Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia
Departamento de Informática

Master Thesis

European Master in Computational Logic

Publishing Linked Data: The Pordata Use Case.

Tatiana Tarasova

Lisboa
(2012)

Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia
Departamento de Informática

Dissertação de Mestrado

Publicação de Linked Data: O Caso de Uso Pordata.

Tatiana Tarasova

Supervisors: Prof. João Leite
Prof. Alfredo Gabaldon

Thesis presented at the Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, in order to obtain a Master's Degree in Computer Science.

Lisboa
(2012)

Acknowledgements

I would like to thank my thesis advisors Prof. João Leite and Prof. Alfredo Gabaldon for their kind supervision, productive discussions and suggestions that guided me in my research. I am grateful to all the professors in the EMCL program whom I was fortunate to learn from. My special thanks go to Prof. Steffen Hölldobler and all organizers of the EMCL program, to Federica Cumer and Sandra Rainha for their help in solving numerous bureaucratic issues. My studies were supported by the Erasmus Mundus scholarship without which it would have been impossible to study and live these two wonderful years in Europe.

No doubts, the program gave me more than just knowledge. I met many interesting people and made new friends from all over the world, who became my family during the studies and hopefully will remain such. I want to thank all of them for supporting me and sharing memorable moments with me. I heartily thank Marco for the proofreadings, giving useful comments and, more importantly, for being there. I am grateful to Katya, Ira, Ana, Andrey, Natasha and all my friends back in Russia who supported me throughout my study. Finally, I am deeply grateful to my grandmother for her love, faith in me and understanding.

Summary

Pordata is an initiative of the Francisco Manuel dos Santos Foundation. The Foundation aims at promoting knowledge about Portugal and Portuguese reality to citizens so that they can participate in the public debate and contribute to the development of the modern democratic society. The Pordata project was established with the goals to collect statistics about Portugal and publish them on the Web.

In the current work we present our Proposal for publishing Pordata as Linked Data. To do so, we analyse how Pordata data is being currently published, i.e., how it is represented and how it can be reused. We introduce the Linked Data principles, a set of recommendations on how to publish and interlink semantically rich machine-readable data on the Web. The idea of Linked Data is to build a Web of Data by applying the same principles that were used to build the traditional Web of documents. We present the Linking Open Data project that aims to convert existing Open Data sets into Linked Data and overview existing applications that were built to benefit from Linked Data. We propose our use cases demonstrating advantages of the Linked Data version of Pordata statistics. Finally, we present a Web application that was developed to implement the use cases.

Keywords: Linked Data, Semantic Web, Pordata, statistics.

Sumário

Pordata é uma iniciativa da Fundação Manuel Francisco dos Santos. A Fundação tem como objectivo promover o conhecimento sobre Portugal, e a sua realidade, junto dos cidadãos para que estes possam participar no debate público e contribuir para o desenvolvimento da sociedade democrática moderna. O projecto Pordata foi criado com o objectivo de coleccionar estatísticas sobre Portugal e publicá-las na web. No presente trabalho, apresentamos uma proposta para publicação do Pordata de acordo com a iniciativa Linked Data. Para isso, analisámos a forma como os dados do Pordata estão publicados actualmente, ou seja, como estão representados e como podem ser reutilizados. Apresentamos os princípios subjacentes à iniciativa Linked Data - um conjunto de recomendações sobre como publicar e interligar dados, com uma semântica rica, de forma a que possam ser interpretados e processados por máquinas na web.

A ideia da iniciativa Linked Data é construir uma rede de dados aplicando os mesmos princípios que foram utilizados para construir a Web tradicional de documentos. Apresentamos o projecto Linked Open Data que visa converter conjuntos de dados publicados de forma aberta (Open Data) de acordo com os princípios Linked Data, e estudamos aplicações desenvolvidas para beneficiar da integração na iniciativa Linked Data. Propomos casos de estudo que demonstram as vantagens da existência do Pordata integrado na iniciativa Linked Data. Finalmente apresentamos uma aplicação Web que implementa os casos de estudo.

Palavras-chave: Linked Data, Semantic Web, Pordata, estatística.

Contents

List of Figures

1

Introduction

1.1 Motivation

Pordata is an initiative of the *Francisco Manuel dos Santos Foundation*[1] (*FFMS*). The Foundation has as its aim to promote existing information about Portugal and Portuguese society to the population, e.g., educational and social issues, the costs of health, culture seminars, etc. According to the Foundation's statutes, *"it believes that progress of societies depends very much on the participation of their citizens in the public debate of all the issues which affect and interest them. The Foundation recognizes that participation and debate are conditioned by knowledge of the facts and possession of relevant information and that creative public debate can only really take place when there are no obstacles among free people with informed opinions*.

Pordata is the first contribution of FFMS to that public debate. With respect to this, FFMS is especially committed to disseminating among the population fundamental information about Portugal: Portuguese statistics. The Pordata website was created to collect, organise and provide access to the Portuguese statistics on the Web[2]. Pordata contains various statistics about Portugal, including demographic, social and economic conditions, health, education, government spending, culture, sports, and many others. These statistics essentially reflect different aspects of the Portuguese society and accumulate knowledge of many years of the Portuguese reality. The foundation is confident that such vital knowledge is of interest for the whole population and should

[1] `http://www.ffms.pt/`

[2] The Pordata website is available at `http://www.pordata.pt/` .

not be restricted to statisticians, policy makers and other experts.

This vision, that FFMS promotes about the modern democratic society, is shared by many other countries. The government of the United States was the first to launch a website containing catalogue of public data in 2009 [36]. The government of the United Kingdom released its public portal with government data in 2010 [51]. Among the data they made available are budget expenses, legislative tracking, topographic data and maps, public transportation and information about other public services, census data and various kinds of statistics. By making this information publicly available, the governments want to achieve better transparency and democratic accountability. The initiative of the US and UK was supported by governments and local authorities in many other countries, including Australia [3], New Zealand [82], Netherlands [60], Spain [105], Austria [4], Canada [24] and others, and developed into a global movement, the Open Government Movement [88]. The advocates of the movement aim to provide better access and improve the delivery of government services to their citizens by publishing their data on the web. By doing so, they encourage others to use government data to create useful civil services. To make data more accessible and reusable and, thus, more valuable, Open Government Movement formulated a set of principles of how to publish government data [85]. In short, they claim that the government data should be released under open license and made available in such a format (preferably non-proprietary) that machines can process it. Currently, the most popular way of making government data available on the Web is to publish it in the form of documents, be it human-oriented HTML Web pages and `pdf` documents, `xml` and `csv` files for data exchange purposes or documents in the proprietary `xls` format.

Pordata publishes statistics in the form of summary tables of different statistical variables over time embedded in the HTML Web pages. Alternatively, the data can be downloaded as `pdf` files. Such representations of statistics can be read by humans, who are able to understand and analyse them. When machines process HTML tables or `pdf` files they can only acknowledge instructions on how to compose the content of the HTML documents into human-readable Web pages (i.e., HTML tags), or how to layout information within different elements such as paragraphs and sections. For example, where we see statistics about air temperature in Portuguese cities, machines can only see a table, columns and rows. But when one wants to reuse Pordata data to build applications upon it, combine it with other datasets or visualise, having the data in human-readable formats complicates and makes this task hardly achievable. Additionally, Portuguese statistics can be exported in `xls` files. This data format (as well as `xml` and `csv`) gives structure to the data and makes it possible to process the data by machines and reuse in applications, but the data remains locked in static documents.

For example, if an entry was added to statistics about air temperature for new years that were not presented before, people have to re-download corresponding `xls` files and figure out what has changed. Moreover, there is no way to download only parts of spreadsheets. For example, when we are interested in air temperature in Lisbon only and don't want to download all the available statistics about other cities. Having data in `xls` does not address the issue of combining Pordata with other datasets. Different data providers adopt different formats and standards to publish their data, which creates the biggest challenge for connecting these disparate datasets. Even if we want to integrate Pordata in `xls` format with other data that resides in `xls` documents, there is no relation between information in two different spreadsheets. Machines see data in `xls` as cells, there is no explicit semantics attached to data that would facilitate machines to combine two pieces of data together.

With the traditional approach to publish data in documents, be it HTML tables or structured `xls` files, the data remains locked in documents, which makes it hard for machines to process it. Tim Berners-Lee, who is involved in the work done by the U.K. government, wrote a design issue note [16] on how to facilitate reuse of government data by "unlocking" it and presenting it in a *machine-readable* format, i.e., the format that allows machines to process the data and extract its meaning. He proposed to adopt the *Linked Data principles* [15] for the task of publishing government data. The Linked Data principles explain how to identify, access, describe and interlink data on the Web using existing Web standards, such as the HyperText Transfer Protocol (HTTP), Uniform Resource Identifier (URI) and the Resource Description Framework (RDF). To be able to capture semantics of data, Linked Data extends the scope of the traditional Web from documents to encompass concepts of the real world which make up the data. For example, statistics about population in different Portuguese cities may contain the following real-world concepts: *population*, *city*, *Lisbon*, *Porto*, etc. Each concept is then described in the form of simple sentences, e.g., "Lisbon is the capital of Portugal", "The population of Lisbon is 545,245", etc., in such a way that machines comprehend the meaning of the concepts. It might be the case that the concept of *Portugal* is defined in another dataset, then the sentence "Lisbon is the capital of Portugal" plays role of a link that connects data from two different datasets. As opposed to the hypertext links that connect documents on the Web, such link not only connects datasets two datasets together, but defines the semantics of the connection, e.g., from the link we know that one piece of data "is the capital" of another piece of data.

Over the past four years, a huge collection of Linked Data from different domains emerged on the Web, including geographic, life-science, libraries, publications, media and government data. This corpus of diverse semantically rich machine-readable

data is called *Semantic Web* or a *Web of Data*. The bootstrapping of the Web of Data was due to the activities of the *Linking Open Data* (LOD) [76] collaborative effort. The LOD participants locate data published using open standards and open licenses, i.e., *Open Data*, and republish it as Linked Data on the Web. As the result of their activities, plenty of Open Data sets were converted into the machine-readable format such as Wikipedia, that was republished as DBpedia datasets [39], Geonames [50] and MusicBrainz [69]. The success of Linked Data can be evidenced by an increasing number of data published directly by data providers such as the BBC [6–8] and The New York Times [84]. The Web of Data gives rise to the next generation of Web-based applications [56]. These applications take advantage of machines that are able to extract the semantics of information on the Web of Data and process it in a manner similar to human reasoning and inference. Examples include Semantic Web search engines that extend the capabilities of traditional keyword based search. For instance, Semantic Web search engines [38,45,111] allow users to obtain more accurate search results according to specific criteria. When we ask traditional search engines a query phrase like "birthday of Armstrong, the American downhill skier", they use the keyword based techniques to parse the phrase and single out several keywords, e.g., "birthday", "Armstrong" and "skier". Then for each keyword they deliver the documents that contain combinations of these words. Thus, among relevant information about Armstrong, the downhill skier, we may end up with a lot of inappropriate data including birthdays of different people, data about other famous Armstrongs or information related to other skiers. Semantic Web search engines are able to interpret the meaning of information on the Web of Data. They can understand that some documents talk about Armstrong who was the famous jazzman not the downhill skier, or that some documents talk about downhill skiers, but who have nothing to do with Armstrong. Such documents are discarded by the Semantic Web search engines.

The Linked Data standards align well with the Open Government Data principles. Moreover, the UK [36, 51] and US governments have already recognised the potential of Linked Data to publish their data in a format that facilitates further data reuse and integration. Interested communities and single individuals started using their to fulfil different purposes, and numerous applications have already been built upon the U.K. and U.S. government Linked Data. For example, enthusiasts from the Open Knowledge Foundation (OKF) [86] use the U.K. government data to create interesting mashup applications and visualisations in their desire to promote societal benefits of Open Data and open government data particularly. Among their work is the application "Where Does My Money Go?" that visualises the U.K. government spending and

helps to understand where the money of the UK taxpayers' goes[3]. Another interesting application they developed was built upon the Eurostat data, the Europe's Energy Dependency[4]. It aims to help to compare different European countries in terms of their carbon emissions, renewable energy share, net imports, and other energy related indicators. Another organisation, the Tetherless World Constellation (TWC) [114], focuses on the U.S. government data [115]. Among their works, an interesting example of visualising and mashing up statistics about smoking rate, tobacco policy and cigarette tax by state[5]. It allows to examine the correlation of these statistical variables over time.

In our work we define a proposal for publishing Pordata statistics as Linked Data to allow machines to process and "understand" them. We demonstrate an example of interesting usage of the Linked Data version similar to the works of OKF and TWC discussed above. For this, we show how information from the Web of Data can be used to enrich Portuguese statistics with data from other datasets. With our work we hope to contribute to the Pordata project and the LOD movement, particularly, to the presence of Portuguese statistics on the Web of Data. For this we:

- provide an overview of the Linked Data technologies;
- develop a proposal for publishing Pordata as Linked Data;
- develop a Web application that performs integration of the Linked version of Pordata with other LODsets, visualises the enriched Pordata data and provides new abilities to analyse it.

1.2 Document Outline

The remainder of the thesis is structured as follows. In Chapter 2 we introduce the Pordata project. Chapter 3 provides and introduction to the Linked Open Data project. We define basic notions of Open Data and Linked Data and discuss major participants of the LOD project. We also present an overview of the LOD activities including developed LOD driven applications. Chapter 4 describes of the proposal for publishing Pordata as Linked Data. We define the methodology and explained how we followed it to transform Pordata into Linked Data. We introduce relevant technologies when needed including URI, HTTP and RDF. To highlight the advantages of the Linked Pordata in Chapter 5 we define two use cases of analysing Pordata statistics that can not be handled with the current Pordata. We demonstrate how we accomplished the use

[3] http://wheredoesmymoneygo.org/
[4] http://energy.publicdata.eu/ee/index.html
[5] http://logd.tw.rpi.edu/demo/tax-cost-policy-prevalence

cases with the Linked Data standards by setting links to other data sources on the Web of Data and reusing information from them to enrich the Pordata data. At the end of the chapter we introduce a demo application that performs data integration and provides means to analyse Pordata in new ways, including the use cases. In Chapter 6 we draw the conclusions and discuss possible future work.

2

Pordata

In this chapter we examine the Pordata project that publishes statistics about Portugal. We start by presenting the goals of the project in Section 2.1. In Section 2.1.1 we study the structural organisation of the statistics on the Pordata website. Section 2.1.2 presents the website's functionalities. Since we do not have a direct access to the Pordata databases, we consider the functionalities of the website that can be used to get the data and transform it into Linked Data. We also inspect the options that the website provides to analyse statistics.

2.1 The Pordata project

Pordata is an initiative of the *Francisco Manuel dos Santos Foundation*[1] (FFMS). The Foundation has as its aim to promote and improve knowledge of the Portuguese reality, and this project reflects the main priority for the first years of the Foundation's activities. Pordata is a public service, a website[2], that accumulates and organises statistical data of the various areas of Portugal and Portuguese society.

According to the Pordata project's presentation given on its website, the motivation of the project was *"to respond to the need for credible information, which is often scant and difficult to access for a wider public, regardless of that public's ability to deal with statistics"*. On the Pordata's page dedicated to the FFMS's presentation it says - *"The Foundation will collect and organise the available information, making it as clear and accessible as possible.*

[1] `http://www.ffms.pt/`

[2] The Pordata website is available at `http://pordata.pt/`.

It will be available in its entirety at no cost to the user.". We can summarise the goals of the Pordata project as follows:

- collect statistical data from various official bodies to provide *credible* statistics;
- make the collected statistics *free* and *easily accessible* on the Web;
- present statistics *in its entirety*, but in a way that is comprehensible for a wider public.

In the following sections we discuss how the project realises its goals through the Pordata website by organising the data in a way that is convenient for exploration, presenting the data in its entirety and adding metadata to correctly interpret it. We also present different website's functionalities to access and analyse the data.

2.1.1 The Pordata data inventory

Hierarchical organisation of Pordata

The goal of the Pordata project is not just to publish raw statistical data, but also to make it easily accessible. For this, a special care is taken to organise the data on the website in the most convenient way for navigation and exploration.

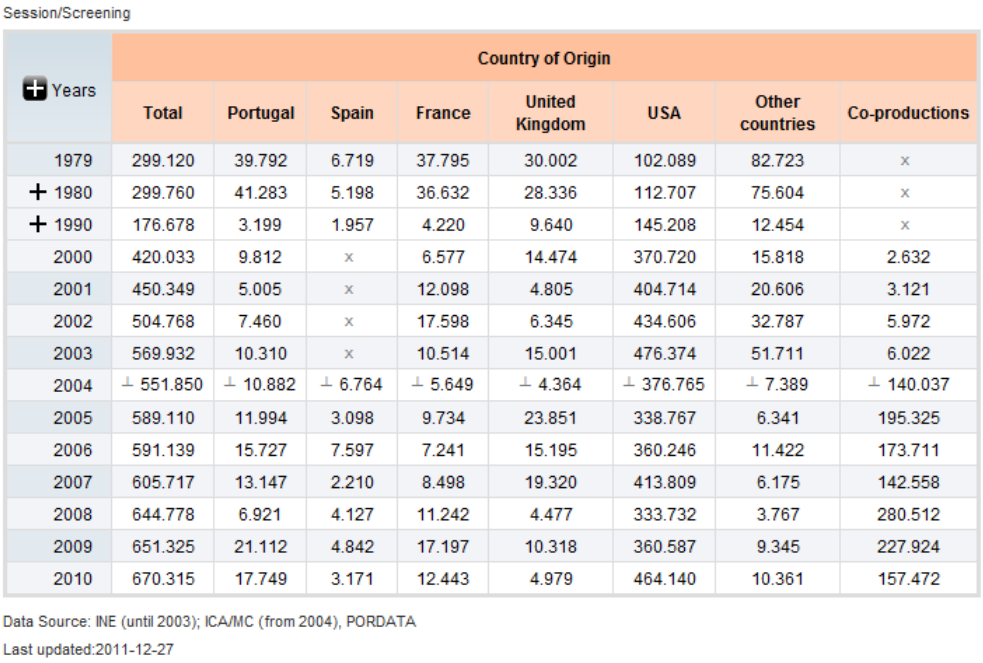
Data in the Pordata website is divided into *Themes*. A wide range of themes is available, e.g., *Population*, *National Accounts*, *Culture and Sports*, *Health*, *Education*, etc. The themes are divided further into *Sub-Themes*. Each sub-theme is composed of *Tables*. The hierarchy is as follows:

$$Theme \rightarrow Sub\text{-}Theme \rightarrow Table.$$

The primary way of representing statistics on the website are tables. Figure 2.1 depicts a screen shot of the table with statistics about screenings of movies from different countries. The statistics is taken from the theme *Culture and Sports* and the sub-theme *Cinema*. For the sake of brevity we will use the following pattern to locate a Pordata table:

<Theme> / <Sub-Theme> / <Table>

There are horizontal and vertical decompositions of the Pordata tables. A horizontal one is done by *years* for which statistics are available. According to the information on the Pordata website, statistics are related to time periods "which begins, wherever possible, in 1960 and continues to the present day." A vertical decomposition of the tables is due to statistical series. Each table can have one or many series, and each series,



Session/Screening

Years	Country of Origin							
	Total	Portugal	Spain	France	United Kingdom	USA	Other countries	Co-productions
1979	299.120	39.792	6.719	37.795	30.002	102.089	82.723	x
+ 1980	299.760	41.283	5.198	36.632	28.336	112.707	75.604	x
+ 1990	176.678	3.199	1.957	4.220	9.640	145.208	12.454	x
2000	420.033	9.812	x	6.577	14.474	370.720	15.818	2.632
2001	450.349	5.005	x	12.098	4.805	404.714	20.606	3.121
2002	504.768	7.460	x	17.598	6.345	434.606	32.787	5.972
2003	569.932	10.310	x	10.514	15.001	476.374	51.711	6.022
2004	⊥ 551.850	⊥ 10.882	⊥ 6.764	⊥ 5.649	⊥ 4.364	⊥ 376.765	⊥ 7.389	⊥ 140.037
2005	589.110	11.994	3.098	9.734	23.851	338.767	6.341	195.325
2006	591.139	15.727	7.597	7.241	15.195	360.246	11.422	173.711
2007	605.717	13.147	2.210	8.498	19.320	413.809	6.175	142.558
2008	644.778	6.921	4.127	11.242	4.477	333.732	3.767	280.512
2009	651.325	21.112	4.842	17.197	10.318	360.587	9.345	227.924
2010	670.315	17.749	3.171	12.443	4.979	464.140	10.361	157.472

Data Source: INE (until 2003); ICA/MC (from 2004), PORDATA

Last updated:2011-12-27

Figure 2.1: The screen shot of *Culture and Sports/Cinema/Screenings: total and by film's country of origin*.

in turn, can have one or many series' instances. For example, the table presented in Figure 2.1 has one statistical series: *Country of Origin*, which has 8 instances: *Total*, *Portugal*, *Spain*, etc. Another table, *National Accounts/Prices/Inflation Rate*, has two series: *Mainland* and *Portugal*, which have two instances each: *General Total* and *General Total (excluding Housing)*.

Figure 2.2 demonstrates the hierarchical representation of the Pordata data. The figure does not aim at representing all the available themes, sub-themes, tables, series and their instances. It rather gives an insight into the structural organisation of the data.

Metadata, Glossary and Symbology

Unambiguous interpretation of the data is another important aspect of the Pordata project. Thus, all the data available in the website goes along with metadata.

Metadata is provided for tables and series. Metadata for tables may contain definitions of the relevant concepts that are used in the table. Series metadata, among others, includes the unit of measure, an important information for correct interpretation of the statistics. For example, the metadata of the table from above and its series are illustrated by Figure 2.3.

Glossary contains definitions of concepts that are used in the Pordata tables.

