client has to strip-off the fragment from the URI before requesting it to server. Server returns a RDF document in which the client has to search the fragment.

Tools implementing these techniques are described on section 5.4.

5.3 Store

Once data is mapped to a proper ontology and the RDF files are generated, is time to store them. Due the big amount of data generated in a city, an appropriate storage has to:

- Support different input data formats.
- Manage and index big amounts of data properly.
- Execute queries over data in an efficient way.
- Offer different interfaces and APIs to users, allowing them to consume data in a wide variety of formats.

Along this section, the first two points are discussed, while third point is discussed in section 5.4.

Before a wide description of each datastore we have analysed, a brief description is presented in Table 2.

Datastore	**Input formats**	**API**	**Output formats**
Virtuoso	RDF/XML, N3, Turtle, N-Triples, N-Quads, etc.	SPARQL, SPARQL UPDATE, Java (Jena, Sesame, Redland)	HTML, Spreadsheet, XML, RDF+JSON, JS, NTriples, RDF/XML, CSV
Stardog	NTRIPLES, RDF/XML, Turtle, TRIG, TRIX, N3, N-Quads	SPARQL (CLI, HTTP), Java, JS, Groovy, Spring, Ruby	N-Triples, RDF/XML, TURTLE, TRIG, TRIX, N3, N-Quads
Fuseki	RDF/XML, N-Triples, Turtle, etc.	SPARQL, SPARQL UPDATE, Java	JSON, XML, Text, CSV, TSV

Table 2. Summary of characteristics of selected datastores.

The first datastore to be reviewed is Virtuoso by Openlink, mentioned in section 5.2. Virtuoso is a hybrid server that manages SQL, XML and RDF in a single server. It is available in both Open Source and commercial versions. As seen in Table 2, it supports a wide variety of input

and output formats. It has a SPARQL endpoint, accessible thus by web-interface as by HTTP GET requests, allowing to web-agents the access to data. Virtuoso supports the SPARQL UPDATE syntax, allowing the update of datastore through HTTP POST requests; and it provides connectors for different Java powered RDF engines, like Jena[22], Sesame[23] or Redland[24]. Further, it supports some OWL properties for reasoning. According to the Berlin SPARQL Benchmark [33] Virtuoso 7 can load one billion triples in 27:11 minutes.
Another datastore which is becoming popular is Stardog[25]. Developed by Clark & Parsia, Stardog is a RDF database which supports SPARQL querying and OWL reasoning. It offers a Command Line Interface to manage the different databases (create, remove, add/remove data, SPARQL queries, etc.), while they can be queried through HTTP too. Furthermore it has its own query syntax which indexes the RDF literals; and its own Java library to manage databases from Java applications. It supports OWL 2 reasoning, supporting different OWL profiles[26] like QL, RL, EL or DL.
The last analysed datastore is Fuseki[27]. Part of Jena framework, Fuseki (formerly known as Joseki) offers RDF data over HTTP, in a REST style. Fuseki implements W3C's SPARQL 1.1 Query, Update, Protocol and Graph Store HTTP Protocol. It has a web-panel to manage the datastore and can interact with the rest of the Jena components.
As can be seen at Table 2, all analysed datastores are similar in terms of input/output formats or offered APIs. But there are differences in other aspects, like security: Fuseki does not support users and access roles. Another difference is the installation and executing complexity: while Fuseki and Stardog are launched as a Java JAR file, Virtuoso can be installed through Debian's package system and launched as a UNIX daemon. In the other hand, Virtuoso is more than a "simple" RDF store, Virtuoso is a relational database engine, an application server in which both preinstalled applications and our own applications can be launched, and much more. Furthermore, Virtuoso can be installed in a cluster formed by multiple servers.
Concluding this section, we can say that Fuseki can be used in light installations, when hardware and data are limited; Stardog in more complex systems, due its fast query execution times. Meanwhile, Virtuoso offers more services like Sponger (described in section 5.2) or Semantic Wiki, whereby, it can be suitable for environments which need more than the simple storage of RDF triples.