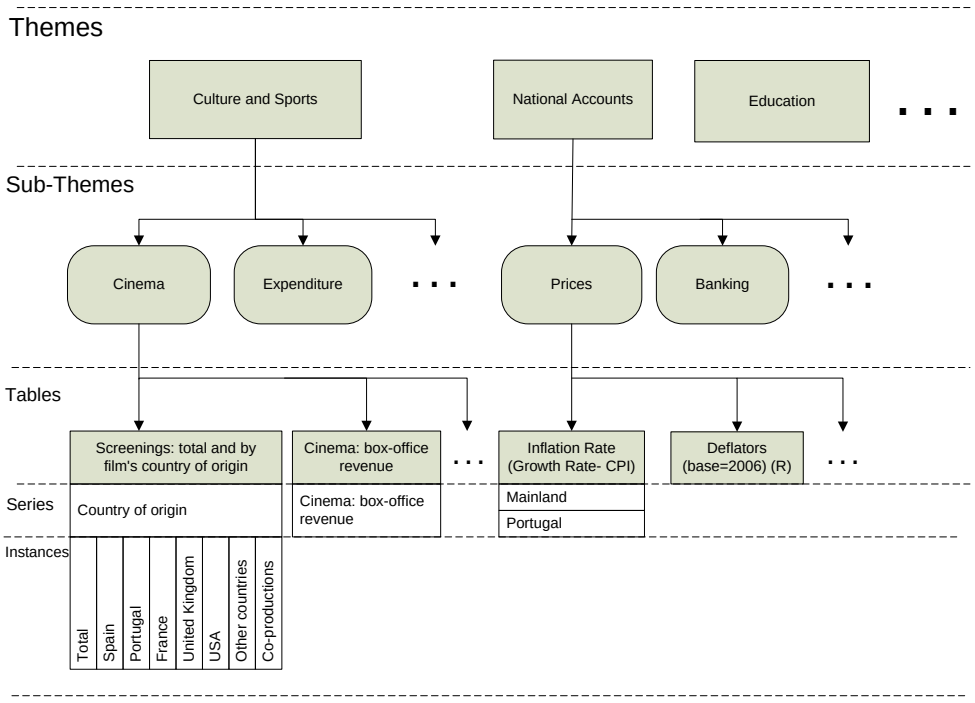
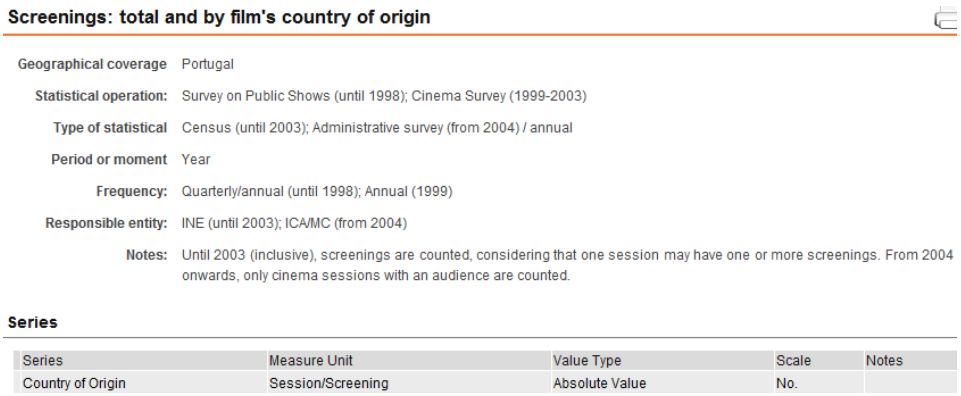


Figure 2.2: Hierarchical organisation of the Pordata data on the website.



Screenings: total and by film's country of origin

Geographical coverage	Portugal
Statistical operation:	Survey on Public Shows (until 1998); Cinema Survey (1999-2003)
Type of statistical	Census (until 2003); Administrative survey (from 2004) / annual
Period or moment	Year
Frequency:	Quarterly/annual (until 1998); Annual (1999)
Responsible entity:	INE (until 2003); ICA/MC (from 2004)
Notes:	Until 2003 (inclusive), screenings are counted, considering that one session may have one or more screenings. From 2004 onwards, only cinema sessions with an audience are counted.

Series

Series	Measure Unit	Value Type	Scale	Notes
Country of Origin	Session/Screening	Absolute Value	No.	

Figure 2.3: The metadata of *Culture and Sports/Cinema/Screenings: total and by film's country of origin*.

Symbology is a decoding of the symbols used in the Pordata tables to designate different types of values, e.g., "⊥" indicates *Series break*, ``x'' stands for *Not available* value, and so on. Symbology is common for all tables.

2.1.2 The Pordata website functionalities

We perform our analysis of the Pordata website functionalities bearing in mind the following three goals. First, we want to understand how the website's functionalities

support the structural organisation of the data on the website and provide easy access to statistics. For this, we discuss the *data search* functionalities. Second, with no access to the Pordata databases, we need to have a way to obtain the data from the website to be able to develop our proposal for transforming Pordata into Linked Data and demonstrate it in action. For this, we study the *data exporting* functionalities. Third, we want to examine the available options that allow data analysis, including *data editing* options, *alternative data views* and *formulae*.

Data search

The Pordata project was divided into three main phases with respect to the geographical scope of the statistical data. The first phase encompasses statistical data for Portugal, the second is dedicated to Portugal and the European Union countries and the third is for the Portuguese regions and municipalities. At the time we were writing this report only the first two phases were implemented, i.e., statistics about Portugal and the European Union countries were available on the website.

Starting from the homepage one can navigate either to the "Portugal Database" or to the "Europe Database". The following options are available then to search for the required statistics:

- *Search by the key-word*: the key-word search service is implemented on the "Homepage". It allows users to look for statistics that contains a specific term.
- *Search by theme*: each database presents the available themes from which a user can start the search by sequentially selecting sub-themes and tables.
- The *Search Environment* page: is an alternative way to search for the required statistics. One can directly navigate to this page and perform the same steps of selecting a theme, sub-theme and a table.

Data export

In addition to the main functionality of viewing statistics in tables, the website provides options to export the data. We are interested in them, because we need to get the data to develop our proposal for publishing Pordata as Linked Data.

The data can be exported from the website both as `xls` and `pdf` files. We evaluate each format taking into account the suitability for processing and interpretation by machines. For this, we use the term *structure* to refer to the the container that hosts the data, and consider two terms, structured and unstructured data [13, 83]. *Structured data* is the data represented in a precise format that provides metadata which can be

used directly by machines to derive semantic content of the data. A typical example of structured data is relational data, where data is grouped into semantic chunks (i.e., tables) each of which has certain semantics. Data in tables can be characterized by a set of attributes that can be processed by machines to derive the semantics of the data. Opposite to this, we call *unstructured data* the data that is represented in such a way that makes automated processing hard and requires humans to define semantics of the data. Examples of unstructured data include plain text, images and videos. For instance, the meaning of plain text is understandable for humans, but the data presented in a plain lacks of metadata that can be used by machines to automatically derive the meaning of the data.

Data in `xls` is structured. It is organised into separate spreadsheets. A spreadsheet contains multiple cells situated in a two-dimensional matrix consisting of rows and columns. Each cell has two attributes: the column's and row's names. We can employ heuristics that allows machines to process data represented in `xls` and extract its meaning based on the column's and row's names.

The `pdf` format is a de facto standard for representing printable documents on the Web. It defines a layout of different elements such as paragraphs and figures within a document to represent the data, but it does not provide a structure that would allow machines to interpret the data itself. In fact, the data in `pdf` files is given as plain text. Thus, we can conclude that data in `pdf` is unstructured.

Functionalities to analyse statistics

Data editing The options for data editing provide different opportunities to work with the original data represented in tables. This includes customization of the table view by dropping or adding series and years and constructing personalized tables. The series can be combined from different statistical tables into a single one.

- *Edit series and years* options are provided for each table, which give users an opportunity to add or drop series or years of a given table.
- *Add data option* allows adding series from other tables, i.e., merging tables into one personalized table.
- *Edit table* option brings a table into an edit mode, where a user can work with individual columns, i.e., represent them as charts, consult metadata for them or apply different formulae to statistics.

The data editing options provide a more flexible way of working with the Pordata statistics, as they allow users to bring together and analyse statistical series from different tables.

Alternative data views By default, statistics on the website are presented in the form of tables. Alternatively, each table can be seen as a static or dynamic chart. Static chart is the line plot. Figure 2.4 shows the line graph of the table *Culture and Sports/Cinema/Screenings: total and by film's country of origin*.

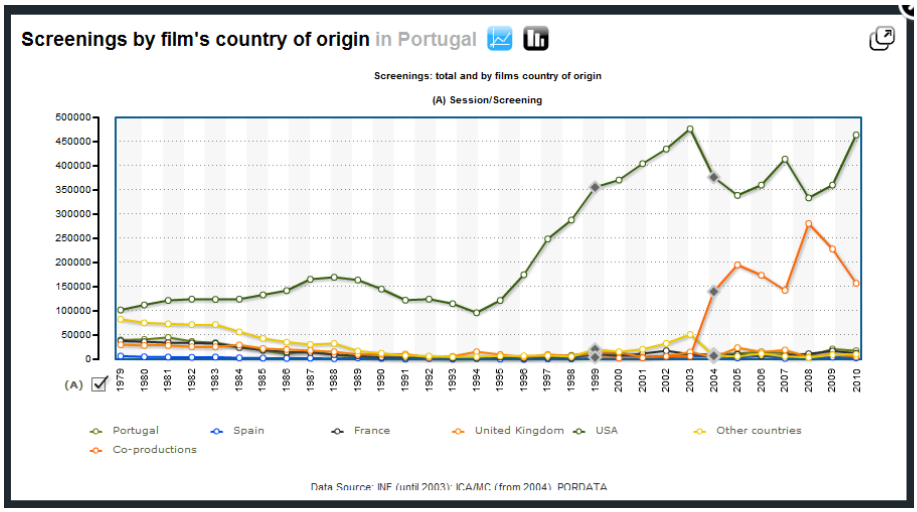


Figure 2.4: Static line graph of *Culture and Sports/Cinema/Screenings: total and by film's country of origin*.

Dynamic graphs can be a bubble or a bar graph representing statistics progressing over years. Figure 2.5 depicts the example table on a bar graph. One can see these statistics in progress over time by pressing the "play" button.

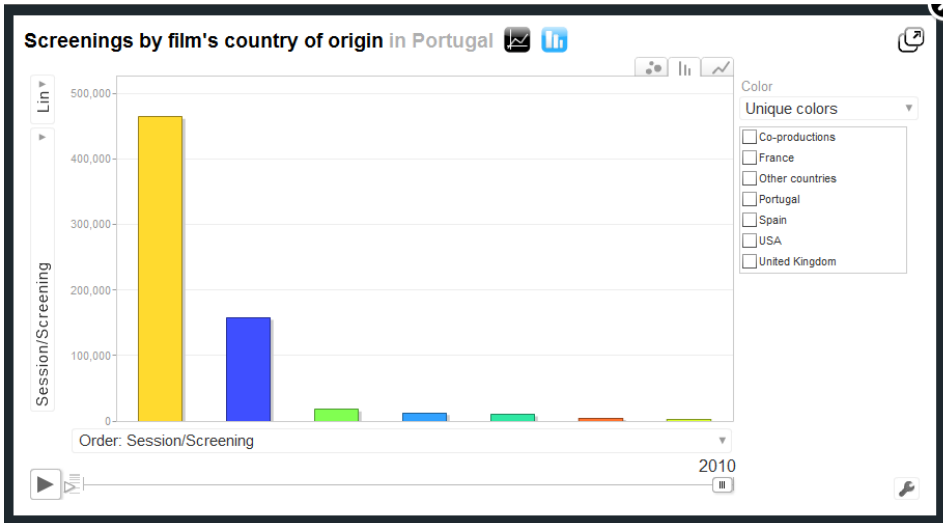


Figure 2.5: Dynamic bar graph of *Culture and Sports/Cinema/Screenings: total and by film's country of origin*.

Formulae There are also interesting options to analyse statistics when we view them in tables in the edit mode'. These options change statistics by applying relevant formulae to them:

- *Index numbers:* an amount measured in relation to the value of a particular base year, which is considered to be equal to 100.
- *Absolute variations:* absolute increase or decrease of statistical values for each year in relation to the previous one.
- *Rate of change (in %):* relative increase or decrease of statistical values for each year in relation to the previous one.
- *Constant prices (in euro).* When the unit of measure is the Euro, statistical values (i.e., prices) can be analysed without the effect of inflation.
- *Percentage:* proportion of statistical values in relation to the total, equivalent to 100.

We consider these options to be useful mainly for experts and researchers who perform specific studies of statistics.

Aggregators The information on the Pordata website is being aggregated into *Counters* and *Indicators*. The former represent a real-time simulation based on series of official statistics, e.g., counting *Resident population* or *Births*. Indicators provide information about important selected indicators, e.g., *Number of children per woman* or *Number of PhDs*.

All these additional aggregators can be consulted from the Pordata homepage[3]. The aggregators operate on the statistics that are already available on the website in tables, but they provide an alternative way of representing statistics, that is more attractive and understandable for ordinary people.

2.2 Chapter summary

In this chapter we provided an introduction to the Pordata project initiated by the Francisco Manuel dos Santos Foundation (FFMS). We determined the aim of the Foundation to promote existing information about Portugal and Portuguese society to the population. By doing so, the want to engage people into the public debate, so that they can contribute to the development of the democratic society, improvement of its public

[3] `http://pordata.pt/en/Portugal`

institutions and consolidation of its citizens' rights. With respect to this, FFMS defined the goal of collecting and disseminating among the population fundamental knowledge about Portugal: Portuguese statistics. We introduced the Pordata project that was established by FFMS with the objectives to organise and represent the statistics in a comprehensible way for ordinary people and make the statistics publicly available on the Web for free. Further in this chapter, we examined how the objectives of the project were realised on the Pordata website. We discussed the hierarchical organisation of the data into themes, sub-themes and tables and decomposition of tables into years and statistical series. This data organisation is effectively complemented by rich functionalities to explore the data on the website. Together they provide an easy access to numerous statistical datasets accumulated from different sources. We will use the results of this review of the structural organisation of Pordata further in Section 4.2.1, where we analyse the Pordata data and extract relevant concepts constituting the data.

When we discussed the website's functionalities we were pursuing different goals. One of the goals was to solve the issue of obtaining the data from the website in the light of having no access to the Pordata databases. For this, we examined the data exporting options. We determined that data exported in `xls` is structured, that is represented in a precise format that allows machines to process it. We will utilise Pordata `xls` files in our work in Section 4.3.1, where we use them to populate our database. We also wanted to understand what options are provided by the website to analyse statistics. We determined that the primary way of viewing statistics on the website is summary tables. There is a possibility to combine statistical series from different tables together and create a single custom table, but users are always left to work with raw statistical data. We argued that statistics represented in such a way are of interest mainly for experts; in fact, it is rather difficult for non-experts to understand them. Among the options that can be potentially interesting to non-expert users is the possibility to view data on static or dynamic graphs. The latter illustrate statistics progressing through time. Some pre-selected statistical data can be examined in different aggregators on the website's homepage, which we consider the most interesting way for ordinary people to view and comprehend statistics on the website. For instance, we can see the real time counters of deaths, births and migration balance in Portugal. We can think of these aggregators as representing statistics in context, e.g., in case of real time simulators it is the context of passing time.

Currently, Pordata statistics are presented as collections of statistical values, i.e., numbers. But statistics are not just numbers. Each statistical value is attached a set of attributes such as statistical series, the year in which the value was collected, unit of measure, etc. We can reuse this information to provide context to statistics, enrich

them and enable more powerful analysis of the data. For example, in the statistics about population in different Portuguese cities we have a geographical dimension. We can combine these statistics with geographical data, i.e., geographical locations, and plot population per cities on a map. Having statistics on a map gives a comprehensible visualisation of the data and a good overview of the whole picture. It also allows to find possible correlations between population and types of land, e.g., coastal or mountain land. In general, any Pordata statistic can be represented and analysed in different interesting ways. We believe that providing context to statistics make them more useful and comprehensible for ordinary people. This would help to accomplish the goals of the Pordata project to disseminate statistics among the population. In Section 3.3.2 we overview existing works that address the issue of statistical data contextualisation, including visualisations and mashups of statistical data provided by the U.K. and U.S. governments. In Chapter 5 we present our application that mashups and visualises Pordata statistics and provides means to analyse it in new ways.

3

Linking Open Data project

In this chapter we aim to communicate a general idea of what Linked Data is and describe progress to date in publishing Linked Data on the Web. We start by defining the notions of Open Data and Linked Data in Sections 3.1 and 3.2 respectively. In Section 3.3 we introduce the Linking Open Data project. To give an idea about the scale of the LOD movement, we discuss its major participants by domain in Section 3.3.1. We review applications that have been developed to exploit the Web of Data in Section 3.3.2.

3.1 Open Data

Even though the term *Open Data* is currently in frequent use, there is no commonly agreed definition. We will use the one provided by the *Open Knowledge Definition (OKD)* project [87]. OKD considers data as any kind of content from *"sonnets to statistics, genes to geodata"*. According to the OKD definition, data is open *"if anyone is free to use, reuse, and redistribute it - subject only, at most, to the requirement to attribute and share-alike"*. This definition considers three aspects of data openness: social, technological and legal. Social openness means that the data must be accessible as a whole, and not only few items of it at a time (e.g., by downloading). By technological openness OKD considers absence of any technological obstacles to access and reuse the data (e.g., no access control and open formats). Legal openness is established by open data licenses.

Open Data licensing Open licenses meet the requirements of Open Data and grant permissions to access, reuse and redistribute data with few or no restrictions. In order to enable people to use the data on the Web on a secure legal basis, one needs to explicitly state which license applies to your data. It is important to apply open data licenses simply for the sake of clarity. Without a license it is not clear if the data can be used, reused and distributed by others. Examples of open licenses include Public Domain Dedication and License (PDDL) and Attribution License by the Open Data Commons project, the GNU Free Documentation License and the licenses prepared by the Creative Commons Attribution project, such as Creative Commons Attribution Share-Alike (cc-by-sa). These and other open data licenses can be found in [90].

Open Data sets There are various interesting open data sets available on the Web. A well-known example of open data is *Wikipedia*. Most of the Wikipedia's text and many of its images are under open licenses. Other examples of open data are *Wikibooks*, *Geonames*, *MusicBrainz*, *WordNet* and *the DBLP bibliography*. *Google Maps* is an example of data that is not open, since the geodata is currently proprietary (copyrighted or protected by DB rights). The CKAN catalogue [26] can be consulted for finding information about existing open data sets.

In general, Open Data can come from anywhere. One of the biggest source of Open Data is the government domain. *Open Government Data* is a global movement of governments starting to open their information from public sector. The pioneers were the governments of the U.S. [36] and the U.K. [51], and many more governments have already joined the movement, including Australia, Netherlands, Spain, Austria, Denmark.

3.2 Linked Data

The term Linked Data was coined by Tim Berners-Lee in 2006 in his design note [15]. He outlined a set of recommendations, referred to as Linked Data principles, on how to complement the current Web of human-oriented documents with a Web of machine-enabled data. Tim Berners-Lee proposed to apply the same ideas that are successfully used for making the current Web, to publish and interlink data in such a way that machines can also process it and extract its meaning, i.e., build a Web of Data.

3.2.1 The Linked Data principles

The Linked Data principles in its original reading are as follows [15]:

1. *Use URIs as names for things*

2. *Use HTTP URIs so that people can look up those names.*

3. *When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)*

4. *Include links to other URIs. so that they can discover more things.*

The Linked Data principles in a nutshell The classic Web provides humans with data. It can be any kind of data, e.g., air temperature, somebody's personal profile, official government reports, etc. The data is represented in the form of documents (e.g., HTML Web pages). Documents can be processed by machines that can "understand" the structure of the documents (e.g., paragraphs, titles, tables, etc.) and present them in a more convenient way for humans to read and comprehend the data carried out by the documents. However, the meaning of the data is mostly given as a plain text and hidden from the machines. It is hard for software applications to extract semantics from HTML pages. To allow machines to "understand" the meaning of the data, we need to be able to describe not only the structure of the documents, but the data itself. At its most basic, data is made up of any kind of things that exists in the world, i.e., real-world objects (e.g., people, countries, building, etc.) or abstract concepts (e.g., air temperature, the fact of knowing somebody, etc.). Thus, the Web of Data extends the scope of the traditional Web from documents to encompass real-world objects and abstract concepts.

The first and second Linked Data principles define a mechanism to name any thing that exists in the world using the existing Web standards, such as URI and HTTP (we will discuss URIs and HTTP technologies in Section 4.2.1), and make them accessible on the Web. For example, by using the Linked Data principles one can give a name the concept of city of Lisbon and publish it on the Web. Note that the city of Lisbon and its homepage are not the same concept. For this reason they must be named with two different URIs.

The conventional Web has its standard way to describe documents on the Web, i.e., HTML is used to create Web pages. The third Linked Data principle recommends to use the Resource Description Framework (RDF) to describe things in the world in a machine-redable manner (we will look in more detail into RDF in Section 4.2.3). In RDF one can provide descriptions of real-world concepts in the form of sentences. For example, one may describe the Lisbon city as follows: *Lisbon is the capital of Portugal*; *Lisbon has population 545,245 people*, etc.

Finally, links are an integral part of a Web. Documents on the classic Web are connected by means of the hypertext links. Similarly, the fourth Linked Data principle claims to connect data on the Web by setting links to data from other data sources. Links on the Web of Data are defined using RDF and referred to as RDF links (we will discuss RDF links in Section 5.1.1). Unlike the hypertext links, the links on the Web of Data not only connect two pieces of data together, they also provide semantics for this connection. The RDF links look like sentences as well, just involve concepts that were defined by different people. For example, if you define the concept of the city of Lisbon and discover that somebody else in another dataset defined the concept of Portugal, then the following link can be set *Lisbon is the capital of Portugal* (where Portugal is the concept from that external dataset). Thus, you can connect your data with this external dataset.

Linked Data vs Linked Open Data The fact that Linked Data is defined in a personal note of Tim Berners-Lee and is not formally endorsed by W3C contributes to the ambiguity of the definition of the concept of Linked Data. The discussions regarding this topic generally come down to which technology is used to represent data. Some people argue that RDF is integral to Linked Data, others suggest that, while it may be desirable, use of RDF is optional rather than mandatory. Some reserve the capitalized term *Linked Data* for data that is based on RDF, preferring lower case *linked data*, or *linkable data*, for data that uses other technologies. In our work we will use the term *Linked Data* to refer to data published on the Web in accordance with the Linked Data principles. We will also stick to the most common opinion that RDF is a standard for representing Linked Data. When we want to emphasize that the Linked Data principles were applied to Open Data we will use the term *Linked Open Data*.

3.3 Linking Open Data project

The LOD project [76] began in 2007 with the support and sponsorship of the W3C Semantic Web Education and Outreach Group (SWEO) [108]. The goal of the project is to bootstrap the Web of Data. Initially, the project was driven mainly by researchers in university research labs and Web enthusiasts, whose aim was to identify Open Data and serve it as Linked Data on the Web.

Since 2007 the project has grown considerably due to the involvement from large organizations from different domains. According to the latest statistics of September 2011, the scope of the Linking Open Data project included 295 datasets. Figure 3.1

demonstrates the range and scale of the Web of Data originating from the LOD project[1].

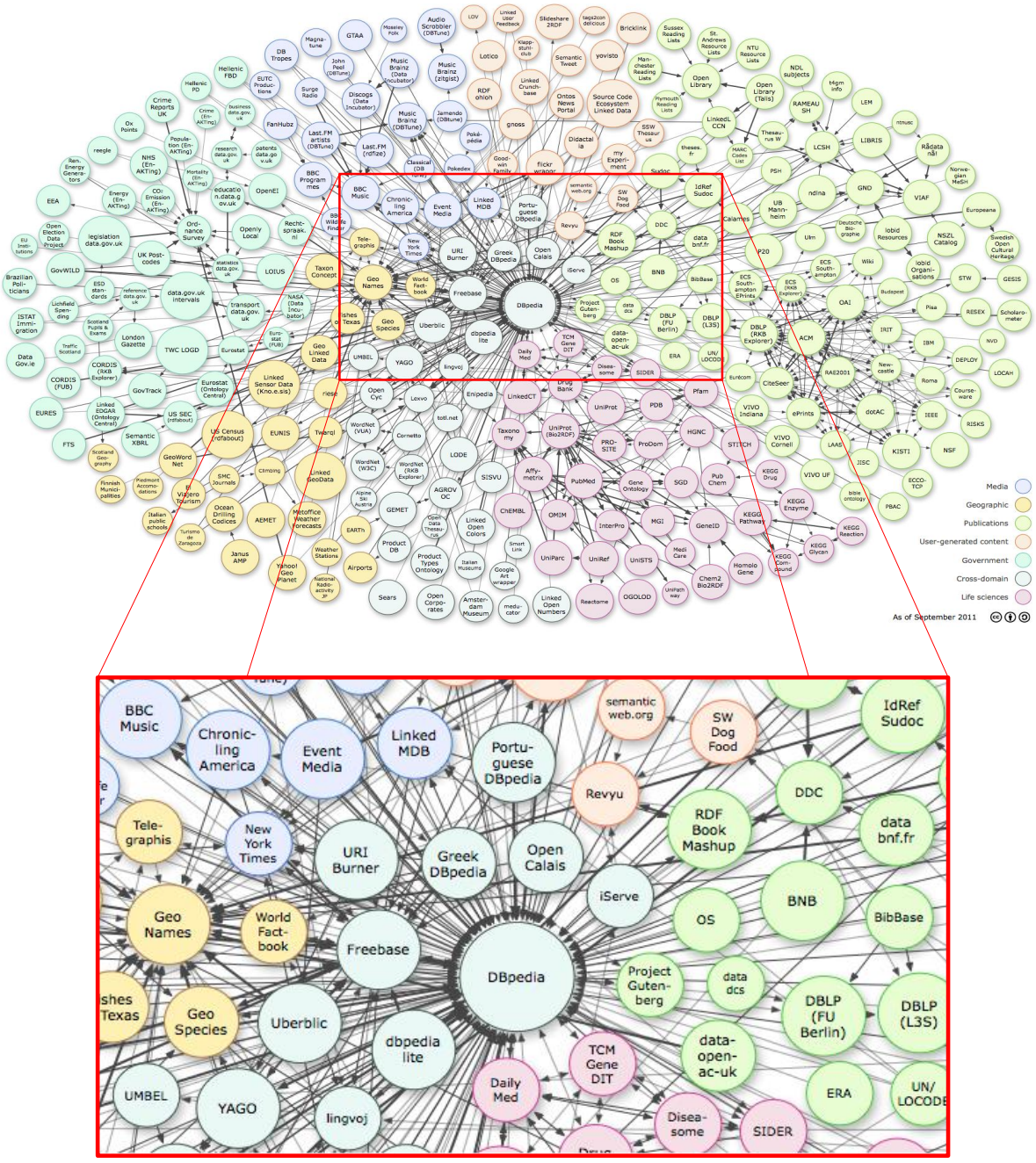


Figure 3.1: LOD cloud diagram as of September 2011.

Each node in the LOD cloud diagram represents a distinct Linked Data set. The arcs indicate the connections between datasets. The heavier the arcs are, the bigger number of links exist. Bidirectional arcs represent links in both directions. For keeping the

[1]The LOD cloud diagram is made by Richard Cyganiak and Anja Jentzsch and is available at [32]. An interactive version of this diagram can be found at [31].

LOD cloud diagram up to date, the Linking Open Data community effort maintains a catalog of known Linked Data sources, the LOD Cloud Data Catalog [73].

3.3.1 Linked Open Data by Domain

The content of the LOD cloud comprises data from many different domains: cross-domain, media, geographic, government, publications, life science and user content. Figure 3.2 depicts a diagram[2] that shows the distribution of data by domain as of September 2011.

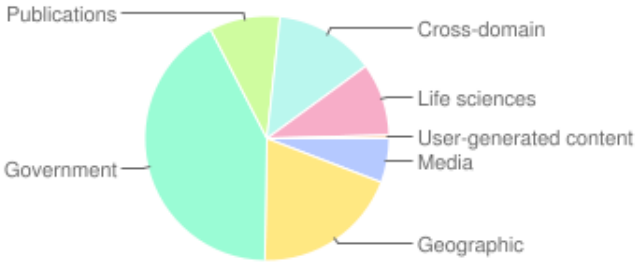


Figure 3.2: Distribution of data by domain, September 2011.

Cross-domain Some of the first datasets that appeared in the Web of Data are not specific to one topic and provide information that span multiple domains. Perhaps, the most popular cross-domain data source is *DBpedia* [39]. DBpedia is a community effort to automatically extract information from Wikipedia and make it available as Linked Open Data on the Web. Thus, DBpedia covers a wide range of topics and describes more than 3.64 million things, including people, places, music albums, films, video games, organizations, species and diseases in up to 97 different languages [39]. Thus, there is a high degree of conceptual overlap between DBpedia and many other LODsets. The DBpedia community aims to provide a "linking hub" for the other LODsets and dedicates a particular effort to establish links to the existing data sources on the Web of Data.

Freebase [49] is a second major source of cross-domain data. Freebase and DBpedia are very similar in a sense that they both extract data from Wikipedia and publish it as Linked Open Data on the Web. However, Freebase imports data from other Open Data sets, including Geonames [50] and MusicBrainz [80], and does not focus on Wikipedia only. Moreover, Freebase allows users to edit the data in a similar fashion as Wikipedia articles can be edited now, whereas DBpedia requires users to edit Wikipedia, first, so that changes appear in DBpedia.

[2]The diagram is taken from [19], Version 0.2, 02/16/2011.

Geographic Another factor that can be considered in other domains is geography. Two large Open Data sets were already converted into LOD: Geonames and OpenStreetMap. *Geonames* [50] provides LOD descriptions of over 10 millions of geographical locations worldwide, including their names, geo coordinates and general information about the places, such as spoken language, population and the type of settlement. The *LinkedGeoData* [70] project publishes *OpenStreetMap* [92] as LOD in an effort to add a spatial dimension to the Web of Data by publishing information about more than 350 million places and allowing users to define their own descriptions of places (e.g., describe a shop as accessible via a wheel chair).

Locations in Geonames and LinkedGeoData are interlinked with corresponding locations in DBpedia and Freebase, and both geographical data sources often serve as hubs for other data sets.

The LOD initiative attracted two official spatial data providers: British Ordnance Survey and National Geographic Institute of Spain (NGIS) who published their data as LOD on the Web. *British Ordnance Survey* [93], the national mapping agency of Great Britain, published topological information about the U.K. administrative areas in efforts related to the U.K. government open government initiative. The project that aims to enrich the Web of Data with Spanish geospatial data, *GeoLinkedData.es* [70], started by publishing diverse information sources belonging to NGIS.

Media A major Linked Data publisher in the media industry is the British Broadcasting Corporation (BBC). The first attempt of BBC to adopt the Linked Data principles relate to the catalogue of the *BBC programmes* [7], that contains descriptions of every episode of every radio and TV program. This catalogue link to the *BBC Music* website [6] that publishes Linked Data about every artist whose music was played on BBC radio stations. Both BBC programmes and music sites were connected with MusicBrainz and DBpedia. More recently, BBC launched the site *Wildlife Finder* [8], which presents information about animal species, behaviours and habitats as Linked Data. Wildlife Finder was also connected to DBpedia and the BBC Programmes that depict the corresponding animals or their habitats. The *BBC's World Cup 2010 Web site*[3] provides information about games, players and other World's Cup related news. The information on this website was dynamically published as Linked Data, which enables its automatic management on the website.

Another major publisher of Linked Open Data in the media sector is the *New York Times* [84] that posted a significant portion of its internal subject headings as LOD and interlinked these topics with DBpedia, Freebase and Geonames.

[3] `http://news.bbc.co.uk/sport2/hi/football/world_cup_2010/default.stm`

Linked Movie Database [71] is the project that aims at publishing various movie related information as Linked Open Data, including links to DBpedia and Geonames, as well as references to related Web pages (such as IMDB, Rotten Tomatoes and Freebase).

MusicBrainz [80] is an Open Data set with information about artists, their recorded works, and different connections between them. *LinkedBrainz* [69] is the project that publishes MusicBrainz as Linked Open Data.

Publications Cultural institutions such as libraries, museums and archives started to understand the potential of the Linked Data technology for bringing together their data, link it to other relevant LODsets and enhance their public services. Examples include the *German National Library* [81], *LIBRIS* [68], Swedish National Union Catalogue and the *American Library of Congress* [1]. Similarly, the *OpenLibrary* [89], a collaborative effort to create *"one Web page for every book ever published"* and publish its catalogue as LOD.

Scholarly articles are represented on the Web of Data by *DBLP as Linked Data* [37], database with bibliographic information on major computer science journals and conference proceedings. The *ReSIST project* [99] publishes bibliographic databases such as the IEEE Digital Library and CiteSeer as LOD.

Life science Linked Data has gained significant uptake in the Life Sciences as a technology to connect various datasets that are used by researchers in this field. In particular, the *Bio2RDF* [12] project has published a more than 30 different biological databases as LOD, including UniProt (the Universal Protein Resource), Entrez Gene (database of gene-specific information), KEGG (the Kyoto Encyclopedia of Genes and Genomes), CAS (the Chemical Abstracts Service), PubMed, and others.

User-generated content Some of the earliest datasets on the Web of Data were based on conversions of the existing web sites with user-generated content, such as DBpedia, that was described earlier, and the *Flickr* photo-sharing service. For the latter a Linked Data wrapper was developed [9] that extends DBpedia with photos from Flickr. Additionally, this work was complemented by user-generated content sites that were built with native support for Linked Data, such as *Revyu.com* [58] and *Semantic MediaWiki* [104].

The latest uptake of the Linked Open Data technology was done in the social media. Facebook released the *Open Graph Protocol* [91] that exploits the Linked Data principles. With this protocol data publishers can describe information on the Web pages so that machines understand it. Open Graph Protocol is currently consumed by Facebook, Google, IMDb, Microsoft.

Government Due to the Open Government Movement (Section 3.1), large collections of public sector information from different governments worldwide were published as Open Data. Governmental data is diverse. It can include different kinds of official statistics, registers of companies and land ownership, reports on the performance of schools, voting records of elected representatives and governmental expenditure.

The U.K. and U.S. governments have recognised the great potential of the Linked Open Data standards to serve the goals of the Open Government Movement. The U.S. government *data.gov* [36] in collaboration with the Tetherless World Constellation [114] adopted the LD principles to publish various government datasets that cover a wide range of topics including government budgets, environmental statistics, housing and population statistics, medical cost, energy consumption, public library statistics, labor statistics, and etc. *data.gov* is an example of a repository that provides access to heterogeneous data distributed over several areas of interest with an aim to "empower" the U.S. population. The U.K. government announced its *data.gov.uk* [51] site in 2010 that contains geo-spatial information (the *British Ordnance Survey* [93] discussed above), transport and legislation information and information about government spending (the *Combined Online Information System (COINS)* [28]). The U.K. government aims to be a responsible publisher of Linked Open Data [65]. For this, they develop publishing patterns and guidelines related to different LOD publishing issues. Due to the both governments' activities, data from the government domain constitute now more than 40% of the Web of Data.

Statistics The first indication of statistics on the Web of Data appeared with *The 2000 U.S. Census* [61] that includes geographical data and detailed statistics collected in the U.S. in 2000.

An attempt to convert the *World Factbook* [44] was done in the Free University of Berlin, who also adopted the Linked Data principles to publish *Eurostat* [43]. Another attempt to expose Eurostat as LOD belongs to *riese* [100], RDFizing and Interlinking the EuroStat Data Set Effort.

LOD associations in different countries republish statistics from the National Statistical Institutes such as Italy [74], Scotland [75] and Spain [40].

3.3.2 The LOD project activities

The next step of promoting Linked Data concerns demonstration how Linked Open Data can actually be used. The Linked Data standards represent the data in a structured machine-readable way with explicitly defined semantics. This gives new opportunities to work with data. Compare, for example, the way the number *2.3* is repre-

sented in an `xls` file and as Linked Data. In `xls` this number is not of a huge value for machines, they operate with it as with another cell, without differentiating it from the number *3.1* for example. With Linked Data we can provide a meaning to these numbers. For example, *The number 2.3 is the inflation rate in Portugal in 2005, when the president of Portugal was Jorge Sampaio, from the Socialist Party. The number 3.1 is the inflation rate in 2006, when the president Anibal Cavaco Silva from the Social Democratic Party was elected*. Now machines can use these numbers to analyse inflation rate in relation to the political situation in the country. The Web of Data contains a big collections of such structured machine-readable data interlinked to form a single global informational space. We can use the Web of Data to connect disparate data sources on the Web and develop new kinds of applications that operate upon such data, so called Linked Data driven Web applications [56]. As Linked Data is a relatively novel technology, the existing applications are mostly prototypes and will likely undergo significant evolution as lessons are learnt from their development and deployment. Nevertheless, they already give an idea of what will be possible in the future. The present Linked Data driven applications can be classified into generic applications, such as Linked Data browsers and search engines, and domain specific applications that cover the needs of specific user communities.

Linked Data browsers

In the classical Web of documents, browsers allow users to navigate between HTML pages by following untyped hypertext links, that can be understood by humans but are meaningless for machines, and, thus, can not be used to assist people in finding information and "guide" them through the Web in a smarter way. The Linked Data browsers are Web applications that provides interactive support for navigating through or exploring Linked Data. Examples of LD browsers include the *LinkSailor* [72], *Tabulator* [17, 111], *Marbles* [78] and *URI Burner* [116]. They can process the semantic links established between different data sources and facilitate users in exploring the Web of Data. For example, when a user is looking for the description of the city of Lisbon in DBpedia, Linked Data browsers can interpret the data available about Lisbon and present it nicely using conventional data presentation methods. Thus, the browsers can "understand" that Lisbon is a city and represent it on a map, as well as other notable locations in the city such as theatres, castles, shops, etc., can be "recognized" by the browsers as having geographical characteristics and represented on the map. Tabulator and Marbles can also merge data about the same concept from different data sources. They can discover that there are other data sources on the Web that also describe Lisbon, combine their data with the DBpedia data and present the

aggregated view to the user.

Linked Data search engines

The Linked Data search engines also take advantage of the ability of machines to process and extract the meaning of information on the Web of Data. The existing LD search engines provide richer interaction capabilities to a user and ensure more accurate search results, than classic search engines with a simple keyword-based search implemented. For example, DBpedia implements the faceted search paradigm [38] that allows users to filter search results according to specific criteria (facets), e.g., people who were born in a certain country or who have a specific profession. Thus, if a user searches for information about Armstrong, the bicyclist, the results can be narrowed to contain only data relevant to bicyclists, excluding those with data about Armstrong, the astronaut, or Armstrong, the jazzman, or other Armstrongs who were not outstanding bicyclists. While DBpedia implements an enhanced search over the DBpedia dataset together with information from interlinked datasets such as Geonames, Freebase and DBLP bibliography, the *Falcons* search engine [45] provides the same functionality at a Web scale.

Domain specific Linked Data applications

Domain specific Linked Data driven applications are those that reuse content of existing LODsets to fulfil different purposes. Numerous examples of such applications exist. The U.K. and U.S. governments are among the key institutions that have recognised the advantages in converting legacy data stores into Linked Data and making explicit links between these heterogeneous data sources. One direct benefit of Linked Data is richer government transparency: citizens can now participate in collaborative government data access, including "mashing up" distributed government data from different agencies, discovering interesting patterns, customizing applications, and providing feedback to enhance the quality of published government data. The Tetherless World Constellation (TWC) [114] investigates the role of Linked Data in producing, enhancing and utilising government data published on *data.gov*. For this, TWC develops visualisations and mashups of different government data, including financial data, spending, energy usage and public healthcare. Their works for consuming Linked government data demonstrate how the value of the data is increased in combination with other datasets. For example, one application utilise the U.S. government data spending on fire fighting to integrate it with data from DBpedia about number of fires and burned area in different years[4]. The application shows correlations between the gov-

[4] http://data-gov.tw.rpi.edu/demo/stable/demo-1187-40x-wildfire-budget.html.

ernment spending and the the actual fires. Another example combines the smoker rates statistics with data about population and cigarette taxes[5]. More works done by TWC can be found in [115].

The U.K. government gave rise to plenty of mashups and visualisations that show immediate benefits of the LD standards for citizens [52]. Among them are applications that provide information about local services, help managing finance and environmental issues. For example, the "Walkonomics"[6] application rates and maps the pedestrian friendliness of streets and urban areas combining government data with real people reviews. It allows to check a street by a post code and helps to understand how walkable it is. Another example is the "BUSit London" application[7] that reuses information about London buses and allows to plan a bus journey with several changes by indicating which buses to take, where to catch them and where to change. An interesting interactive visualisation of the U.K. government spending is developed by the Open Knowledge Foundation [86]. "Where Does My Money Go?"[8] represents spending by area and helps to understand where the money of the UK taxpayers' goes.

BBC is utilising the benefits of Linked Data as means for storing and sharing news. The BBC Programmes site [7] reuses information from other LODsets (e.g., DBpedia and Freebase) to identify and link semantically related information owned by the BBC to increase usability of their web pages and other applications that make use of it. The BBC Music site [6] is enriched with artists information from MusicBrainz [69] and artists' biographies fetched from DBpedia to compose introductory texts.

Talis Aspire [112] is a Linked Data driven application that helps educators to create and manage lists of learning resources, e.g., books, journal articles, Web pages. Users interact with the application via a conventional Web interface, while the data they create is stored as Linked Data. Aspire then uses the Linked Data principles to connect the learning resources with related data elsewhere on the Web and enrich the range of material available to support the educational process. This resource list management system is currently used by thousands of students at the University of Plymouth and the University of Sussex [20].

Another interesting application that reuses the DBpedia data is a generic reviewing and rating site, *Revyu*[9]. For example, when a film is reviewed on Revyu, the site attempts to provide more information about the film (e.g., the director's name and the film poster) by reusing the data from DBpedia.

An interesting mobile application was developed upon the DBpedia dataset, *DB-*

[5] http://data-gov.tw.rpi.edu/demo/stable/tobacco-smoker/demo-state-10026-smoke-rate-statevarsapi.html.
[6] http://walkonomics.com
[7] http://www.busitlondon.co.uk/
[8] http://wheredoesmymoneygo.org/
[9] The site is available at http://revyu.com/.

pedia Mobile [10]. This application helps tourists to explore a city by identifying their location based on the current GPS signal of the mobile device and rendering a map with indications of nearby interesting places.

One of the examples of Linked Data driven applications in the life science domain is the *NCBO Resource Index* [18]. It relies on LOD from different biomedical datasets on the Web of Data and supports researchers in exploring them.

Zemanta [126] is a content recommendation tool that reuses data from DBpedia and Freebase in order to help users to better organize their blogging activities. It suggests relevant links, articles or images while users write their blogs to make them more interesting and attractive for readers.

Google uses Linked Data describing people, products, businesses, organizations, reviews, recipes and events in its search results to provide them in the form of rich snippets [107]. It also uses the extracted Linked Data to directly answer simple factual questions such as the birth date or place of somebody. Google answers such queries not only with a list of relevant links, but it provides the actual answer.

3.4 Chapter summary

In this chapter we gave a general understanding of the concept of Linked Open Data. We started by clarifying the notion of Open Data. We identified that data is open if it is being published on the Web using open standards and under open licenses that allow other people to freely reuse, share, create derivative works and distribute the data on the Web. The term Linked Data refers to data published on the Web according to the Linked Data principles. These principles were first introduced by Tim Berners-Lee in his design issues note, where he proposed to adopt the existing Web standards (URI, HTTP and RDF) to publish and interlink machine-readable data on the Web, i.e., build the Web of Data. The new data publishing paradigm enables machine to process the data and the links between different datasets and extract their meaning. Linked Data by itself does not have to be open. Thus, we agreed that in our work we will use the term Linked Open Data when we want to emphasise the open nature of Linked Data.