5.4 Publish

Publication stage is one of the most important stages in the life cycle of Linked Data in smart cities, this stage determines how citizens or

[22] `http://jena.apache.org/`

[23] `http://www.openrdf.org/`

[24] `http://librdf.org/`

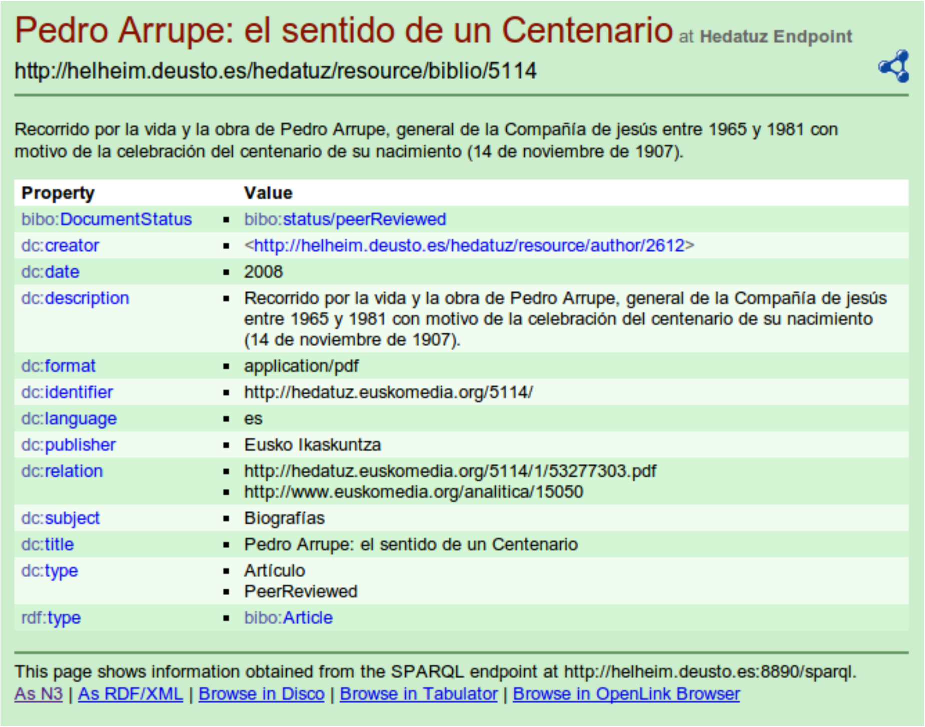
[25] `http://stardog.com/`

[26] `http://www.w3.org/TR/owl2-profiles/`

[27] `http://jena.apache.org/documentation/serving_data/index.html`

developers can acquire Linked Data to consume (section 5.7) through different discovery methods (section 5.5). As we saw in section 5.2, the three proposed RDF stores include some publication API or SPARQL endpoint, but, sometimes the specifications of the system to be deployed need additional features at this publication stage.

One of these features can be the `303 Redirection` explained at section 5.3. Although all datastores mentioned on section 5.3 offer a SPARQL endpoint to explore data, Linked Data paradigm demands resolvable HTTP URIs as resource identifiers. Fortunately, there are tools which fulfil this demand. One of them, Pubby[28], adds Linked Data interfaces to SPARQL endpoint. Pubby queries the proper SPARQL endpoint to retrieve data related to a given URI and manages the `303 Redirection` mechanism. Depending on the `Accept` header of the HTTP request, Pubby redirects the client to the HTML view of data or to the RDF document describing the resource. Pubby can export data in RDF/XML, NTriples, N3 and Turtle. In Figure 6 an example of the HTML view of a resource is shown.



Pedro Arrupe: el sentido de un Centenario at Hedatuz Endpoint

http://helheim.deusto.es/hedatuz/resource/biblio/5114

Recorrido por la vida y la obra de Pedro Arrupe, general de la Compañía de jesús entre 1965 y 1981 con motivo de la celebración del centenario de su nacimiento (14 de noviembre de 1907).