Further in this chapter we introduced the Linking Open Data project that was created to bootstrap the Web of Data. The primary goal of the project is to identify the existing Open Data sets and publish them according to the Linked Data principles. As a result of the project's activities, hundreds of datasets have been published as LOD. The Web of Data currently covers a wide variety of topical domains, including government data, life science, media, library and publications, geographic and cross-domain datasets. We introduced the major participants from different domains. They include

DBpedia and Geonames that, due to their nature, play an important role of providing linking hubs on the Web of Data, since they have a high degree of topical overlapping with other LODsets. In the early stage the LOD community was composed of researchers and Web enthusiasts. Recently published datasets, such as the U.K. government data, the BBC sites and the New York Times data, demonstrate how the Web of Data is evolving from data published primarily by researchers, to data publication at source by large media and public sector organizations.

After a significant success in building the Web of Data, the LOD advocates focused on demonstrating how Linked Data can actually be used. We concluded this chapter by discussing the developed Linked Data driven applications. We classified the Linked Data driven applications into two categories: generic and domain specific applications. The existing generic Linked Data driven applications (e.g., data browsers and search engines) provide enhanced capabilities to explore and search data in comparison with the traditional Web browsers and search engines. They take advantage of the fact that data on the Web of Data can be processed and "understood" by machines. We discussed how the existing Linked Data browsers can aggregate data from different data sources and represent it to a user using the corresponding representation method (e.g., maps for cities). We examined how the present Linked Data search engines can improve the simple keyword-based search by filtering out irrelevant data. We concluded the chapter by introducing the existing domain specific Linked Data driven applications. The government domain becomes a popular source of data that underpins interesting mashup applications. The ongoing work of the Open Knowledge Foundation and the Web enthusiasts is dedicated to visualising and developing mashups of the government LOD that ease its analysis and understanding. Data from DBpedia, Freebase, Geonames and other LODsets becomes a main source for other Linked Data driven applications, such as the BBC music site, the DBpedia mobile tourists guide, the Revyu website with reviews and ratings, the recommendation system Zemanta and many others.

4

Publishing Pordata as Linked Data

In this chapter we present our proposal for publishing Pordata as Linked Data. We start by introducing the underlying methodology in Section 4.1. The rest of the work is divided according to the two main successive steps of the methodology: data modelling and data publication. In the first part in Section 4.2 we explain how we performed the Pordata data modelling step. We start by eliciting the Pordata entities in Section 4.2.1. Section 4.2.2 introduces the URI and HTTP technologies that are used on the Web of Data to give names to concepts. We show how we adopted these technologies to define names of the Pordata concepts and embodied them in the Pordata URI scheme. In Section 4.2.3 we introduce RDF, the data model to describe entities of the real world. We evaluate the existing RDF vocabularies to model statistics and temporal entities and present our modelling solution, the Pordata Vocabulary. The second part of this chapter, Section 4.3 reports on the technical realization of the LD principles for publishing Pordata as Linked Data. We introduce the selected publishing pattern that is centered around the Virtuoso Universal Server. We also discuss the preliminary work we did to simulate the Pordata database. The publishing pattern consists of two steps: RDF generation and Linked Data deployment. In Section 4.3.2 we discuss how we generated the RDF representation of Pordata. In Section 4.3.3 we consider the deployment of Linked Pordata. We introduce SPARQL, the language to query RDF, and discuss the Virtuoso URL-rewriter mechanism which can be used to deploy the Linked Pordata.

4.1 Methodology

The following main ideas stated in [22] underlie our methodology for publishing Pordata as Linked Data:

- Publishing Linked Data requires adoption of the Linked Data principles.
- Compliance with the Linked Data principles does not entail abandonment of existing data management systems and business applications but simply the addition of extra technical layer of glue to connect these into the Web of Data.

Our methodology for publishing Pordata as Linked Data consists of the following steps:

1. *Modelling Pordata*
    - *Identifying the Pordata concepts*
    - *Designing the Pordata URI scheme*
    - *Describing Pordata*
        - *Sourcing the existing vocabularies*
        - *Defining the Pordata Vocabulary*
2. *Publishing Pordata*
    - *Selecting a Publishing Pattern*
    - *Generating RDF*
    - *Deploying Linked Pordata*

We divided the methodology into two basic steps: modelling Pordata and publishing Pordata. The Business Process Diagram presented in Figure 4.1 illustrates top level activities of the methodology.

During the Pordata modelling step we make the primary design considerations that must be taken into account when preparing Pordata to be published as Linked Data on the Web. These considerations break down into three areas, each of which maps onto one or two of the Linked Data principles (we introduce the principles in Section 3.2):

- naming Pordata concepts with HTTP URIs;
- describing the concepts with RDF.

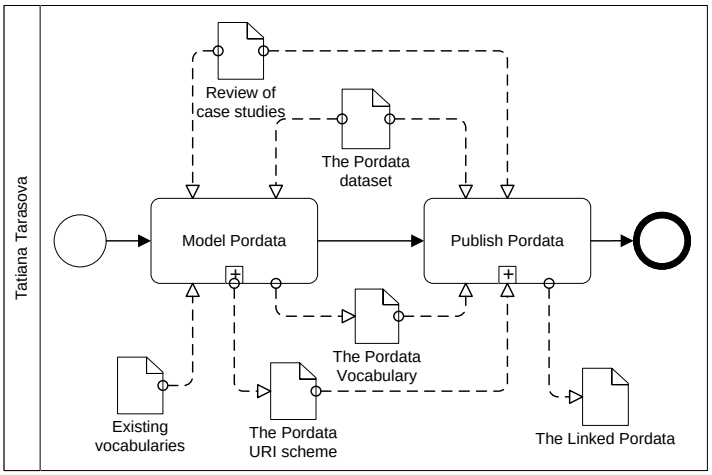
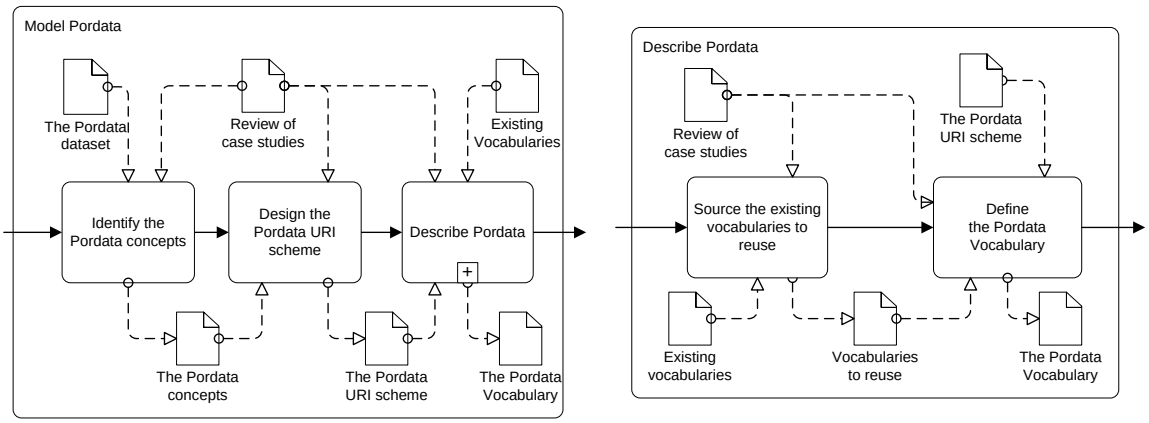


Figure 4.1: BPD Top level: *Publishing Pordata as Linked Data* process.



(a) BPD Top level: *Data Modelling* (b) BPD *Data Modelling*: *Describe Pordata*

Figure 4.2: BPD *Publishing Pordata as Linked Data*: *Data Modelling*.

Figure 4.2 depicts the data modelling activity in detail.

The Pordata publication step concerns the technical realization of the LD principles. We based the selected publishing pattern on the existing Pordata management infrastructure. Namely, in the pattern we specified how to generate the RDF representation of Pordata from the relational database and how to deploy the Linked Pordata. Figure 4.3 provides a detailed illustration of the data publication activity.

In the subsequent sections we describe how we developed a proposal for publishing Pordata as Linked Data, following the methodology presented above. Section 4.2 explains the data modelling step. Section 4.3 contains description of the Pordata data publication step. While developing the proposal we considered best practices described in different case studies. Among others, we considered works done for publishing *DBpedia* [2], *Environment Specimen Bank* [5, 101], *LinkedGeoData* [106],

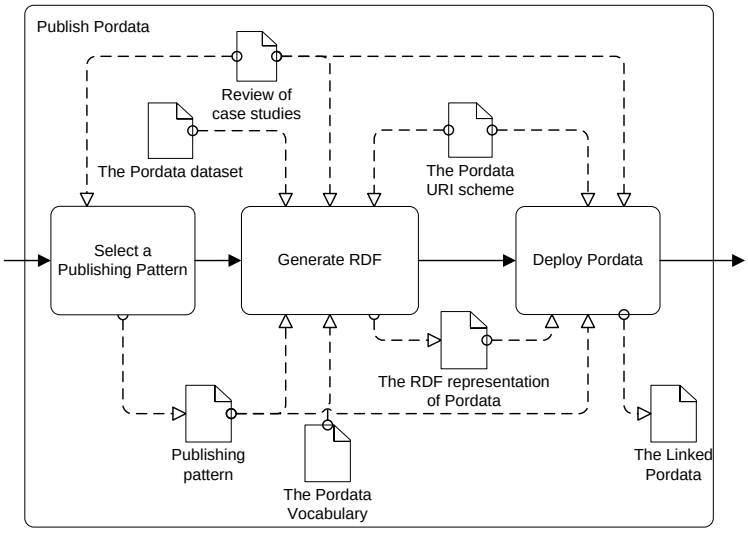


Figure 4.3: BPD *Publishing Pordata as Linked Data*: *Data Publication*.

Linking UK Government Data [65], *Spanish GeoLinked Data* [40], *RDFising and Interlinking the Eurostat Data Set Effort* [53], *the Nomenclature of Territorial Units for Statistics* [29], *The Linked Data Service of the German National Library* [123], *Linking Open Drug Data* [64] and *Revyu.com* [58].

4.2 Modelling Pordata

The initial step in the data modelling process is to single out *things* (real-world objects and abstract concepts) in Pordata that carry the semantics of the data. Having these things extracted from Pordata, we can express and describe them in a machine-readable way as the Linked Data principles suggest.

4.2.1 Identifying the Pordata concepts

In order to identify things in Pordata we describe the Pordata data characteristics based on the analysis of the Pordata dataset performed in Chapter 2. The Pordata table presented in Figure 4.4 is used to exemplify the characteristics.

Pordata characteristics

- *Multiple dimensions*.

  Statistics in Pordata are presented in tables. We may think of a Pordata table as a multidimensional space filled in with statistical observations. We identified two types of dimensions in Pordata: temporal and series. Temporal dimensions are

Session/Screening

Years	Country of Origin							
	Total	Portugal	Spain	France	United Kingdom	USA	Other countries	Co-productions
1979	299.120	39.792	6.719	37.795	30.002	102.089	82.723	x
+ 1980	299.760	41.283	5.198	36.632	28.336	112.707	75.604	x
+ 1990	176.678	3.199	1.957	4.220	9.640	145.208	12.454	x
2000	420.033	9.812	x	6.577	14.474	370.720	15.818	2.632
2001	450.349	5.005	x	12.098	4.805	404.714	20.606	3.121
2002	504.768	7.460	x	17.598	6.345	434.606	32.787	5.972
2003	569.932	10.310	x	10.514	15.001	476.374	51.711	6.022
2004	⊥ 551.850	⊥ 10.882	⊥ 6.764	⊥ 5.649	⊥ 4.364	⊥ 376.765	⊥ 7.389	⊥ 140.037
2005	589.110	11.994	3.098	9.734	23.851	338.767	6.341	195.325
2006	591.139	15.727	7.597	7.241	15.195	360.246	11.422	173.711
2007	605.717	13.147	2.210	8.498	19.320	413.809	6.175	142.558
2008	644.778	6.921	4.127	11.242	4.477	333.732	3.767	280.512
2009	651.325	21.112	4.842	17.197	10.318	360.587	9.345	227.924
2010	670.315	17.749	3.171	12.443	4.979	464.140	10.361	157.472

Data Source: INE (until 2003); ICA/MC (from 2004), PORDATA

Last updated:2011-12-27

Figure 4.4: The screen shot of the *Culture and Sports/Cinema/Screenings: total and by film's country of origin* Pordata table taken from `http://pordata.pt/`.

years. Series is just a generic term to indicate what statistics are applied to. Each Pordata table has one or more different statistical series. Thus, the example table has the temporal dimension *Year* and one series dimension *Country of origin*.

- *Observations*.

  Each observation is associated with a set of dimension values that uniquely identifies this fact in the multidimensional space. The temporal dimension has years as values starting from 1960 till the present time. Series dimensions have one or more instances of the series. For example, the table above has 8 instances of *Country of Origin*: *Total*, *Portugal*, *Spain*, *France*, *United Kingdom*, *USA*, *Other countries* and *Co-productions*.

  The observation that "the number of screenings of movies produced in Portugal in year 1979 was 39.792" can be located in the example table as follows: *Year* = "1979" and *Country of Origin* = "Portugal".

- *General and Specific Metadata*.

  Statistics always go along with metadata. Metadata is needed to correctly interpret the data. Pordata metadata can be divided into general and specific.

General metadata provides information common for all observations of a statistical dataset. This includes the *measured phenomenon* (i.e., what was measured) and the *unit of measure* (i.e., how it was measured). Statistics from our example table, contain data about "Screenings" (i.e., measured phenomenon) which was measured in "Numbers" (unit of measure) .

Specific metadata provides information relevant to a particular observation. This refers to the *type of the value*. In the example table, the observation for *Year* = "1979" and *Country of Origin* = "Co-productions" is of type *not available* (that is the meaning of "x"). The value types are common for all Pordata tables and defined in the Table Symbology (Section 2.1.1).

- *Structural vs Domain semantics*.

  We can identify two kinds of semantics when we consider Pordata statistical tables: domain and structural.

  Structural semantics stem from *how statistics are presented*, i.e., what dimensions and metadata we need to define the statistics. For instance, to be able to encode observations of the example table we need two dimensions (one temporal and the statistical series "Country of origin"), the unit of measure, the measured phenomenon and different types of values.

  Domain semantics are *what statistics are about*. The table above, for example, is about screening of movies from different countries. The domain semantics in Pordata is encapsulated as instance of structural concepts, i.e., instances of dimensions and metadata. For example, the table above contains statistics about screenings of movies (as the measured phenomenon suggests) from different countries (according to the instances of series) over years (i.e., instances of the temporal dimension).

Pordata concepts Based on the discussion above, we derived two types of concepts: concepts that capture structural semantics (*structural concepts*) and concepts that capture domain semantics of Pordata tables (*data concepts*).

Structural concepts are generic for all Pordata tables. There are two fundamental structural concepts: *Table* and *Observation*. The former embodies a collection of statistical observations. The latter refers to an observation of a particular Pordata table (i.e., a single cell in a Pordata table). There are two dimension concepts that are used to indicate what the statistics apply to: *Series dimension* and *Temporal dimension*. Finally, we have structural concepts to provide metadata for statistics. They include *Unit of measure*, *Measured phenomenon* and *Type of value*.

The data concepts are specific to each Pordata table and instantiate generic structural concepts. Basically, in each Pordata table we can derive one instance of *Table* (i.e., the table name), several instances of *Observation* (i.e., cells in the table), at least on instance of *Series dimension* (i.e., series instances), several instances of *Temporal dimension* (i.e., years) and instances of the metadata structural concepts.

The following list summarizes the discussion about the Pordata concepts:

- Structural concepts:
  - *Table*
  - *Observation*
  - *Series dimension*
  - *Temporal dimension*
  - *Metadata*:
    - *Unit of measure*
    - *Measure phenomenon*
    - *Type of value*
- Data concepts, i.e., instances of the structural concepts:
  - table names;
  - observations;
  - instances of series;
  - years;
  - metadata instances.

4.2.2 Designing the Pordata URI scheme

In the previous sections we identified concepts in the Pordata dataset that capture semantics of Pordata. The next step is to give names to these concepts. The first and second Linked Data principle say to use HTTP URIs to name things [15].

URIs

A Uniform Resource Identifier (URI) [110] is a compact sequence of characters. The generic URI syntax consists of a hierarchical sequence of components referred to as the scheme, authority, path, query, and fragment:

scheme "://" **authority** "/"**path** ["?"**query**] ["#"**fragment**]

For example, we can single out the following components in `http://example.com/Portugal?theme=cinema#screenings`:

scheme: `http`
authority: `example.com`
path: `Portugal`
query: `theme=cinema`
fragment: `screenings`

URIs are a standard way to identify resources on the Web. A Web resource can be a file, an image or a digital document that is accessible on the Web. They are called *information* resources ([63], Section 2.2.). Linked Data extends the notion of the Web resource from information resources to encompass human beings, abstract concepts, physical products, places, in other words, anything in the world. In contrast to information resources, these are called *non-information* resources [95]. Non-information resources are also identified by URIs.

HTTP URIs

There are many URI schemes available in the Web of documents, e.g., `http`, `ftp`, `urn`, `mailto` and others. However, according to the second Linked Data principle the `http` scheme should be used to identify real-world concepts. HTTP URIs were chosen to identify non-information resources because:

- They provide a simple way to create globally unique names with decentralized management that allows every owner of a domain name to create new global identifiers within this domain name.

  The possibility to define custom names for things of interest makes it easier for data publishers to enter the Web of Data. Since they do not need to search for the terms that other people or organizations define and analyse them. In fact, having a centralised management of the names is barely possible since it means that everybody must agree on using the same names to refer to the same concepts.

- They work not just as names for resources but also as means of accessing them on the Web and retrieving their descriptions using the Hypertext Transfer Protocol (HTTP) [48].

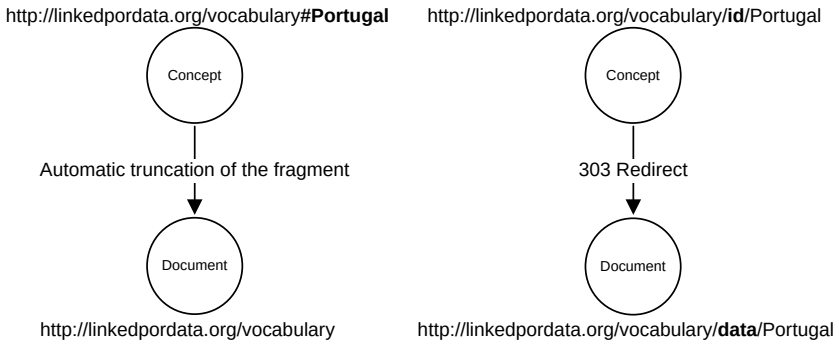
  HTTP defines a simple yet universal mechanism for retrieving resources, the *dereferencing mechanism* [95]. The mechanism provides means for any client that

speaks the HTTP protocol (e.g., a Web browser), to look up resources identified by HTTP URIs and retrieve their descriptions, i.e., Web pages. Using dereferenceable HTTP URIs for identifying real-world entities allows to serve their descriptions on the Web using the HTTP protocol.

Information vs Non-Information resources

On the Web of Data URIs are assigned to both non-information resources (real-world things) and information resources (i.e., documents describing these things). This is useful, since we can easily describe documents in terms of RDF. However, this can also cause mixing up real-world concepts and documents describing them. We can not use the same URI to name both, since the concept and the document describing it are two different things and might require two different assertions to be made about them. We can demonstrate it on the following example. Assume we have the concept of Portugal and a Web page of a tourist agency that describes Portugal. If we use a single URI to name Portugal and the page there is no way to state that, for example, *Alice likes Portugal, but she does not like the look of the tourist agency's page*. Having one URI for Portugal and the page will not allow us to distinguish the objects of this statement. So, two URIs are needed: one for the concept of Portugal (i.e., non-information resource) and one for the Web page describing Portugal (i.e., information resource). Thus, in the Web of Data it is important to have a mechanism that, given a URI, can determine what kind of a resource, information or non-information, this URI identifies. Two solutions exist to address this problem: *hash URIs* and *303 URIs*.

Hash URIs The Hash URIs strategy makes use of a special part of a URI, the *fragment*, which is separated from the base part of the URI by a hash symbol "#". When a Web client looks up a hash URI, the HTTP protocol requires the fragment part to be stripped off before requesting this URI from a Web server. This means that hash URIs can not be directly retrieved from the server and, therefore, do not necessarily identify documents (information resources) in the Web. Thus, hash URIs can be used in Linked Data to identify real-world objects (non-information resources) without creating ambiguity. Hash URIs are used where it is essential not to issue many requests to a Web server to retrieve a single resource's description. Assume we have a vocabulary with a dependency between terms, i.e., when definitions of terms require definitions of other terms. In order to obtain the complete definition of a dependent term, we need to retrieve several resources. This involves several communications with the Web server. What we can do instead is to retrieve the entire vocabulary once.



(a) Hash URIs (b) 303 URIs

Figure 4.5: Unambiguous interpretation of non-information resources.

303 URIs The 303 URIs strategy uses a special HTTP status code: *303 See Other* [47]. If a Web server responds with such a code to an HTTP URI request, it means that the requested resource is not a regular Web document. Indeed, a non-information resource is a real-world object and can not be transferred over the network! However, it is useful to provide information about such resources. Since *303* is a redirect status code, the server can give the location of a document that represents the resource. Then, the client dereferences this new URI and gets an RDF document describing the real world object. If the initial request is answered with *200 OK*, then the client knows that the URI identifies a Web document, i.e., an information resource. Thus, the ambiguity between the real-world object and its description is avoided.

Figure 4.5 demonstrates the hash URI and 303 Redirect approaches.

The hash URIs approach is not suitable when a document representing a real-world thing is large, because this approach may lead to unnecessary transmission of the descriptions of all resources that share the same non-fragment URI part, irrespectively of whether the client is interested in only one resource or all of them. To deal with such cases, the *303 URIs* strategy can be used. The 303 URIs approach is very flexible and the redirection target can be configured separately for each resource. There could be one single document representing each small resource, one large document for all of them or any combination in between. It is also possible to change the policy later on. With such flexibility, the 303 URIs approach is suitable to describe resources that are parts of very large datasets. Both 303 and Hash URIs can be combined, allowing a large dataset to be separated into multiple parts and have an identifier for a non-information resource.