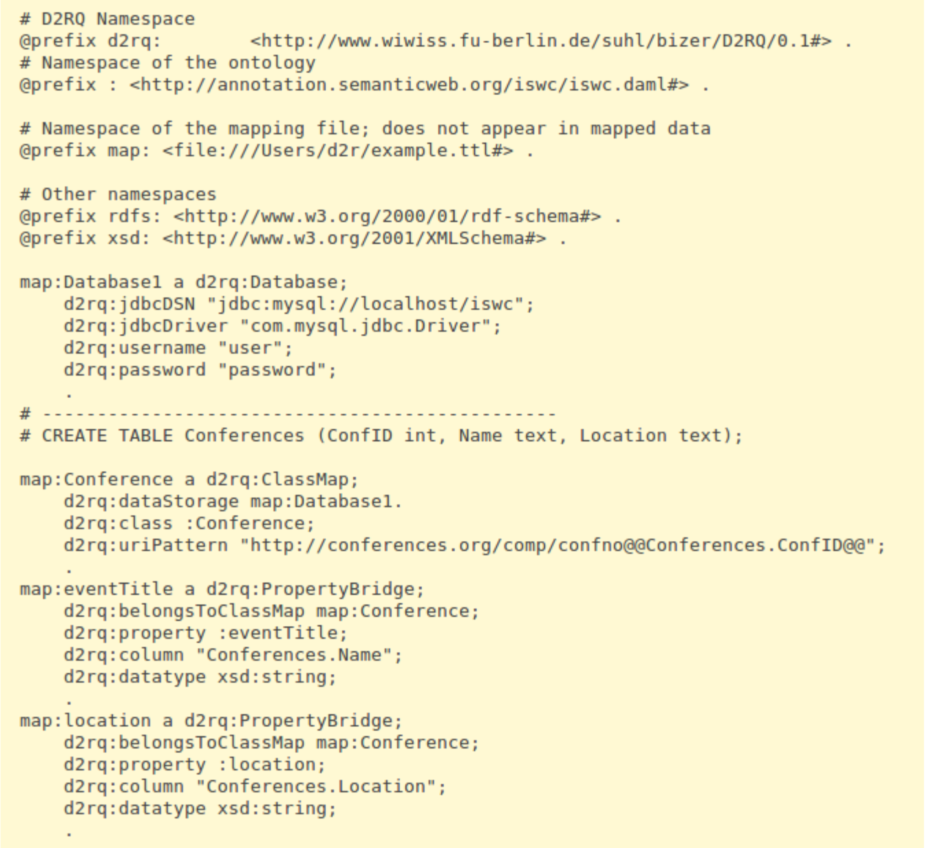
Property	Value
bibo:DocumentStatus	▪ bibo:status/peerReviewed
dc:creator	▪ <http://helheim.deusto.es/hedatuz/resource/author/2612>
dc:date	▪ 2008
dc:description	▪ Recorrido por la vida y la obra de Pedro Arrupe, general de la Compañía de jesús entre 1965 y 1981 con motivo de la celebración del centenario de su nacimiento (14 de noviembre de 1907).
dc:format	▪ application/pdf
dc:identifier	▪ http://hedatuz.euskomedia.org/5114/
dc:language	▪ es
dc:publisher	▪ Eusko Ikaskuntza
dc:relation	▪ http://hedatuz.euskomedia.org/5114/1/53277303.pdf ▪ http://www.euskomedia.org/analitica/15050
dc:subject	▪ Biografías
dc:title	▪ Pedro Arrupe: el sentido de un Centenario
dc:type	▪ Artículo ▪ PeerReviewed
rdf:type	▪ bibo:Article

This page shows information obtained from the SPARQL endpoint at http://helheim.deusto.es:8890/sparql.
As N3 | As RDF/XML | Browse in Disco | Browse in Tabulator | Browse in OpenLink Browser

Fig. 6. Example of HTML visualization of a resource from Hedatuz dataset by Pubby.