The Pordata URI scheme

Similarly to URIs for Web pages in a conventional Web site, URIs for real-world concepts in a Linked Data set should also be designed carefully. The second Linked Data principle is that URIs should be created using the `http` URI scheme [15]. In practical terms, it amounts to developing a path structure of an HTTP namespace that a data publisher owns. Assume we own the domain name `linkedpordata.org` and run a server at `http://linkedpordata.org`. Next, we will design the path structure to assign names of the Pordata things within the `http://linkedpordata.org` namespace and illustrate it by example of "Screening by country of film" table.

All the good practices that exist for designing HTTP URIs of traditional Web pages should be applied to designing a URI scheme of a Linked Data set [14]. In short, this includes:

- Simplicity: short, mnemonic URIs are easier to remember and type. In principle, users should not need to care about URIs which are left to machines to deal with. In practice, it is often the case when humans try to guess URIs of companies and brands or communicate them in messages and emails [113]. So, to make URIs of real-world concepts easier to remember, they should be short.

- Stability: URIs should be persistent, as other people might refer to them and expect them to be stable. On the Web of Data it is also important to have stable URIs, since other people may use them to link to your dataset. Links are crucial for performing data integration on the Web of Data. We will discuss them in Chapter 5.

- Manageability: URIs should be designed in such a way that they can easily be changed without breaking the whole URI structure. This also applies to the URIs of real-world concepts.

In addition, URIs of real-world entities on the Web of Data should fulfil two more requirements, as we discussed in Section 4.2.2. First, they should be dereferenceable, i.e., their descriptions should be retrievable on the Web via HTTP. Second, they should not be confused with the documents representing them. To fulfil these requirements, Linked Data adopts two approaches to name real-world entities: hash URIs and 303 URIs. We adopted both hash and 303 URIs approaches for different kinds of entities. In this section we will explain the choice of each of them to identify Pordata concepts. In Section 4.3.3 we discuss the issue of making dereferenceable the URIs presented here.

For the sake of brevity, we will omit writing the namespace `http://linkedpordata.org` each time we introduce a new path pattern. We will only write the part that starts right after the namespace.

1. *Structural concepts*.

   Structural concepts will be named using the hash URIs strategy, since it might be useful to retrieve the descriptions of all the available structural concepts at once. Also the document describing all of them will be relatively small, as we have few structural concepts: series and temporal dimensions, unit of measure, measured phenomenon and types of values.

   We will use the `/schema#{concept-name}` path pattern to identify structural concepts. For example, the unit of measure concept is identified by:

   `http://linkedpordata.org/schema#UnitMeasure`

2. *Data concepts: table names*.

   We can adopt the hash URI strategy to name tables by using the following path pattern: `/dataset#{table-name}`. When a hash URI is looked up on the Web, the fragment part is trucated before the URI is requested from a Web server. Then the rest of the URI, `http://linkedpordata.org/dataset`, is used to serve descriptions of all Pordata tables. However, there are many tables in Pordata, and each of them is a self-contained thing. There is no sense to put together descriptions of all the tables. With the 303 URI strategy we can provide a separate description for each table by redirecting to the corresponding URI.

   Thus, we will adopt the 303 strategy to name Pordata tables and use the following path pattern for this: `/dataset/{table-name}`. The URI identifying the example table looks as follows[1]:

   `http://linkedpordata.org/dataset/Screening_by_country_of_film`

3. *Data concepts: observations*.

   Each observation belongs to a concrete table. We will name them within the dataset path. We will adopt the 303 URIs approach to identify observations, since it might be useful to retrieve the description of a single observation. Descriptions of all the observations of a particular table are served by a document identifying this table.

   We will use the `/dataset/{table-name}/{obs-id}` path pattern to identify observations. An observation of the example table will be identified by:

   `http://linkedpordata.org/dataset/Screening_by_country_of_film/obs-1`

[1]Note spaces in the names of concepts are changed to underscores to form valid URIs.

4. *Data concepts: years*.

   We can apply equally both strategies to name *Years*. We chose the hash URIs strategy.

   URIs of years will be created using the path pattern: `/time#Year{year}`. For example, year 1979 is identified by:

   `http://linkedpordata.org/time#Year1979`

5. *Data concepts: instances of series and metadata instances*.

   We will maintain the Pordata vocabulary for describing instances of series and metadata instances such as "Portugal", the series instance, "Screening", the instance of a unit of measure, etc. We could adopt the hash URIs strategy to name these concepts. However, the Pordata vocabulary does not have dependencies between definitions, and it is unlikely that somebody will be interested in retrieving descriptions of all the domain specific concepts at once.

   Thus, we adopt the 303 URI strategy to name these concepts. Each concept will be named using the following path pattern: `/vocabulary/{concept-name}`. For example, "Portugal" will be identified by:

   `http://linkedpordata.org/vocabulary/Portugal`

4.2.3 Describing Pordata

Linked Data proposes to use the Resource Description Framework (RDF) [66] to describe datasets.

RDF data model

RDF is a general model for conceptual description of information on Web resources. A Web resource can be a file or a digital document that is accessible on the Web. They are called *information* resources ([63], Section 2.2.). For example, in RDF one can represent metadata about information resources such as titles, authors, modification dates, copyright and licensing information. Linked Data extends the notion of the Web resource from information resources to encompass human beings, abstract concepts, physical products, places, in other words, anything in the world. In contrast to information resources, these are called *non-information* resources [95]. RDF can be used to represent names of people or cities, the fact that one person knows another, etc. The RDF data model is described in detail as part of the W3C RDF Primer [77]. Next, we discuss the basic ideas of the RDF data model.

In RDF, the description of a resource is represented as a number of triples. Each RDF triple intrinsically reflects the basic structure of a simple sentence:

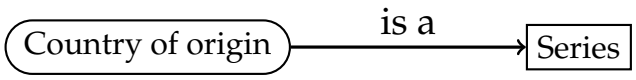
<subject predicate object>

For example, to describe the example table, we may come up with the following sentence "Country of origin is a series". In RDF this sentence looks as follows:

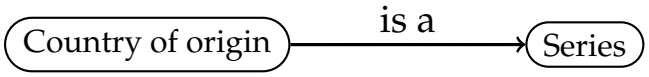
subject: Country of origin
predicate: is a
object: Series

The RDF data model represents triples as a labeled directed graph, where subjects and objects are represented as nodes and predicates as edges connecting subjects to objects. The subject and predicate of an RDF triple are resources, whereas, the object can either be a resource or a literal. Literals are just raw text that is used to define numbers, dates or relate the subject of an RDF triple, e.g., a city, to a human-readable representation of the city's name. Resources in RDF graphs are represented as ovals, and literals as rectangles. The example above may have two alternatives:

1. *Series* is a literal. The RDF graph below corresponds to the example sentence:

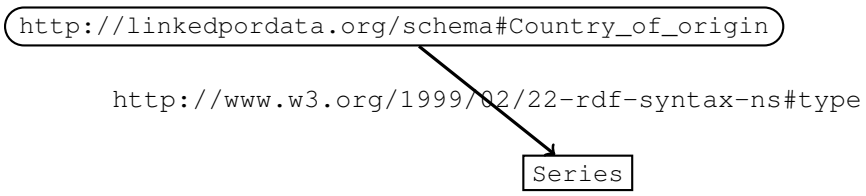


2. *Series* is a resource. The RDF graph below corresponds to the example sentence:

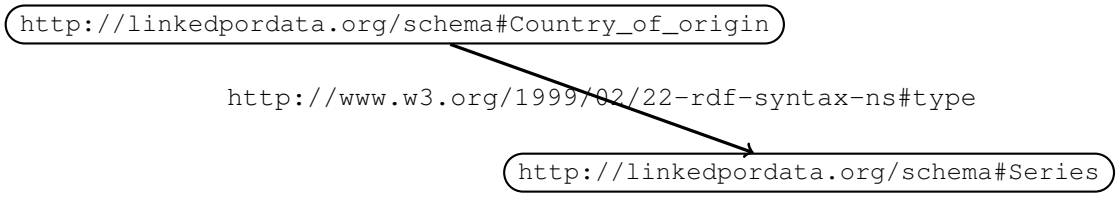


Let us assume that "Country of origin" is identified by `http://linkedpordata.org/schema#Country_of_origin`. To represent "is a" relation there is a predefined property in RDF `http://www.w3.org/1999/02/22-rdf-syntax-ns#type`. Then there are two options:

1. "Series" is a literal. The RDF graph below corresponds to the example sentence.



2. "Series" is a resource identified by, for example, `http://linkedpordata.org/schema#Series`. The following RDF graph represents the example sentence:



RDF can be used to provide descriptions of the Pordata concepts in the form of simple sentences that use these concepts as subjects or predicates and relate them to each other, e.g., "Country of origin is a series". In general, a specialised set of predefined concepts and relationships between these concepts with their own special meanings is called a *vocabulary*. In the following we will explain how we defined the Pordata Vocabulary.

Sourcing the existing vocabularies

The best practices for publishing Linked Data recommend to reuse the terms of the existing vocabularies to define custom vocabularies [22]. This will facilitate further reuse and uptake of the dataset on the Web of Data. First of all, many Linked Data consuming software applications are most likely "tuned" to understand the widely deployed vocabularies. Second, reusing terms of the existing vocabularies increases interoperability of dataset with others that reused the same vocabularies.

There are already plenty of vocabularies developed by different communities and organizations for different purposes and distributed on the Web. A list of the most commonly used vocabularies is maintained by W3C [122]. There are also several sites collecting vocabularies, such as *Linked Open Vocabularies (LOV)* [118] and *Swoogle* [109]. Our examination of the available resources that collect the existing vocabularies revealed two vocabularies to represent statistics: the *Statistical Core Vocabulary (SCOVO)* [57] and the *Data Cube vocabulary* [35]. In the next sections we study both vocabularies and exemplify them with the table presented in Figure 4.4.

SCOVO

The Statistical Core Vocabulary (SCOVO) [57] is a light-weight RDF vocabulary for expressing statistical data. It defines three classes to describe statistics:

- `Dataset`: corresponds to a statistical table (i.e., a Pordata table);

- `Item`: corresponds to a statistical observation (i.e., a cell in a Pordata table);

- `Dimension`: corresponds to a dimension (i.e., *Time* or *Series* in Pordata tables).

Figure 4.6 depicts the RDF model of the SCOVO vocabulary.

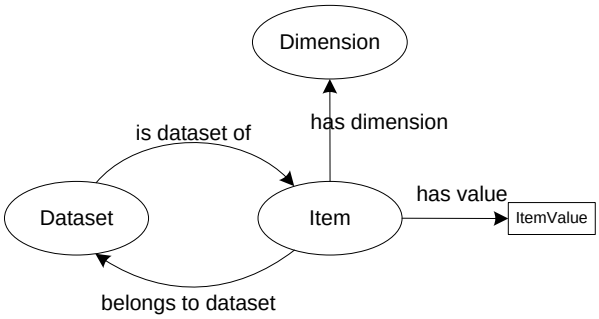


Figure 4.6: The RDF SCOVO model.

The observation that "the number of screening of Portuguese movies in 1979 in Portugal was 39.792" in SCOVO is depicted by Figure 4.7. `ScreeningsByCountryOfOrigin` is a dataset of the observation, and vice versa, the observation belongs to the dataset `ScreeningsByCountryOfOrigin`. `Observation` is attached two dimensions: `Portugal` and `Year1979`, and has value `39.792`. Note that the metadata (highlighted in red on the figure), the unit of measure `Number` and the measured phenomenon `Screenings`, is attached as dimensions via the `has dimension` property.

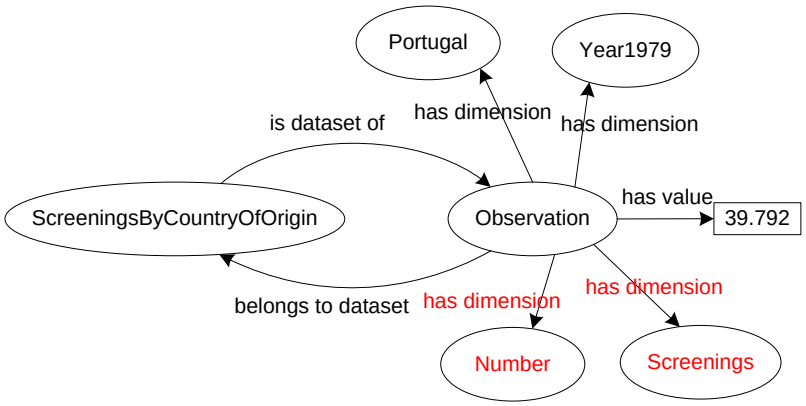


Figure 4.7: SCOVO encoding of the fact that "the number of screening of Portuguese movies in 1979 in Portugal was 39.792".

The Data Cube vocabulary

The RDF Data Cube model is depicted by Figure 4.8[2].

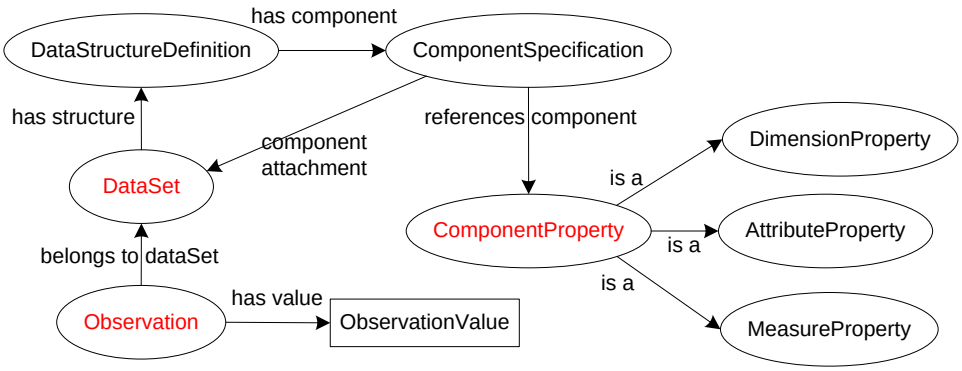


Figure 4.8: The RDF Data Cube model.

The RDF Data Cube vocabulary [35] defines similar classes as SCOVO to represent statistical datasets, observations and dimensions (highlighted in red on the Data Cube model):

- `DataSet` corresponds to the SCOVO `Dataset`;
- `Observation` corresponds to the SCOVO `Item`;
- `ComponentProperty` corresponds to the SCOVO `Observation`;

Additionally, the Data Cube vocabulary specifies `ComponentProperty` to be `DimensionProperty`, `AttributeProperty` or `MeasureProperty`. In general, they are called *components* of a statistical table:

- `DimensionProperty` defines what an observation applies to;
- `AttributeProperty` defines attributes of observations;
- `MeasureProperty` defines the phenomenon being observed.

The example table has:

- two dimension components: the time dimension `refPeriod` (i.e., time period) and the series dimension `refCountry_of_Origin`;

[2]Note the figure does not represent all the Data Cube classes. It aims at illustrating basic classes that give a general idea of how statistics are modeled in Data Cube.

- two attribute components: the unit of measure `hasUnitMeasure` and the type of the value `hasValueType`;

- one measure component: the measured phenomenon `hasMeasure`.

See how these components are encoded in Data Cube on Figure 4.9.

Components of a dataset are combined into specifications, `ComponentSpecification`. A specification in the simplest case just references the corresponding component. By default a component is attached to an observation. Alternatively, the level of the component's attachment can be set to the dataset.

Component specifications are then combined into a data structure definition. `DataStructureDefinition` defines a structure of a dataset and can be reused across datasets with the same structure. We can define the corresponding component specifications for the example table as follows:

- `CSrefPeriod` just references the time dimension component `refPeriod`;

- `CSrefCountryOfOrigin` just references the country of origin dimension `refCountry_of_Origin`;

- `CShasMeasure` references the measured phenomenon component `hasMeasure` and is attached to the dataset;

- `CShasUnitMeasure` references the unit of measure attribute component `hasUnitMeasure` and is attached to the dataset;

- `CShasValueType` just references the type of the value attribute component `hasValueType`.

The structure `DSD` of the example table is shown on Figure 4.9.

It should be noted that all the components, except `refCountry_of_Origin`, are generic in the context of Pordata. This means that in order to encode observations of an arbitrary Pordata table we always need to specify a time dimension component, provide attribute components unit of measure and the types of values[3] and the measured phenomenon component. The series dimension component is the only one that varies from table to table.

Thus, on the one hand, we can reuse this structure for other Pordata tables that have the same series dimension "Country of Origin". On the other hand, we can reuse each component specifications to define new structures. Specifically, all the component specifications but `CSrefCountry_of_Origin` can be reused to encode the structure

[3]Types of values of observations are the same for all the tables and declared in the Table's Symbology.

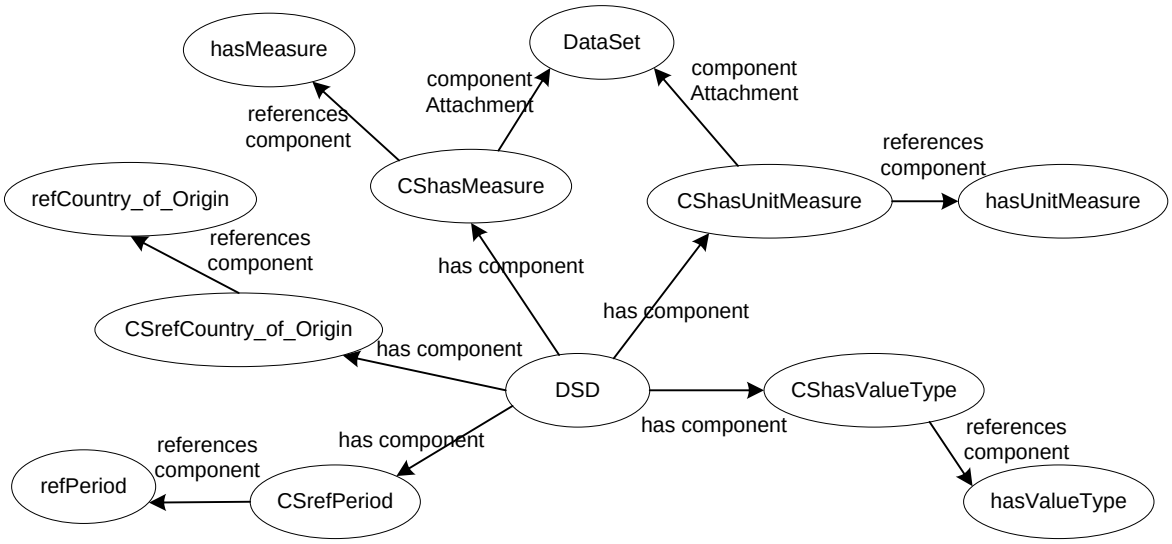


Figure 4.9: The data structure definition of "Screenings by country of origin" in Data Cube.

of any other Pordata table. The components themselves can be reused to define new specifications if required. For example, for some purposes we may wish to have another specification for the unit of measure component that leaves the attachment level of the component to observation.

Finally, the observation that "the number of screening of Portuguese movies in 1979 in Portugal was 39.792" in Data Cube is shown by Figure 4.10.

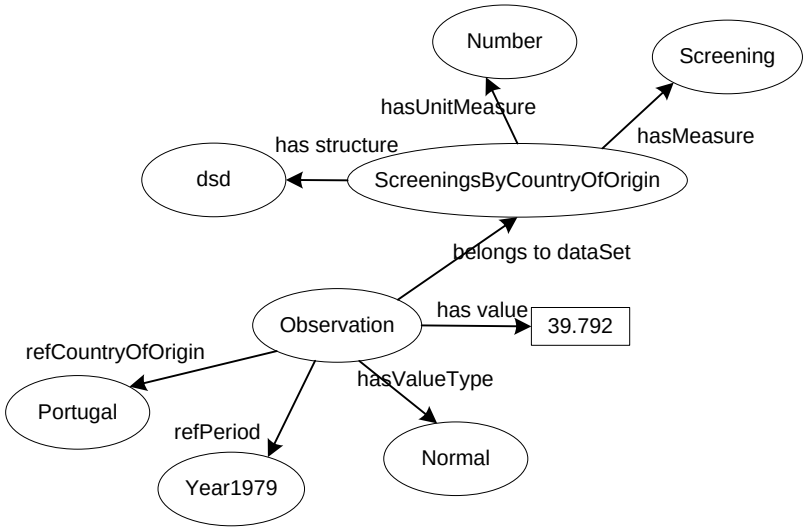


Figure 4.10: Data Cube encoding of the observation that "the number of screening of Portuguese movies in 1979 in Portugal was 39.792".

Note how we attached the corresponding values of dimensions and metadata to the observation by using the components we defined before.

SCOVO vs Data Cube

The analysis of SCOVO and Data Cube revealed that both vocabularies are expressive enough to model arbitrary statistics in terms of three basic classes that represent statistical tables, observations and dimensions. Both vocabularies allow to incorporate the domain semantics by encoding dimensions and metadata. This allows to establish more connections with other datasets that cover similar topics. Thus, the series dimensions of the example table (countries) can be associated with other datasets that define the same countries, e.g., Geonames [50].

SCOVO and Data Cube provide a high level of granularity, i.e., we can express a single statistical observation separately from the statistical dataset by attaching corresponding dimensions and metadata to it. Such flexibility allows to address statistical observations in small pieces and reuse them somewhere else. For example, a journalist writing an article about year 1979 can address a single statistical observation about screenings of Portuguese movies in 1979 rather than referencing the whole statistical table.

In comparison with Data Cube, SCOVO has several limitations:

- SCOVO does not distinguish between dimensions and metadata. Both are modelled as `Dimension`. Data Cube has three different concepts, `DimensionProperty`, `AttributeProperty` and `MeasureProperty` to address dimensions and different kins of metadata.

- As a consequence of the previous limitation, SCOVO does not differentiate between general and specific metadata. Data Cube allows to attach general metadata to a dataset and specific to observations by using the property `componentAttachment`.

- SCOVO does not allow to encode the structural semantics of a dataset separately from the data. Data Cube provides means to encode structural semantics separately by means of `ComponentSpecification` and `DataStructureDefinition`. Both concepts can be reused across different datasets that share the same structure.