[28] `http://wifo5-03.informatik.uni-mannheim.de/pubby/`

D2R Server [34] allows the publication of relational databases as Linked Data. At first D2R requires the mapping of tables and columns from database to selected ontologies using D2RQ Mapping Language. Once this mapping is done, D2R offers a SPARQL endpoint to query data and a Pubby powered interface.



```
# D2RQ Namespace
@prefix d2rq:        <http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/0.1#> .
# Namespace of the ontology
@prefix : <http://annotation.semanticweb.org/iswc/iswc.daml#> .

# Namespace of the mapping file; does not appear in mapped data
@prefix map: <file:///Users/d2r/example.ttl#> .

# Other namespaces
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

map:Database1 a d2rq:Database;
    d2rq:jdbcDSN "jdbc:mysql://localhost/iswc";
    d2rq:jdbcDriver "com.mysql.jdbc.Driver";
    d2rq:username "user";
    d2rq:password "password";
    .
# -----------------------------------------------
# CREATE TABLE Conferences (ConfID int, Name text, Location text);

map:Conference a d2rq:ClassMap;
    d2rq:dataStorage map:Database1.
    d2rq:class :Conference;
    d2rq:uriPattern "http://conferences.org/comp/confno@@Conferences.ConfID@@";
    .
map:eventTitle a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:Conference;
    d2rq:property :eventTitle;
    d2rq:column "Conferences.Name";
    d2rq:datatype xsd:string;
    .
map:location a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:Conference;
    d2rq:property :location;
    d2rq:column "Conferences.Location";
    d2rq:datatype xsd:string;
    .
```

Fig. 7. D2RQ mapping file. Example taken from http://d2rq.org/d2rq-language.

In the process of publishing data from smart cities as Linked Data, new ontologies are going to be created to model the particularities of each city. The creators of these ontologies have to publish a suitable documentation, allowing the proper reuse of them. A tool for publishing ontologies and their documentation is Neologism[29]. Neologism shows ontologies in a human-readable manner, representing class, subclass and property relationships through diagrams.

[29] http://neologism.deri.ie/

5.5 Discovery

The discovery stage starts a loop on the model after the publishing step, augmenting the number of third parties resources data is taken from. Discovery can be achieved following two different paths: *a)* incrementing the number of datasets from where data are taken from and *b)* stablishing relationships between existing data and external datasets, in order to provide links for new information stores.

The first approach is followed by [35], a semantic sensor network middleware which allows for effective discovery of sensors on the Web. This work's outcomes can be applied to sensorized smart cities, as it enables the possibility to add new sensors' data as input to the processing algorithms by location (identifying those sensors which are near-by or within a certain distance from a given place).

Regarding the second approximation, different frameworks have been developed to deal with class and properties matching. The basis of these frameworks is to provide data discovery features through links to external entities related to the items used in the analysis.

The *Silk - Link Discovery Framework* [36] offers a flexible tool for discovering links between entities within different Web data sources. Silk makes use of *Silk - Link Specification Language* (Silk-LSL), a declarative language which lets data publishers specify which RDF link types should be discovered providing two related datasets, and the conditions under data items must fulfill to be interlinked. As an example, a script in Silk-LSL can be written to match cities between *DBpedia* ontology's *City* or *PopulatedPlace* classes, and *GeoName*'s feature class *gn:P*. As constraints, string similarity metrics can be used to match city names, and take into consideration cities' bounding boxes (i.e. the margins projected on a map) to check overlaps.

With a similar approach *LIMES* (LInk discovery framework for MEtric Spaces) [37] can be used for the discovery of links between Linked Data knowledge bases, focusing on a time-efficient approach especially when working with large-scale matching tasks. LIMES relies on *triangle inequality* mathematical principles for distance calculations, which reduce the number of comparisons necessary to complete a mapping by several orders of magnitude. This approach helps detecting the pairs that will not fulfil the requirements in an early stage, thus avoiding spending time in more time-consuming processing.

5.6 Enrichment

5.7 Consume

At this stage, the focus is located on consuming data for business-logic processes, should they involve data mining algorithms, analytics, reasoning, etc.

Whereas complex processing algorithms can be used independently of the dataset format, Linked Open Data can greatly help at reasoning purposes. Linked Open Data describes entities using ontologies, semantic constraints and restriction rules (belonging, domain, range, etc.) which

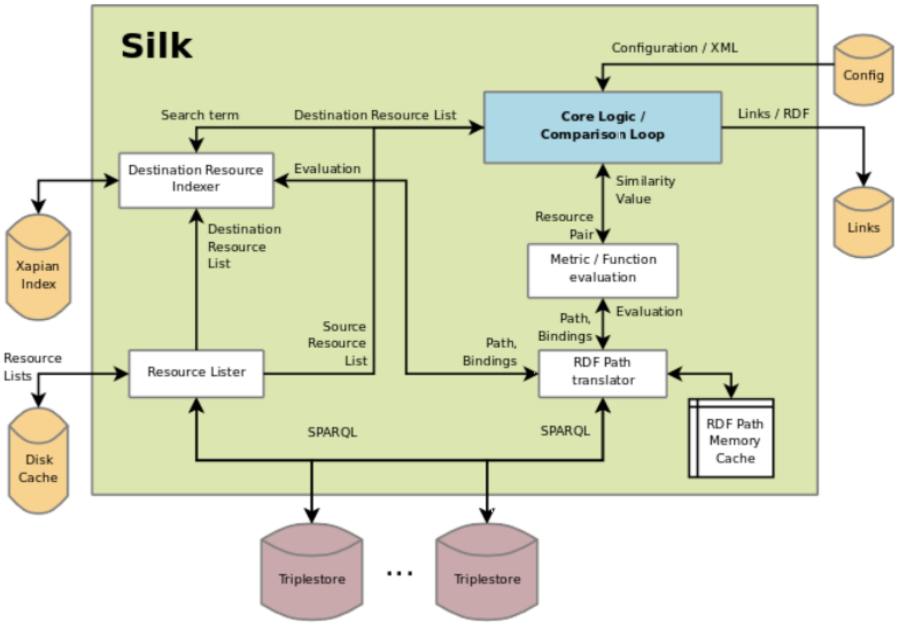


Fig. 8. Silk framework architecture

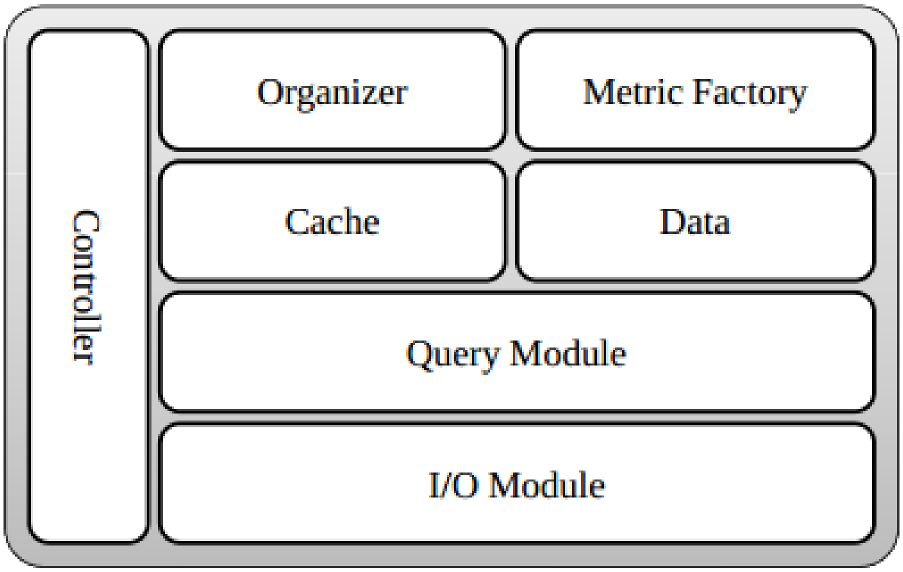


Fig. 9. LIMES framework architecture

favor the inference of new information from the existing one. Thanks to those semantics present in Linked Data, algorithms are not fed with raw values (numbers, strings...), but with semantically meaningful information (height in cm, world countries, company names...), thus resulting in higher quality outputs and making algorithms more error aware (i.e., if a given algorithm is in charge of mapping the layout of a mountainous region, and founds the height of one of the mountains to be *3.45*, it is possible to detect the conversion has failed at some point, as height datatype was expected to be given in *meters*).
As seen in section **??**, sensors and social networks are common data input resources, generating huge amounts of data streamed in real time. The work done in [38] comprises a set of best practices to publish and link stream data to be part of the Semantic Web.
However, when it comes to consume Linked Data streams, SPARQL can find its limits [39]. Stream-querying languages such as CQELS's[30] language (an extension of the declarative SPARQL 1.1 language using the EBNF notation) can greatly help in the task. CQELS[40] (Continuous Query Evaluation over Linked Stream) is a native and adaptive query processor for unified query processing over Linked Stream Data and Linked Data developed at DERI Galway[31].
Initially, a query pattern is added to represent window operators on RDF Stream:

```
GraphPatternNotTriples ::= GroupOrUnionGraphPattern |
OptionalGraphPattern | MinusGraphPattern | GraphGraphPattern |
*StreamGraphPattern* | ServiceGraphPattern | Filter | Bind
```

Assuming that each stream has an IRI as identification, the *StreamGraphPattern* is defined as follows:

```
StreamGraphPattern ::= ’STREAM’ ’[’Window’]’ VarOrIRIref
’{’TriplesTemplate’}’
Window ::= Rangle|Triple|’NOW’|’ALL’
Range ::= ’RANGE’ Duration (’SLIDE’ Duration |’TUMBLING’)?
Triple ::= ’TRIPLES’ INTEGER
Duration ::= (INTEGER ’d’|’h’|’m’|’s’|’ms’|’ns’)+
```

An example query could be:

```