We will adopt the Data Cube vocabulary to encode the Pordata statistics. In fact, the Data Cube vocabulary was developed to address the limitations of SCOVO. As a result, the Data Cube vocabulary enables more concise encoding, since we can encode

general metadata once for the whole Pordata table, without attaching it to each observation. This fact is essential, since the Pordata dataset consists of many statistical tables. Moreover, the distinction between dimensions and different kinds of metadata in Data Cube allows to produce semantically richer encoding. We not only attach different dimension values to observations but state, for example, what is the unit of measure and what is the type of value. Then, for instance, we can extract datasets that have the same unit of measure (not necessarily Pordata datasets). As we discussed above in the analysis of the Pordata data, there are structural and data concepts in Pordata. The former are generic for all the Pordata tables, the latter are specific to each table. Thus, it is good to have the possibility to define structural concepts once and reuse them across Pordata datasets to encode data concepts. It will help to reduce the redundancy in encoding and facilitate interoperability between datasets that share the same structure.

The temporal dimension in RDF

Representing time in RDF is a well studied problem [30,46,67]. Our examination of the existing vocabularies to model time dimension in Linked Data revealed two commonly used solutions: Time Ontology [59] and the Timeline Ontology [125].

The *Time Ontology* defines two basic types of temporal entities:

- *time instants* with different levels of granularity, e.g., *22d February 2012: 17:41:00 GMT*, a particular time point up to seconds, or *22d February 2012*, a particular day of the year;

- *time intervals*, e.g., *year 2012* to refer to a time period from 1st of January to 31st of December;

The Time Ontology was developed to enable complex reasoning over time. It provides means to model different relationships between time entities. For example, we can express that one time instant happened before or after another one; two time instants are equivalent or two time intervals overlap.

The *Timeline Ontology* defines the same basic temporal entities, *Instant* and *Interval*. In addition, it defines the notion of *Timeline* as a way to refer to a coherent temporal backbone with a starting and ending date. Thus, we can associate different time instants and intervals on a single timeline and work with them simultaneously as with a single temporal logical space.

In Pordata we have temporal concepts of only one type, time intervals. They are always represented by years, we do not need to detail them further into months or days. Also, there are no complex relationships between the Pordata temporal concepts.

We can only state that one year happened before or after another. Thus, we do not need such a sophisticated modelling framework as the Time Ontology. Moreover, we found the concept of *Timeline* might be useful for different tasks, e.g., integrating Pordata statistics that use years as a temporal dimension with other data that is characterised by different kind of temporal entities, time instants instead of intervals. We can define one timeline and associate different temporal entities with it, e.g., the Pordata statistical years and the dates of presidential elections from the DBpedia dataset. Thus, using the timeline as a common platform, we can bring together and analyse dependencies between Pordata statistics and information about presidential elections.

Figure 4.11 demonstrate how we used the Timeline Ontology to encode the temporal dimension by example of year 1979.

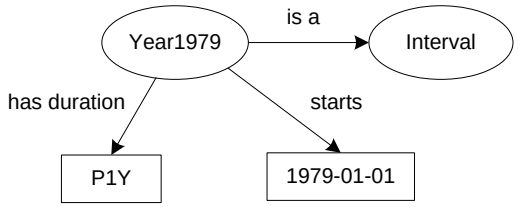


Figure 4.11: Year 1979 in terms of the Timeline Ontology.

Note that the duration of the year we expressed in terms of the XML Schema duration that uses pattern `PnYnMnDTnHnMnS` [124]. According to this pattern, `P` is a literal value to start the expression and `nY` is the number of years.

Defining the Pordata Vocabulary

In the previous sections we analysed the existing vocabularies that can be used to define the Pordata Vocabulary. We justified the choice of Data Cube to describe statistics and Timeline Ontology to encode years. In this section we explain how we define the Pordata Vocabulary by reusing terms from Data Cube and Timeline.

Vocabularies define specific kinds of concepts, i.e., categories of concepts. For example, Data Cube defines a generic concept of `DataSet` to refer to statistical tables, or the Timeline Ontology defines a generic concept of `Interval` to refer to a time period. Reusing these concepts means that we need to declare memberships of the Pordata entities in the categories of Data Cube. In other words, each Pordata concept should be an instance of some class of the vocabulary. For example, "Screenings by country of origin" is an instance of `DataSet`, "Country of origin" is an instance of `ComponentProperty`, and so on. This can not be done in RDF, as RDF does not

provide such mechanisms. This function is dedicated to the vocabulary description language RDFS.

RDFS

The RDF Vocabulary Description Language (also known as **RDF Schema**) [23] was developed to define vocabularies in RDF. RDFS provides a mechanism to declare specific categories of concepts (referred to as *classes*) and specific relationships with other concepts (referred to as *properties*). The RDF Schema facilities are themselves provided in the form of the RDF vocabulary.

For historic reasons, the terms of the RDFS language are defined in two vocabularies `http://www.w3.org/1999/02/22-rdf-syntax-ns#` and `http://www.w3.org/2000/01/rdf-schema#`. *Prefixes* are used to introduce compact notations of terms. For example, `<prefix>:<term>` identifies the *term* within the namespace specified by `<prefix>`. The prefix `rdf` is a conventional notation of `http://www.w3.org/1999/02/22-rdf-syntax-ns#`. Writing `rdf:type` corresponds to the full URI of the concept *type* `http://www.w3.org/1999/02/22-rdf-syntax-ns#type`. `rdfs` is a conventional notation of `http://www.w3.org/2000/01/rdf-schema#`. We will use these and other conventional prefixes to refer to terms from the commonly used vocabularies. Consult Appendix A for all the prefixes used in this thesis. Additionally, we will use the following prefixes to identify concepts of the Pordata Vocabulary:

prefix	**namespace URI**
porschema	`http://linkedpordata.org/schema#`
porvocab	`http://linkedpordata.org/vocabulary/id/`
pordata	`http://linkedpordata.org/pordata/id/`
portime	`http://linkedpordata.org/time#`

Next, we will demonstrate the basic RDFS classes and properties that we will need for our work[4]. The complete set of the RDFS terms is accessible from [97] and [98].

RDFS core classes:

- `rdfs:Resource` all the resources being described by RDF are considered to be instances of this class;
- `rdfs:Literal` is used to represent literal values such as strings and integers;
- `rdf:Property` is used to represent the subset of the RDF resources that are properties;

[4] A common convention is that class names are written with an initial uppercase letter, while properties are written with an initial lowercase letter.

- `rdfs:Class` is a generic concept of a Category (similar to the notion of a Class in object-oriented programming languages such as Java).

RDFS core properties:

- `rdf:type` is used to indicate that a resource is an instance of a class;
- `rdfs:domain` is used to state that any resource that has a given property is an instance of one or more resources;
- `rdfs:range` is used to state that all values of a property are instances of one or more classes.

RDFS defines two properties for annotating resources that provide guidance to potential users of the vocabulary and are relied upon by many Linked Data applications when displaying data:

- `rdfs:label` provides a human-readable name for a resource;
- `rdfs:comment` provides a human-readable description of a resource.

In the following section we demonstrate how we defined the Pordata Vocabulary by reusing Data Cube and Timeline classes of Data Cube to declare memberships of the Pordata concepts. The familiar already table presented on Figure 4.4 will be used to exemplify our description.

Definition of components

Components in Data Cube are represented as both instances of `rdf:Property` and the appropriate classes `qb:DimensionProperty`, `qb:AttributeProperty` and `qb:MeasureProperty`. Figure 4.12 illustrates the encoding of the Data Cube components that we need to represent the example table. The Pordata entities are highlighted in red, the Data Cube terms are in green.

We identified the following components in the example table:

- the measure component `porschema:hasMeasure` specifies the observed phenomenon of a table;
- attribute components: `porschema:hasUnitMeasure` defines the unit of measure, and `porschema:hasValueType` defines the type of an observation's value;
- dimension components: `por:refPeriod` locates an observation in the temporal dimension, and `porschema:refCountry_of_Origin` locates an observation in the series dimension.

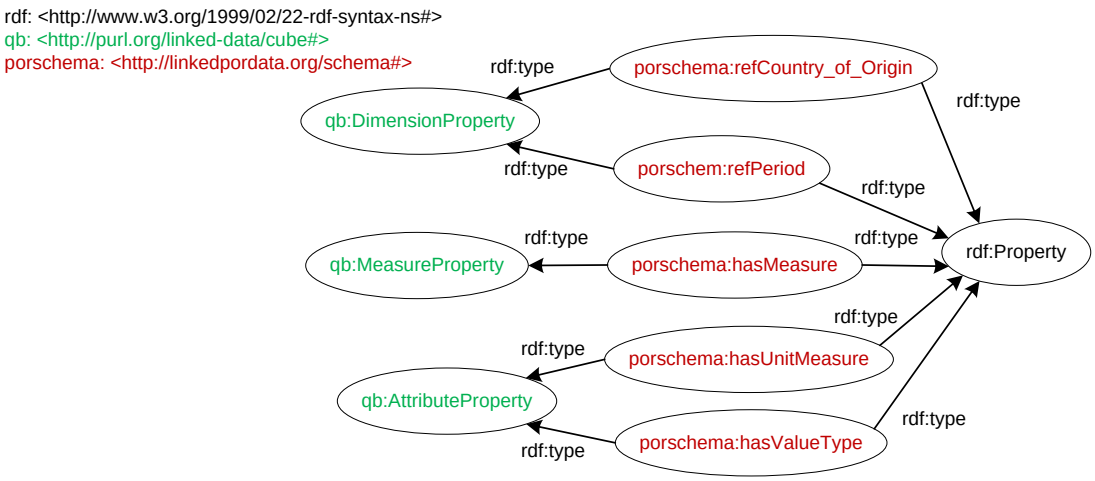


Figure 4.12: Components of the Pordata table *Screenings: total and by film's country of origin* in Data Cube.

We will use these components to connect the observation of the table to the corresponding values of dimensions and metadata. We defined the following generic classes that represents concepts of dimensions and metadata:

- `porschema:Measure` specifies a generic class of a measured phenomenon;
- `porschema:UnitMeasure` specifies a generic class of a unit of measure;
- `porschema:ValueType` specifies a type of an observation's value;
- `tl:Interval` is the class from the Timeline Ontology that represents a time interval;
- `porschema:Country_of_Origin` specifies a generic class that refers to a country of origin.

The representation of the possible values of components is done using the `rdfs:range` property as depicted by Figure 4.13.

Now, the observations of the example table have to be connected via the component properties to the instances of the corresponding classes as follows:

- `porschema:refCountry_of_Origin` connects observations to the instances of the class `porschema:Country_of_Origin`;
- `porschema:hasMeasure` connects observations to the instances of the class `porschema:Measure`;

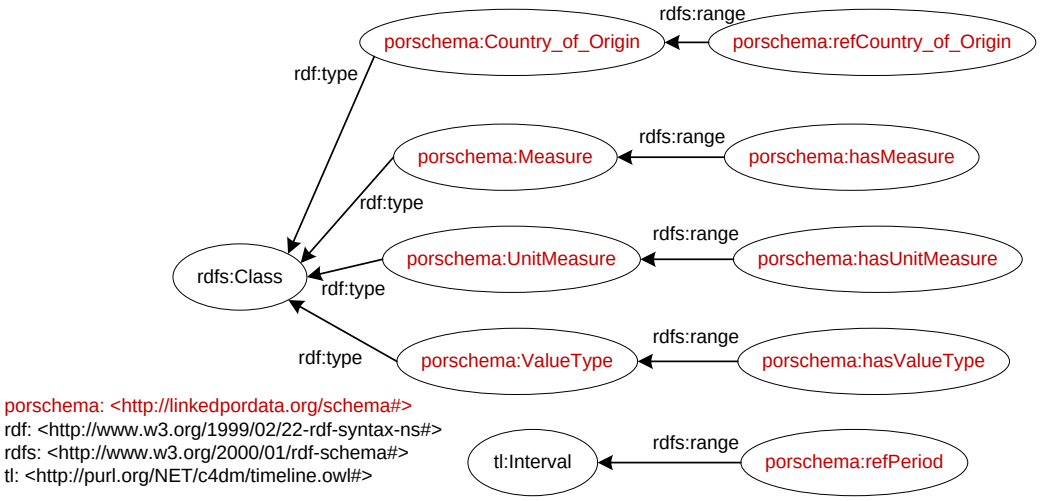


Figure 4.13: Ranges of the components of the Pordata table *Screenings: total and by film's country of origin* in Data Cube.

- `porschema:hasUnitMeasure` connects observations to the instances of the class `porschema:UnitMeasure`;
- `porschema:hasValueType` connects observations to the instances of the class `porschema:ValueType`;
- `porschema:refPeriod` connects observations to the instances of the class `tl:Interval`.

Definition of component specifications and the data structure definition

Figure 4.14 illustrates how we define a specification of `porschema:hasUnitMeasure`.

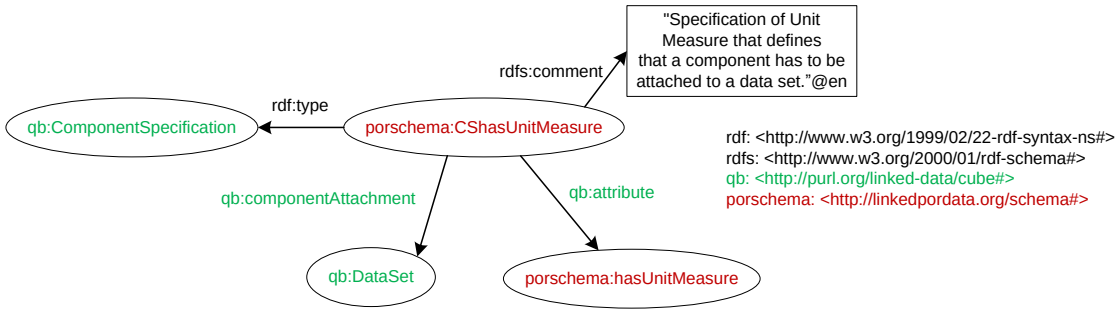


Figure 4.14: Specifications of the unit of measure component in Data Cube.

The specification `porschema:CShasUnitMeasure` references the attribute component and defines the level of attachment to the dataset. Other component specifica-

tions are defined similarly as follows:

- `porschema:CSrefCountry_of_Origin`, `porschema:CShasValueType` and `porschema:CSrefPeriod` simply reference corresponding components.

  By default, the corresponding components `porschema:refCountry_of_Origin`, `porschema:refPeriod` and `porschema:hasValueType` are attached to observations. Obviously, we have to know the values of both dimensions to locate an observation in the dataset, and each observation has its own type of value, we can not generalise these components at the dataset level.

- `porschema:CShasMeasure` and `porschema:CShasUnitMeasure` reference the corresponding components and set the level of attachment to the dataset.

  All the observations in a table have the same measure and unit of measure, it is better to define them once for the whole dataset than each time for every observation. All the specifications can be reused to create a data structure for other Pordata tables. If we want to encode a dataset with `porschema:hasUnitMeasure` attached to an observation, for example, we have to create a corresponding component specification for `porschema:hasUnitMeasure` and a different data structure that employs this specification.

Figure 4.15 illustrates how we defined the data structure definition of the running example is `porschema:dsd_Screening_by_country_of_film`.

Appendix B presents an encoding of the component properties, component classes and component specifications.

"Screenings: total and by film's country of origin" in Data Cube

Figure 4.16 illustrates the Data Cube encoding of the example dataset and the observation that *the number of screenings of Portuguese movies in Portugal in 1979 was 39.792*.

We defined custom concepts as follows:

- `porvocab:Number` is an instance of `porschema:UnitMeasure`;

- `porvocab:Screening` is an instance of `porschema:Measure`;

- `porvocab:Portugal` is an instance of `porvocab:Country_of_Origin`;

- `portime:Year1979` is an instance of `tl:Interval`;

- `porvocab:Normal` is an instance of `porschema:ValueType`.

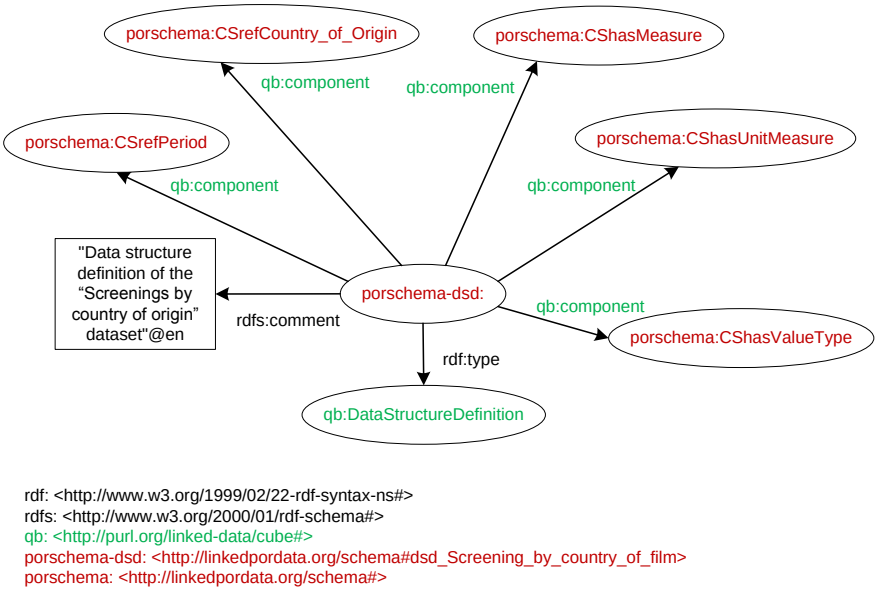


Figure 4.15: Data structure definition of *Screenings: total and by film's country of origin* in Data Cube.

Their exact definitions are presented in Appendix C.

`pordata:obs-1` is the observation that "the number of screenings of movies from Portugal happened Portugal in 1979 was 39.792". We associated the observation to the corresponding dataset via `qb:dataSet`. The dataset, in turn, is associated with the data structure definition that was defined before via `qb:structure`. According to this data structure definition, we specified the unit of measure (`porvocab:Number`) and the measured phenomenon (`porvocab:Screening`) at the dataset level. The temporal dimension (`portime:Year1979`), the country of origin (`porvocab:Portugal`), the type of the observation's value (`porvocab:Normal`) and the value of the observation (`39.972`) were attached to the observation.

4.3 Publishing Pordata

The *Data Publication* steps are purely technical. The realisation of these steps should be done after selecting a publishing pattern. A publishing pattern is a set of Linked Data design considerations that provide a guidance on how to apply the Linked Data principles to Pordata and complement, rather than replace, the existing data management infrastructure. Namely, a publishing pattern should define how to generate RDF out of the original data format, how to store it and how to deploy Linked Data. Understanding the publishing pattern helps to select the existing tools and technologies that

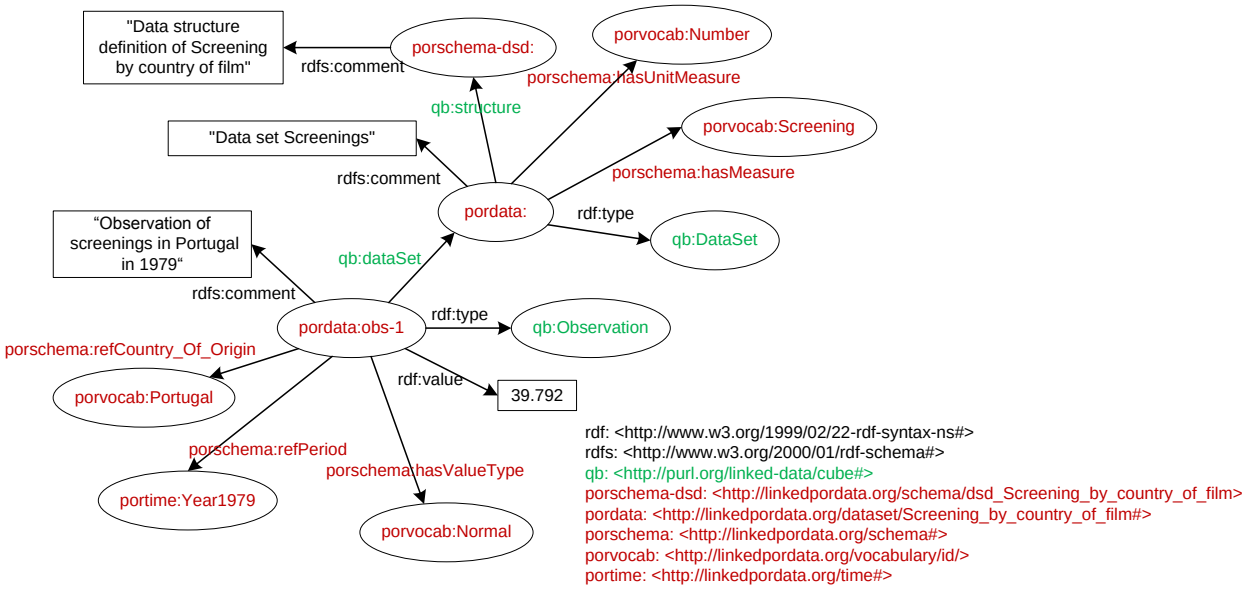


Figure 4.16: *Screenings: total and by film's country of origin* and the fact that *the number of screenings of Portuguese movies in Portugal in 1979 was 39.792* in Data Cube.

assist in the process of Linked Data publication.

4.3.1 Selecting a Publishing Pattern

In general, the choice of a publishing pattern is dictated by the original format of the data. Different formats imply different technological solutions for RDF generation. Pordata resides in relational databases.

Significant work has been done in the area of mapping relational databases to RDF; various manual and semi-automatic approaches have been proposed [79]. There is an ongoing activity to develop a standard mapping language, for mapping relational databases to RDF, *W3C RDB2RDF Working Group* [96]. The group produced *"A Survey of Current Approaches for Mapping of Relational Databases to RDF"* [102] that captures capturing existing tools that propose their specific solutions to map relational data to RDF. Existing tools implement dynamic conversion of relational data into RDF. Thus, RDF is not physically generated but provided as views over the relational data via pre-defined mappings that specify how to construct subjects, predicates and objects of RDF triples out of relation schema and data (i.e., tables, columns and cells). Most of the existing systems generate a new RDF vocabulary for each database using tables names as class names and column names as property names. However, there are solutions that allow to adjust mappings to reuse the terms from the existing vocabulary. In our case we are interested in the latter, because we want to map Pordata in terms of the

Pordata Vocabulary.

Among the existing tools that were examined, *OpenLink Virtuoso* [120] and *D2R Server* [21] enable customization of the automatically generated mappings. We chose Virtuoso because it is also a middleware and database engine hybrid that combines the functionality of a traditional RDBMS, RDF storage, web application server and file server functionality. In addition to this, we found useful the Virtuoso Sponger functionality that enables dereferencing of URIs at run time. For example, by means of the Virtuoso Sponger we could access datasets that do not provide SPARQL-endpoints and integrate their information with Linked Pordata (Section 5.2.2).

We centred the publishing pattern around the Virtuoso Universal Server. Figure 4.17 illustrates our solution. There are three phases: data preparation, RDF generation and data publication. The data preparation work includes the creation of the simulated Pordata. We created the database in the Virtuoso RDBMS. The RDF representation of the relational data will be generated by means of the Virtuoso RDB-to-RDF mappings. RDF representation of Pordata then is available via the virtual RDF storage that can be accessed through the Virtuoso SPARQL-endpoint. The Virtuoso URL rewriter will be used to make URIs identifying Pordata concepts dereferenceable into their RDF descriptions. For this, the rewrite change the requested HTTP URI into corresponding SPARQL query. SPARQL is the query language for RDF (we discuss SPARQL in Section 4.3.3). The query is posted over the virtual RDF storage, where it is rewritten into the corresponding SQL query using the corresponding mappings. The SQL query is run over the Pordata database and the result of the query is returned back as the RDF description of the requested HTTP URI.

The Pordata simulated database

Our proposal is based on the fact that the original data resides in relational databases. Keeping the relational databases as a starting point allows as to preserve the existing Pordata management infrastructure. To develop our proposal, since we don't have direct access to the Pordata databases, we implemented a database that simulates it. In order to populate the database we downloaded several Pordata tables as XLS files (Section 2.1.2) and run a Python script to process the XLS files and output corresponding SQL insert statements[5]

We designed the database after analyzing the Pordata data in Chapter 2. Figure 4.18 depicts the simulate database model.

`Tables` stores metadata about Pordata tables. `Series` stores available series.

[5]The scripts for creating and populating the database can be downloaded from `http://linkedpordata.dyndns.org/static/SQL.tar.gz`.

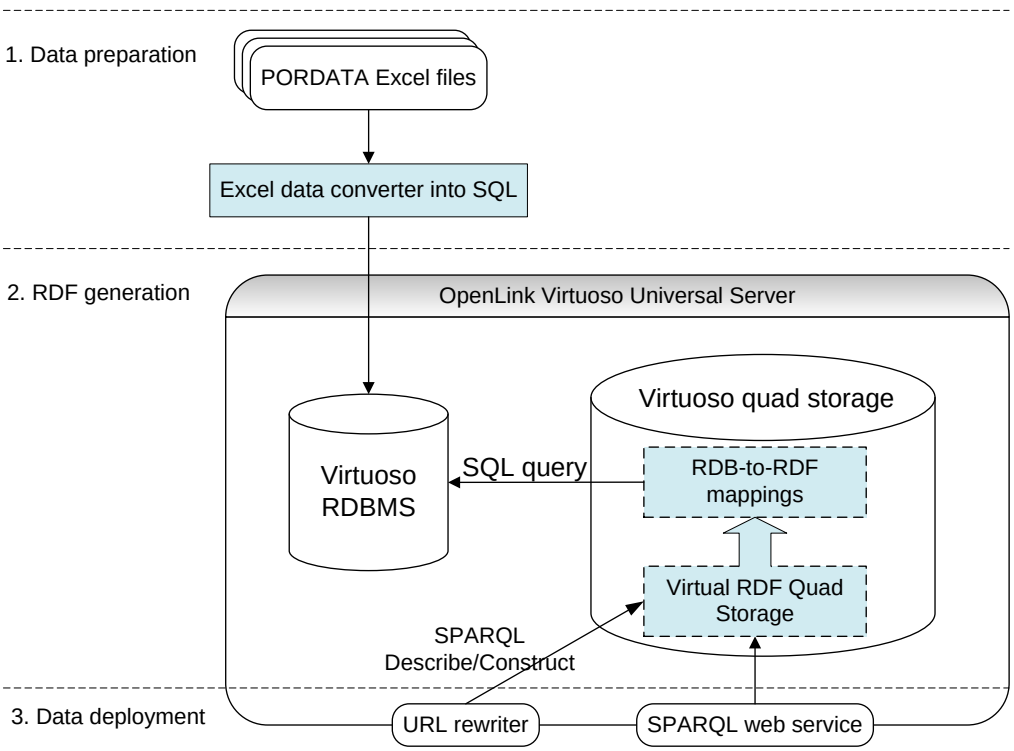


Figure 4.17: Linked Pordata: publishing pattern

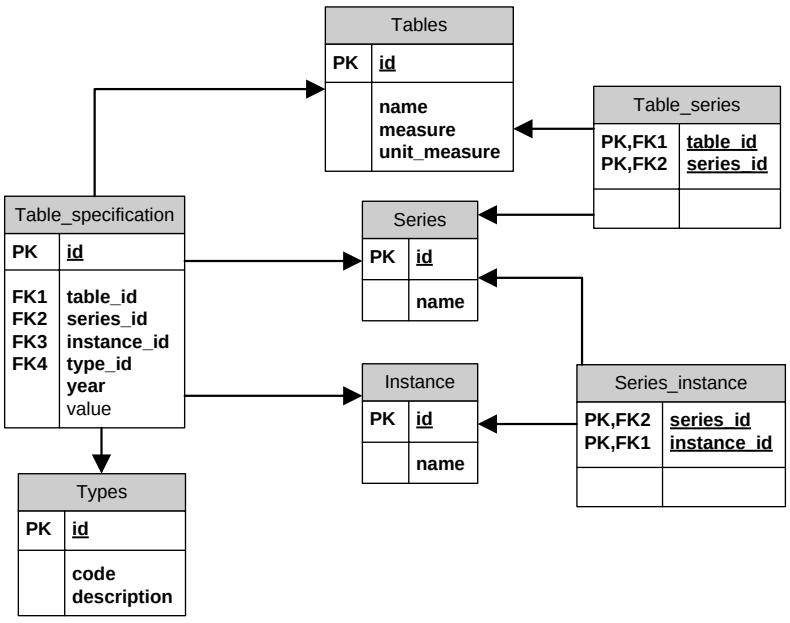


Figure 4.18: Simulated Pordata Database model

`Table_series` stores information what tables have what series. `Instances` stores available instances of series. `Series_instance` stores information what series

have what instances. For example, in order to store data about *Culture and Sports/Cinema/Screenings: total and by film's country of origin* (Figure 4.4), we need to add one record to `Tables` with values: `name='Screening by country of film'` and `measure='Screening', unit_measure='Number'`; one record to `Series` with `name='Country of Origin'`; eight records to `Instance` with names `Total`, `Portugal`, `Spain`, `France`, `United Kingdom`, `USA`, `Other countries` and `Co-productions`; corresponding pairs `{table_id}-{series_id}` to `Table_series` and corresponding pairs `{series_id}-{instance_id}` to `Series_instance`.

`Types` stores data about different types of statistical values defined in the Tables' Symbology (e.g., "not available"). Finally, `Table_specification` is used to store actual data. Each record in this table refers to one cell of a Pordata table. For example, to insert the fact that *"in 1979 the number of screenings of Portuguese movies was 39.792"*, we would insert a record into `Table_specification` with the corresponding `table_id`, `series_id`, `instance_id`, `type_id` and `value=39.792`.

4.3.2 Generating RDF

In this section we explain how we developed RDF representation of Pordata in terms the Pordata Vocabulary. We can divide this work into two parts: manual and automatic generation of RDF.

Manual RDF generation Recall, in Section 4.2.1 we defined two types of Pordata entities: structural and data entities. We said that structural concepts are generic for all Pordata tables, whereas data concepts are specific to each table. Manually we defined the Pordata structural concepts. In Section 4.2.3 we demonstrated how to use the Data Cube vocabulary to encode structure concepts of Pordata. Here we only summarise the structural components that we defined manually. Further, we demonstrate how we used them to encode data concepts via the Virtuoso RDB-to-RDF mappings.

We defined the following structural components:

- `qb:refPeriod`, a time dimension component;
- `qb:hasUnitMeasure`, a unit of measure component;
- `qb:hasMeasure`, a measure component;
- `qb:hasValueType`, a type of a value of an observation.

For each component we defined the corresponding class:

- `porschema:UnitMeasure`, the class of unit of measurements;

- `porschema:Measure`, the class of measured phenomena;

- `porschema:ValueType`, the class of types of observations' values.

We used these classes to define instances of metadata specific to each table. For example, `porvocab:Number` will be defined as instance of `porschema:UnitMeasure` and all the observations of the table *Screening by country of film* will be attached this instance via the component `qb:hasUnitMeasure`. To define instances of the time dimension component we used the class of the Timeline Ontology `tl:Interval`.

For each component we defined its specification:

- `porschema:CSrefPeriod`, the specification of the time dimension component;

- `porschema:CShasUnitMeasure`, the specification of the unit of measure component;

- `porschema:CShasMeasure`, the specification of the measure component;

- `porschema:CShasValueType`, the specification of the type of a value of an observation.

Encoding of all the concepts discussed above is presented in Appendix C.

Automatic RDF generation

Virtuoso provides functionalities to map relational data into RDF and allow the RDF representation of the relational data to be customised. To define mappings Virtuoso includes a declarative Meta Schema Language [42]. The mappings are dynamic. Consequently, changes to the underlying data are reflected immediately in the RDF representation. No changes are required to the underlying relational schema.

Virtuoso implements a classic "table-to-class", "column-to-predicate" approach to transform relational data to RDF as follows. For example, consider the Pordata table *Tables* given in Figure 4.19a. Assume that *Tables* contains three rows as Figure 4.19b suggests.

By applying the Virtuoso mappings to *Tables* we want to construct the corresponding RDF graph depicted in Figure 4.20[6].

[6]The figure reflects the relational data only for *Screening by country of film*, as if the mappings processed only the third row of the table. The other rows are mapped similarly.

Tables	
PK	**id**
	name **measure** **unit_measure**

(a) *Tables* schema.

Tables			
id	**name**	**measure**	**unit_measure**
1	Deflators (base=2006) (R)	Deflator	Decimal
2	Inflation Rate (Growth Rate - Consumer Price Index)	Inflation Rate	Percentage
3	Screening by country of film	Screening	Number

(b) *Tables* data.

Figure 4.19: Unambiguous interpretation of non-information resources.

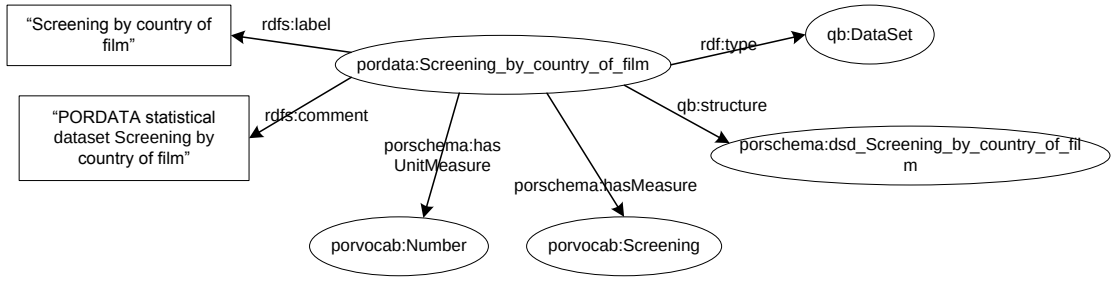


Figure 4.20: The target RDF for *Tables*:*Screening by country of film*.

Tables defines a particular class of entity: statistical tables. Each column in the table defines one of the attributes of the entity. We identified the following attributes of statistical tables: name, measure and unit of measure. Each row in *Tables* describes an entity instance of the table that has certain values of the attributes and is uniquely identified by the primary key. For example, the row that is identified by id `3` is an instance of *Tables* and has name `'Screening by country of film'`, measure `Screening` and unit of measure `Number`.

An RDF triple consists of a subject, a predicate and an object. If we apply the terminology of relational data, we could say that the constituent parts of an RDF triple are: the subject, attribute of the subject and the attribute value. Thus, in the "table-to-class", "column-to-predicate" approach a table is represented as an RDF class, e.g., `pordata:Screening_by_country_of_film` is a `qb:DataSet` class. Columns of the table are converted into predicates, e.g., `porschema:hasUnitMeasure`. An entity of the table is represented as collection of RDF triples with the same subject, where the predicates relate the subject to different attribute values of the entity, e.g., `porschema:hasUnitMeasure` relates the subject to `porvocab:Number`.

Thus, Virtuoso draws the following parallels between the relation data and RDF:

- a table is an RDF class;

- primary keys and non-key columns are assigned URIs;
- each row of a table is assigned the `rdf:type` predicate that maps it to the corresponding RDF class;
- for each column of a table we can construct a triple as follows:

subject:	`the primary key's URI`
predicate:	`the column's URI`
object:	`the column's value`

Quad map patterns The Virtuoso transformations of relational data into RDF are described by constructs called *quad map patterns*. They resemble RDF triples, just with unknown components (subject, predicate or/and object). One quad map pattern generate one kind of RDF triples. For example, the following pattern can be used to generate RDF triples that states, that "Screenings by country of film" table has unit of measure "Number":

subject:	`dataset URI`
predicate:	`porschema:hasUnitMeasure`
object:	`measure URI`

The pattern above can match, for example, the following RDF triple:

subject:	`pordata:Screening_by_country_of_film`
predicate:	`porschema:hasUnitMeasure`
object:	`porvocab:Number`

In order to generate proper URIs for each quad map pattern, Virtuoso uses IRI and Literal classes.

IRI classes Virtuoso manages the conversion of column values to URIs using the construct called `IRI classes`. IRI class is a feature of the Virtuoso Meat Schema Language implemented as extension to SPARQL (we will discuss SPARQL, the query language to RDF, in Section 4.3.3). An IRI class defines how a column gets converted into a URI. Consider the following listing that the IRI class `pdt:measure_iri` that maps the column value *measure* into URI:

```
prefix pdt: <http://pordata/schemas#>
create iri class pdt:measure_iri
"http://linkedpordata./vocabulary/%s"
                (in measure varchar not null) .
```

The IRI class consists of two parts: the URI pattern and the input parameter. The URI pattern is a sprintf style format string that constructs the URI based on the input parameter. The input parameter is the column value. For example, if we pass the column *measure* of *Tables*, `pdt:measure_iri(tables.measure)`, we obtain three URIs, one per each row of the table:

```
"http://linkedpordata.org/vocabulary/Deflator"
"http://linkedpordata.org/vocabulary/Inflation Rate"
"http://linkedpordata.org/vocabulary/Screening"
```

However, the second URI is not valid, as it contains the space. To handle such cases and to allow more sophisticated transformation, IRI class can make use of user-defined functions. Consider, for example, the following function:

```
create function DB.PORDATA.MEASURE_IRI
                  (in _measure varchar){
return sprintf_iri('http://%s/vocabulary/%s',
                     'http://linkedpordata.org/',
                     replace(_measure, ' ', '_'));
};
```

It takes a string as input and returns back the same string with spaces replaced by underscores. The IRI class `pdt:measure_iri` that makes use of this function looks as follows:

```
prefix pdt: <http://pordata/schemas#>
create iri class pdt:measure_iri
using function DB.PORDATA.MEASURE_IRI
 (in measure varchar not null) returns varchar.
```

Another important role of functions in IRI classes is to enable transformations in both directions: column values into URIs and URIs into column values. For this, Virtuoso defined inverse functions. The inverse function for `DB.PORDATA.MEASURE_IRI` that transform a measure URI into the corresponding column value looks as follows:

```
create function DB.PORDATA.MEASURE_IRI_INVERSE
                       (in measure_iri varchar){
 declare parts any;
 parts := sprintf_inverse (measure_iri,
 'http://http://linkedpordata.org/'||'/vocabulary/%s', 0);
 if (parts is not NULL)
  { return replace(cast(parts[0] as varchar), '_', ' '); }
 return NULL;
```

```
};
```

Finally, the IRI class `pdt:measure_iri` that makes use of both functions looks as follows:

```
prefix pdt: <http://pordata/schemas#>
create iri class pdt:measure_iri using
function DB.PORDATA.MEASURE_IRI
 (in measure varchar not null) returns varchar ,
function DB.PORDATA.MEASURE_IRI_INVERSE
 (in measure_iri varchar)
            returns varchar option (bijection) .
```

Literal classes To define literal objects of triples, Virtuoso uses the construct called *literal class*. They are defined similarly to IRI classes. For example, the following is the definition of the literal class that defines the comment to `pordata:Screening_by_country_of_film`:

```
create literal class pdt:dataset_comment using
 function DB.PORDATA.DATASET_COMMENT
          (in id integer not null) returns varchar ,
 function DB.PORDATA.DATASET_COMMENT_INVERSE
          (in dataset_comment varchar)
                returns integer option (bijection) .
```

The function that performs direct RDB-to-RDF transformation looks as follows:

```
create function DB.PORDATA.DATASET_COMMENT
                         (in _id integer) {
 declare _name varchar ;
 _name := (select name
             from DB.PORDATA.TABLES
            where id = _id);
 return sprintf('PORDATA statistical dataset %s',
                 cast(_name as varchar));
};
```

Identity class is a special case of literal classes. It converts a value from a SQL varchar column into an untyped literal and a value from a column of any other SQL datatype into an XML Schema typed literal i.e. `xsd:integer`, `xsd:dateTime` and so on. We will see examples of identity classes later.

Quad storage Quad map patterns are grouped into a *quad storage* that has a URI associated with via the `graph` word. For example, the following listing shows how we grouped quad map patterns related to *Tables* under the quad storage `http://linkedpordata.org/pordata`

```
1  prefix pdt: <http://pordata/schemas#>
2  prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3  prefix qb: <http://purl.org/linked-data/cube#>
4
5  from "DB"."PORDATA"."measure" as measures
6  from "DB"."PORDATA"."unit_measure" as unit_measures
7  from "DB"."PORDATA"."tables" as tables
8
9  graph iri ("http://linkedpordata.org/pordata")
10 {
11  pdt:dataset_iri(tables.id) qb:structure pdt:dsd_iri(tables.id).
12  pdt:dataset_iri(tables.id) rdfs:label tables.name .
13  pdt:dataset_iri(tables.id) rdfs:comment
14                      pdt:dataset_comment(tables.id) .
15  pdt:dataset_iri(tables.id) qb:structure
16                             pdt:dsd_iri(tables.id) .
17  pdt:dataset_iri(tables.id) porschema:hasMeasure
18                     pdt:measure_iri(tables.measure) .
19  pdt:dataset_iri(tables.id) porschema:hasUnitMeasure
20           pdt:unit_measure_iri(tables.unit_measure) .
21 }
```

- The IRI class `pdt:dsd_iri` was implemented to generate corresponding URIs of data structure definitions (cf. line 16), e.g., `porschema:dsd_Screening_by_country_of_film` for the running example.

- Note how we defined the label for the dataset (cf. line 12) by simply using the identity class. Namely, by simply passing the value of the column that will be converted into the corresponding plain literal. The comment for the dataset requires more sophisticated transformation. For this, we defined the literal class `pdt:dataset_comment` (cf. lines 13-14).

- *Measure* and *unit_measure* of the table require URIs as well. However, IRI classes can be used only with key columns or columns with unique constraints. To overcome this issue we created two views over *Tables*, *Measures*

and *Unit_Measure*, which contain unique values of the *measure* and *unit_measure* columns correspondingly. Corresponding IRI classes, `measure_iri` and `unit_measure_iri`, were created to operate on these views and generate proper URIs (cf. lines 18,20).

As a result of the transformations discussed in this section we generated the virtual RDF graph `http://linkedpordata.org/pordata` partially depicted by Figure 4.20.

Similarly, we defined the graph `http://linkedpordata.org/porschema` to store data about specific series dimensions; the graph `http://linkedpordata.org/vocabulary` to maintain the proprietary terms and the graph `http://linkedpordata.org/portime` to define years. All these graphs were further combined into one single virtual graph `virtrdf:DATACUBE`[7].

All the IRI and literal classes created for the running example can be consulted in Appendix D.

4.3.3 Deploying Linked Pordata

The basic means to access Linked Data on the Web is to dereference HTTP URIs in RDF descriptions. There are two alternative ways to make Linked Data sets accessible on the Web: *RDF dumps* and *SPARQL endpoints*. The RDF dump of a Linked Data set is simply an RDF file (or multiple RDF files) containing the RDF graph which describes the whole dataset. Data publishers can provide RDF dumps for downloading. In this section we, first, introduce SPARQL, the language to query RDF. Second, we discuss how we can used Virtuoso to make the Pordata URIs dereferenceable.

SPARQL

SPARQL is a W3C standardized language to query RDF data[8] [94].

Assume, we have the following RDF graph:

```
porvocab:Portugal rdf:type porschema:Country_of_Origin .
```

and we want to know what countries of origins the RDF graph defines. Intuitively, we would write something like:

```
?country rdf:type porschema:Country_of_Origin .
```

[7]The definition of the `virtrdf:DATACUBE`, relevant IRI and literal classes and functions can be downloaded from `http://linkedpordata.dyndns.org/static/QuadStorage.tar.gz`

[8]SPARQL is a recursive acronym that stands for **S**PARQL **P**rotocol **a**nd **R**DF **Q**uery **L**anguage

and try to find a value for `?country` that appears in the RDF graph, e.g., `porvocab:Portugal`.

We call a *basic graph pattern* (*BGP*) a set of triple patterns. *Triple patterns* are RDF triples, except that they may contain *variables* at the *subject*, *predicate* or *object* positions.