PREFIX lv: <http://deri.org/floorplan/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?locName FROM NAMED <http://deri.org/floorplan/>
WHERE { STREAM <http://deri.org/streams/rfid> [NOW]
        {?person lv:detectedAt ?loc}
        {?person foaf:name ”AUTHORNAME”^^xsd:string }

        GRAPH <http://deri.org/floorplan/>
        {?loc lv:name ?locName}
}
```

Eventually, both batch and streaming results can be consumed through *REST* services by web or mobile applications, or serve as input for more processing algorithms until they are finally presented to end-users.

[30] https://code.google.com/p/cqels/

[31] http://www.deri.ie/

5.8 Visualize

In order to make meaning from data, humans have developed a great ability to understand visual representations. The main objective of data visualization is to communicate information in a clean and effective way through graphical means. It's also suggested that visualization should also encourage users engagement and attention.

The *"A picture is worth a thousand words"* saying reflects the power images and graphics have when expressing information, and can condense big datasets into a couple of representative, powerful images.

As Linked Data is based on subject-predicate-object triples, graphs are a natural way to represent triple stores, where subject and object nodes are inter-connected through predicate links. When further analysis is applied on triples, a diverse variety of representations can be chosen to show processed information: charts, infographics, flows, etc.[41]

Browser-side visualization technologies such as d3.js[32] (by Michael Bostock) and Raphaël[33] are JavaScript-based libraries to allow the visual representation of data on modern web browsers, allowing anybody with a minimum internet connection try to understand data patterns in a graphical form.

For developers not familiar with visualization techniques, some investigations are trying to enable the automatic generation of graphical representations of Linked Data query results. The *LDVM* (Linked Data Visualization Model) [42] is proposed as a model to rapidly create visualizations of RDF data. *LODVisualization*[34] is an implemented prototype which supports the LDVM.

Visualbox[35] is a simplified edition of LODSPeaKr[36] focused on allowing people create visualizations using Linked Data.

Independently of how visualizations are generated, they provide a perfect solution to present high-quality, refined data to end-users.

[32] http://d3js.org/

[33] http://raphaeljs.com/

[34] http://lodvisualization.appspot.com/

[35] http://alangrafu.github.io/visualbox/

[36] http://lodspeakr.org/

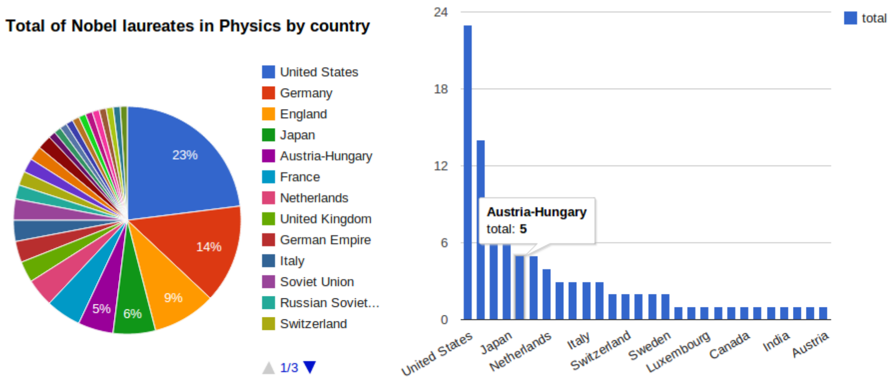


Fig. 10. Visualbox graph example (ownership: Alvaro Graves).

6 Evaluation

7 Lessons learned

8 Further research

References

1. Bizer, C., Boncz, P., Brodie, M.L., Erling, O.: The meaningful use of big data: four perspectives – four challenges. SIGMOD Rec. **40**(4) (January 2012) 56–60
2. Initiative, D.D.: Overview of the DDI version 3.0 conceptual model (April 2008)
3. Ball, A.: Review of data management lifecycle models. (February 2012)
4. Burton, A., Treloar, A.: Designing for discovery and re-use: the 'ANDS data sharing verbs' approach to service decomposition. International Journal of Digital Curation **4**(3) (2009) 44–56
5. Michener, W.H., Jones, M.B.: Ecoinformatics: supporting ecology as a data-intensive science. (2012)
6. Auer, S., Bühmann, L., Dirschl, C., Erling, O., Hausenblas, M., Isele, R., Lehmann, J., Martin, M., Mendes, P.N., Van Nuffelen, B., et al.: Managing the life-cycle of linked data with the LOD2 stack. In: The Semantic Web–ISWC 2012, Springer (2012) 1–16
7. Zikopoulos, P., Eaton, C., et al.: Understanding big data: Analytics for enterprise class hadoop and streaming data. McGraw-Hill Osborne Media (2011)
8. Russom, P.: Big data analytics. TDWI Best Practices Report, Fourth Quarter (2011)
9. Buneman, P., Davidson, S.B.: Data provenance–the foundation of data quality (2013)
10. Emaldi, M., Pena, O., Lázaro, J., Vanhecke, S., Mannens, E.: To trust, or not to trust: Highlighting the need for data provenance in mobile apps for smart cities
11. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. Communications magazine, IEEE **40**(8) (2002) 102–114
12. Sanchez, L., Galache, J.A., Gutierrez, V., Hernandez, J., Bernat, J., Gluhak, A., Garcia, T.: SmartSantander: the meeting point between future internet research and experimentation and the smart cities. In: Future Network & Mobile Summit (FutureNetw), 2011. (2011) 1–8
13. Le-Phuoc, D., Quoc, H.N.M., Parreira, J.X., Hauswirth, M.: The linked sensor middleware–connecting the real world and the semantic web. Proceedings of the Semantic Web Challenge (2011)
14. O'reilly, T.: What is web 2.0: Design patterns and business models for the next generation of software. Communications & strategies (1) (2007) 17
15. Maynard, D., Tablan, V., Ursu, C., Cunningham, H., Wilks, Y.: Named entity recognition from diverse text types. In: Recent Advances in Natural Language Processing 2001 Conference. (2001) 257–274
16. Sixto, J., Pena, O., Klein, B., López-de Ipina, D.: Enable tweet-geolocation and don't drive ERTs crazy! improving situational awareness using twitter

17. Martins, B., Anastácio, I., Calado, P.: A machine learning approach for resolving place references in text. In: Geospatial Thinking. Springer (2010) 221–236
18. Abel, F., Hauff, C., Houben, G.J., Stronkman, R., Tao, K.: Twitcident: fighting fire with information from social web streams. In: Proceedings of the 21st international conference companion on World Wide Web. (2012) 305–308
19. Vieweg, S., Hughes, A.L., Starbird, K., Palen, L.: Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. (2010) 1079–1088
20. Hughes, A.L., Palen, L.: Twitter adoption and use in mass convergence and emergency events. International Journal of Emergency Management **6**(3) (2009) 248–260
21. Noy, N.F., McGuinness, D.L., et al.: Ontology development 101: A guide to creating your first ontology. Stanford knowledge systems laboratory technical report KSL-01-05 and Stanford medical informatics technical report SMI-2001-0880 (2001)
22. Emaldi, M., Lázaro, J., Aguilera, U., Peña, O., López de Ipiña, D.: Short paper: Semantic annotations for sensor open data. Proc. of the 5th International Workshop on Semantic Sensor Networks (2012)
23. Lefort, L., Henson, C., Taylor, K., Barnaghi, P., Compton, M., Corcho, O., Garcia-Castro, R., Graybeal, J., Herzog, A., Janowicz, K.: Semantic sensor network XG final report. W3C Incubator Group Report (2011)
24. Raskin, R.G., Pan, M.J.: Knowledge representation in the semantic web for earth and environmental terminology (SWEET). Computers & Geosciences **31**(9) (2005) 1119–1125
25. d'Aquin, M., Nikolov, A., Motta, E.: Enabling lightweight semantic sensor networks on android devices. (2011)
26. Emaldi, M., Lázaro, J., Laiseca, X., López-de Ipiña, D.: Linkedqr: improving tourism experience through linked data and qr codes. In: Ubiquitous Computing and Ambient Intelligence. Springer (2012) 371–378
27. Raimond, Y., Abdallah, S.A., Sandler, M.B., Giasson, F.: The music ontology. In: ISMIR, Citeseer (2007) 417–422
28. Weibel, S., Kunze, J., Lagoze, C., Wolf, M.: Dublin core metadata for resource discovery. Internet Engineering Task Force RFC **2413** (1998) 222
29. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th international conference on World Wide Web, ACM (2007) 697–706
30. : The RDF data cube vocabulary (2013)
31. Stasch, C., Schade, S., Llaves, A., Janowicz, K., Bröring145, A.: Aggregating linked sensor data. SEMANTIC SENSOR NETWORKS (2011) 46
32. Ayers, A., Völkel, M.: Cool uris for the semantic web. Woking Draft. W3C (2008)
33. Bizer, C., Schultz, A.: The berlin sparql benchmark. International Journal on Semantic Web and Information Systems (IJSWIS) **5**(2) (2009) 1–24

34. Bizer, C., Cyganiak, R.: D2r server-publishing relational databases on the semantic web. In: 5th international Semantic Web conference. (2006) 26
35. Pschorr, J., Henson, C., Patni, H., Sheth, A.: Sensor discovery on linked data. In: Proceedings of the 7th Extended Semantic Web Conference, ESWC2010, Heraklion, Greece. Volume 30. (2010)
36. Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Silk-a link discovery framework for the web of data. In: LDOW, Citeseer (2009)
37. Ngomo, A.C.N., Auer, S.: Limes: a time-efficient approach for large-scale link discovery on the web of data. In: Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three, AAAI Press (2011) 2312–2317
38. Sequeda, J.F., Corcho, O.: Linked stream data: A position paper. (2009)
39. Della Valle, E., Ceri, S., van Harmelen, F., Fensel, D.: It's a streaming world! reasoning upon rapidly changing information. Intelligent Systems, IEEE **24**(6) (2009) 83–89
40. Le-Phuoc, D., Dao-Tran, M., Parreira, J.X., Hauswirth, M.: A native and adaptive approach for unified processing of linked streams and linked data. In: The Semantic Web–ISWC 2011. Springer (2011) 370–388
41. Khan, M., Khan, S.S.: Data and information visualization methods, and interactive mechanisms: A survey. International Journal of Computer Applications **34**(1) (2011) 1–14
42. Brunetti, J.M., Auer, S., García, R.: The linked data visualization model. In: International Semantic Web Conference (Posters & Demos). (2012)

9 Acronyms and terms