Solutions are RDF graphs per se. They represent a matching sub-graph in the queried RDF graph. The sub-graph is constructed by a solution mapping that binds query variables to the terms from the given RDF graph. The following is the solution obtained by running our query over the given RDF graph (the variable `?country` is bound to the term `porvocab:Portugal`):

?country
$porvocab : Portugal$

SPARQL syntax SPARQL has an SQL-like syntax. The complete specification of the SPARQL syntax is given in [94] in Section 4. Here we will only highlight the aspects of the syntax crucial for our work.

A SPARQL query has two mandatory clauses:

- `SELECT`: specifies what subset of the variables to return;

- `WHERE`: defines BGP to find a match for it in the queried dataset.

Additionally, optional `PREFIX` can be used to declare prefixes that are used in a SPARQL query. There are also optional query modifiers with the same meaning as in SQL: `DISTINCT`, `ORDER BY`, `LIMIT`, `OFFSET`, and others.

The query from the example above in SPARQL looks as follows:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX porschema: <http://linkedpordata.org/schema#>
SELECT ?country
WHERE {?country rdf:type porschema:Country_of_Origin .}
```

SPARQL queries are executed against an *RDF dataset* that represents a collection of RDF graphs. An RDF dataset always contains one `DEFAULT GRAPH`, which does not have a name and comprises all the RDF graphs defined in the dataset. Additionally, the RDF dataset can have zero or more named graphs. A `NAMED GRAPH` is an RDF graph which is assigned a name in the form of a URI. For example, assuming our dataset already contains the graph defined above, we can add the following RDF graph in the dataset with the name `http://linkedpordata.org/vocabulary`:

```
porvocab:Portugal rdfs:label "Portugal" .
```

When we run a SPARQL query against an RDF dataset, by default it is executed against the default graph of the dataset. Thus, by running the query from above against our RDF dataset, we get the solution that binds `?country` to `porvocab:Portugal`.

?country
$porvocab : Portugal$

FROM vs FROM NAMED There are two optional clauses, `FROM` and `FROM NAMED`, that can be used to influence the default behaviour of the query answering mechanism.

`FROM` specifies the URI of the named graph to be queried. A SPARQL query can contain many `FROM` clauses. All the graphs from the clauses are merged together and replace the content of the default graph. For example, consider the following query that replaces the content of the default graph with the named graph `http://linkedpordata.org/vocabulary`:

```
SELECT ?country
FROM <http://linkedpordata.org/vocabulary>
WHERE {?country rdf:type porschema:Country_of_Origin .}
```

Running this query against our dataset does not give any solutions, since there is no match for the triple pattern in the named graph.

`FROM NAMED` specifies the named graph in sub-queries. A SPARQL query can contain many `FROM NAMED` clauses, however, as opposed to the previous clause, they do not change the default graph, but allows portions of a SPARQL query to match against the named graphs. Anything outside the named graph matches against the default graph. For example, consider the following query:

```
SELECT ?countryName ?type
FROM NAMED <http://linkedpordata.org/vocabulary>
WHERE {
GRAPH ?g {?country rdfs:label ?countryName .}
?country rdf:type ?type .
}
```

The clause `GRAPH` is used to define the pattern of the sub-query that can contain the graph variable (in addition to the subject, predicate and object variables). The solution to this query binds `?g` to the only named graph we have in our dataset and match the triple pattern against the content of this graph. Thus, `?country` is bound to `porvocab:Portugal` and `?countryName` to `Portugal`. As we said, everything that is outside of the sub-query is matched against the default graph. Since the default graph remains unchanged, the solution will also bind `?type` to `porschema:Country_of_Origin`:

?g	?countryName	?type
$< http://linkedpordata.org/vocabulary >$	$Portugal$	$porschema : Country\_of\_Origin$

DESCRIBE vs CONSTRUCT SPARQL can be used not only to query RDF data, but to *describe* resources or *construct* new graphs.

The `DESCRIBE` clause is used to return a single RDF graph describing resources of interest. It can be used when we do not have enough information about the RDF dataset to properly query it, e.g., we might not know all the predicates. The scope of the RDF graph that is returned by `DESCRIBE` is determined by the SPARQL query processor. For example, the following query can be used to obtain an RDF graph constructed by Virtuoso with relevant information about *Portugal* defined in the Linked Pordata:

```
DESCRIBE <http://linkedpordata.org/vocabulary/Portugal>
```

The result of the query is the RDF graph[9]:

```
@PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#> .
@PREFIX owl: <http://www.w3.org/2002/07/owl#> .
@PREFIX skos: <http://www.w3.org/2004/02/skos/core#> .
@PREFIX porvocab: <http://linkedpordata.org/vocabulary/> .
@PREFIX pordata:
  <http://linkedpordata.org/dataset/Screening_by_country_of_film#> .
@PREFIX porschema: <http://linkedpordata.org/schema#> .

porvocab:Portugal rdf:type skos:Concept .
porvocab:Portugal rdf:type porschema:Country_of_Origin .
porvocab:Portugal rdfs:label "Portugal" .
porvocab:Portugal owl:sameAs
          <http://data.linkedmdb.org/resource/country/PT> .
pordata:obs-543 porschema:refCountry_of_Origin porvocab:Portugal .
...
```

The `CONSTRUCT` option can be used to specify a graph template we want to obtain. We can extract parts or the whole graph from the RDF dataset. For example, in the resulting graph above we have triples that were defined in the virtual graph `http://linkedpordata.org/porschema` (Section 4.3.2). The following query returns back this virtual graph completely:

[9]The complete RDF graph contains more triples that we present here.

```
CONSTRUCT { ?s ?p ?o }
WHERE { GRAPH <http://linkedpordata.org/porschema>
      { ?s ?p ?o } .}
```

Virtuoso SPARQL-endpoint *SPARQL-endpoint* is an HTTP-based query service that can be accessed using the SPARQL protocol [27], a W3C standard method for remote invocation of SPARQL queries over a Linked Data set. Virtuoso provides a built-in SPARQL-endpoint that allows to query both physical and virtual RDF storages. Thus, the RDF representation of Pordata is available for querying via the Virtuoso SPARQL-endpoint. We will use it in our work in Chapter 5 to integrate Pordata with other Linked Open Data sets. Next, we discuss how the Virtuoso SPARQL-endpoint can be used to make the Pordata URIs dereferenceable.

Dereferencing the Pordata URIs with Virtuoso

HTTP URIs are used to serve two essential roles in Linked Data deployment: identify real-world entities and provide access to their descriptions on the Web. Descriptions of resources on the Web are embodied in the form of documents. In the traditional Web HTML is the primary format to represent information resources (e.g., Web pages). On the Web of Data this function is served by RDF documents. As we discussed in Section 4.2.2, we need to use separate URIs to identify real-world entities and documents describing them, so that they are not mixed up. Linked Data defines two approaches for this: hash URIs and 303 URIs. Both approaches allow to make the URIs of the real-world entities dereferenceable and distinguishable from the URIs of the documents describing them.

In Section 4.2.2 we explained how we adopted both strategies to design HTTP URIs to name Pordata entities. When URIs identifying these entities are requested from Virtuoso we need to provide descriptions of these entities in the form of RDF documents. However, in our proposal we generate RDF representation dynamically, and there are no physical RDF documents to return back. Virtuoso proposes to use the URL-rewriter mechanism that allows dynamic generation of RDF descriptions of the requested HTTP URIs identifying Pordata concepts.

Virtuoso URL-rewriter The URL-rewriting mechanism is based on modifying a requested HTTP URI prior to the final processing of that URI by a Web Server. URL-rewriting capabilities are provided by many traditional Web servers for different purposes, e.g., create short URIs out of long ones [113].

Virtuoso, as a full-blown HTTP server, also has the URL-rewriting functionality [117]. The Virtuoso URL-rewriter allows to translate incoming HTTP URI requests into corresponding SPARQL DESCRIBE or CONSTRUCT queries. The queries are run over the Virtuoso RDF storage to obtain the descriptions of the requested concepts. The results of the queries are sent back to requesters along with the `200 OK` status code. Figure 4.21 represents an example of communication between a Web client and the Virtuoso Server.

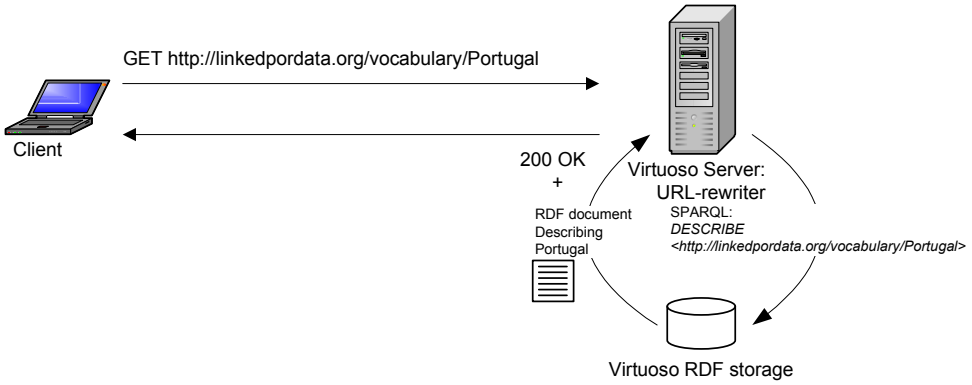


Figure 4.21: Communication between a Web client and the Virtuoso Server.

A Web client requests `http://linkedpordata.org/vocabulary/Portugal` from Virtuoso. The Virtuoso URL-rewriter constructs a corresponding SPARQL query and runs it against the Virtuoso RDF storage. By means of the pre-defined Virtuoso RDB-to-RDF mappings (Section 4.3.2) the RDF view over the Pordata database is created to answer the query. The result of the query is returned back to the Web client as the RDF document describing Portugal.

Matching rules To be able to construct appropriate queries out of requested HTTP URIs, the Virtuoso rewriter defines *matching rules*. Each rule contains a regular expression, the *source pattern*, that matches the path of a requested URI, i.e., the part of the URI without the domain name. The matched part is passed to the *destination path pattern*, a template of a corresponding CONSTRUCT or DESCRIBE SPARQL query. Next, we discuss how we defined matching rules for hash and 303 URIs.

Hash URIs According to the HTTP protocol [48], when a Web client looks up a hash URI, the fragment part is stripped off, and the rest of the URI is requested from the server. For example, when a Web client looks up `http://linkedpordata.org/schema#Country_of_Origin`, the fragment part `#Country_of_Origin` is truncated. The rest,

`http://linkedpordata.org/schema`, is requested from Virtuoso. The source pattern `(/[^#]*)` matches `http://linkedpordata.org/schema` and passes the matched part, `/schema`, to the destination path pattern. We can use `CONSTRUCT` queries to obtain descriptions of the Pordata concepts identified by hash URIs. Thus, the following query can be used to construct an RDF graph describing the concept *Country of origin*:

```
CONSTRUCT { ?s ?p ?o }
WHERE { GRAPH <http://linkedpordata.org/schema>
      { ?s ?p ?o } .}
```

Figure 4.22 illustrates how the requested URI is processed by the Virtuoso URL-rewriter.

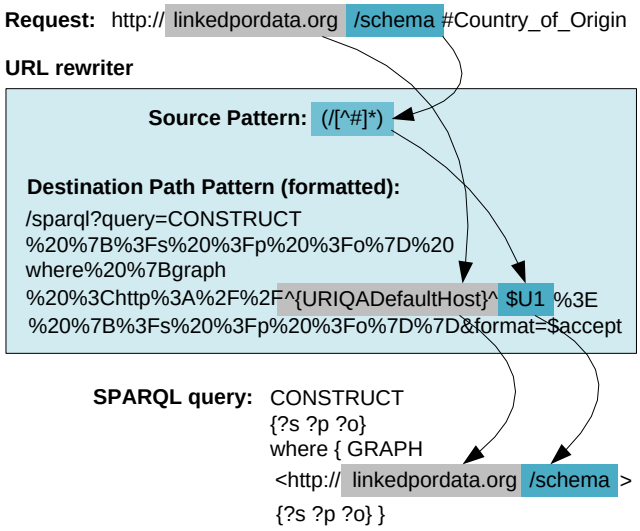


Figure 4.22: Virtuoso URL-rewriter for hash URIs.

303 URIs For 303 URIs we can use corresponding `SPARQL DESCRIBE` queries. For example, with the source pattern `/vocabulary(/[^#]*)` we can match vocabulary URIs, i.e., URIs that were defined in the `http://linkedpordata.org/vocabulary` virtual graph. This pattern matches `http://linkedpordata.org/vocabulary/Portugal` and passes the matched part, `/vocabulary/Portugal`, to the destination path pattern to construct the corresponding query:

```
DESCRIBE <http://linkedpordata.org/vocabulary/Portugal>
```

Figure 4.23 illustrates how the requested URI is processed by the Virtuoso URL-rewriter.

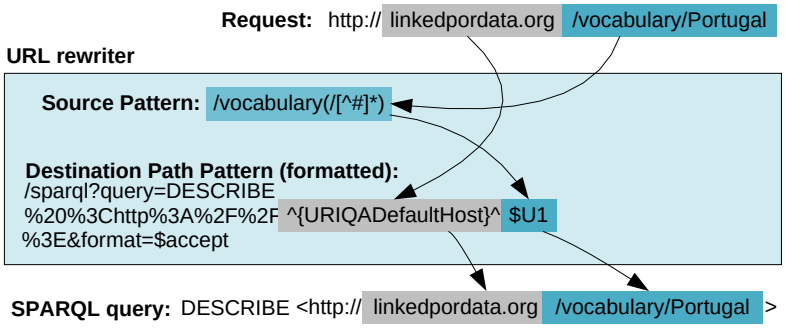


Figure 4.23: Virtuoso URL-rewriter for 303 URIs.

4.4 Chapter summary

In this chapter we described our proposal for publishing Pordata as Linked Data. We presented the methodology that we followed to develop the proposal. Our methodology consists of the two main steps: conceptual modelling of Pordata and publishing of the Linked Pordata. In the rest of the chapter we described both steps in detail. During the modelling step we explained our design considerations concerning the adoption of the Linked Data principles to use HTTP URIs to give names to real-world concepts and RDF to provide machine-readable descriptions of them. We started by analysing the Pordata dataset and extracting relevant concepts that capture the Pordata data. We identified two basic types of concepts: structural and data concepts. Structural concepts are generic concepts that basically can be used to encode any statistical table, not necessarily Pordata one. We identified the following basic structural concepts: *Table* and *Observation*. Indeed, raw statistical data is represented in the form of a summary table, where each cell refers to a single observation. Each observation within a particular statistical table is characterised by a set of dimension values. Thus, dimensions are other basic structural concepts that we need to encode structure of statistical data. We identified the concept *Temporal dimension* that is also generic to all statistical tables, as statistics are always collected for a certain time frame. The concept of *Series dimension* refers to statistical series to which statistics are applied. We argued that this concept is not generic, since different statistical tables have different series. Metadata is important to any statistics. We identified a set of generic structural concepts that can be used to encode an arbitrary statistical table: *Unit of measure*, *Measured phenomenon* and *Type of value*. Data concepts are instances of the structural concepts, e.g., *Screening by country of film* is the instance of the concept *Table*, *Number* is the instance of *Unit of measure*, *Portugal* is the instance of *Series dimension*, etc. The next step in the modelling process was to give names to the identified concepts. The Linked Data principles propose to

use HTTP URIs for this. We referred to entities of the real world as non-information resources, as opposed to descriptions of these entities that are also called information resources. We argued that on the Web of Data we have both of them and we employ the same technology, HTTP URIs, to identify them. Not to confuse information resources with non-information resource, as they are different things, the URIs identifying them should be different. On the other hand, the URI identifying a real-world concept must be dereferenceable, that is when someone looks up such URI on the Web a description of the concept should be retrieved. We introduced two strategies that are used on the Web of Data to ensure unambiguous identification of the real-world concepts and their descriptions: hash URIs and 303 URIs. We proposed the Pordata URI scheme where we explained how we adopted these strategies to name the Pordata concepts. Having the concepts named, we needed to describe them in such a way that machines can process their descriptions and "understand" their meanings. We introduced the RDF data model that is proposed by the LD principles for this. RDF allows to describe resources in the form of simple sentences: $< subject\ predicate\ object >$. As the best practices for publishing Linked Data suggest, we examined the existing vocabularies which can be used to model statistics: SCOVO and Data Cube. We chose Data Cube to encode Pordata, because the analysis of both vocabularies revealed that Data Cube allows better interoperability with other LODsets and provides more concise encoding of the data. To model the time dimension we selected the Timeline Ontology. We reused its concept of the time interval to encode years in Pordata. Even though the design considerations we made during the data modelling step refer to a particular dataset, Pordata, they can be used as a case study for other statistical datasets.

In the second part of the chapter we considered the technical aspects of the Linked Pordata publication. We started by introducing the publishing pattern that consists of three basic steps: data preparation, RDF data generation and Linked Pordata deployment. We based our proposal on the existing Pordata management infrastructure, that is relational databases. In the data preparation step we explained how we solved the issue of getting data from Pordata in the light of having no access to the Pordata databases. We developed and populated our own database that simulates the Pordata database. We centred our publishing solution around the Virtuoso Universal Server that provides means to develop RDF veiws over the relational data, i.e., the RDF representation is not generated physically, but is constructed dynamically when querying the data. We introduced the Virtuoso RDB-to-RDF language that we used to define mappings from the Pordata relational schema and data into subjects, objects and predicates of corresponding RDF triples. Virtuoso allows to customise the mappings to reuse terms from the existing vocabularies. This was important for our proposal, since

we defined the Pordata Vocabulary in terms of Data Cube and Timeline Ontology. We explained how we developed the Virtuoso mappings to generate the RDF representation of Pordata. In the Linked Pordata deployment step we argued that making HTTP URIs dereferenceable is the basic means to access Linked Data on the Web. Among two alternative ways of accessing Linked Data, via RDF dumps and by querying the data through SPARQL-endpoints, Virtuoso provides the built-n SPARQL-endpoint. We introduced SPARQL, the query language to RDF. The Virtuoso URL-rewriter mechanism solved the issue of making Pordata URIs dereferenceable. Since we do not have physical RDF descriptions of Pordata concepts, we used the rewriter to pre-process requested URIs and form corresponding SPARQL DESCRIBE and CONSTRUCT queries to Linked Pordata. These queries trigger the Virtuoso RDB-to-RDF mappings, which, in turn, generate the RDF representations of the requested URIs, which is sent back to the requester.

5

Linked Pordata use cases

In this chapter we demonstrate a practical usage of the Linked Data version of Pordata. Namely, we illustrate how Pordata statistics can be analysed when put in context. We propose our use cases of contextualising Pordata in Section 5.1. In Section 5.1.1 we discuss the use of vocabulary and identity links that solve the issue of data integration on the Web of Data. Section 5.2 explains how we link Pordata to other LODsets to reuse their information and enrich Pordata statistics. In Section 5.3 we present a demo application that was developed to illustrate the use cases in action. The application extracts data from the LODsets by means of corresponding SPARQL queries. We provide an overview of the existing approaches to query Linked Data and present our solution in Section 5.3.2.

5.1 Use Cases

In Chapter 2 we studied the Pordata project. We determined that statistics are presented to end users as raw data in summary tables. We believe that, published this way, statistics are not of particular interest for non experts. We argued that when put in context statistical data becomes more attractive and understandable for ordinary people. For instance, Pordata aggregators present statistics in context of passing time, e.g., real time counters aggregate data from different statistics, including number of births, deaths or migration balance, and simulate its progressing in time. However, the aggregators work with a small set of pre-selected Pordata data, and we concluded that the current representation of the Pordata data is targeted mainly to expert users.

In Section 3.3.2 we gave examples of how statistics can be put in context and presented in a way that is more comprehensible for ordinary people. Next, we propose our use cases of analysing Pordata statistics in context.

"Presidential" Use Case

> *A journalist is writing a historical article about Portuguese political parties. He is interested in seeing the inflation rate in Portugal in years when presidents of Portugal were from the Socialist Party. He also wants to compare it with the inflation rate in years when the presidents were from the Social Democratic Party.*

"Movie" Use Case

> *The same journalist moved to the "entertainment" department, and now he is writing an article on the Portuguese movie director Manoel de Oliveira's. He wants to see how fortunate the director was when he released his movies. For this, the journalist wants to examine the numbers of screenings of Portuguese movies in years when movies of Manoel de Oliveira had been released. He also wants to compare these numbers with numbers of screenings in years when the director didn't produce movies.*

Currently, we can not accomplish these use cases with Pordata. We can have corresponding statistics in Pordata tables (*National Accounts/Prices/Inflation Rate* and *Culture ans Sports/Cinema/Screenings: total and by film's country of origin* Figure 2.1). However, we do not have enough information to be able to separate years when the president of Portugal was from the Socialist Party from years when he was from the Social Democratic Party. Similarly, the "Movie" use case requires adding to the statistics more information about years when Manoel de Oliveira released his movies. We can obtain the required additional information from other data sources and add it to the Portuguese statistics. For this, the problem of data integration has to be solved. Typically, when one wants to integrate data from different datasets, before agreeing on a common schema, one has to understand the semantics of the datasets. For example, for the "Presidential" use case we need to add extra information about presidents and their parties to the corresponding years. If there is a data source that defines this information, we need to analyse whether the two datasets define the concept of *Year* in the same way to be able to establish proper connections. In Pordata *Year* refers to the Gregorian calendar year, i.e., starts on 1st of January and ends on 31st of December. However, in the target dataset the concept of year could refer to the Julian or Chinese calendar, or it can mean a fiscal or the astronomic year, i.e., a period of time equal to 365(6) days. Next, we discuss how the issue of data integration is addressed on the Web of Data.

5.1.1 RDF links

Integration of data on the Web of Data is made through links that were defined in different datasets. A link takes the form of an RDF triple where the object of a triple is a resource[1]. RDF links can be *internal* or *external*. Internal links connect resources belonging to a single Linked Data set, i.e., the URIs of the subject and object are in the same namespace. For example, the following triple is an internal RDF link:

subject: `http://linkedpordata.org/vocabulary/Number`
predicate: `rdf:type`
object: `http://linkedpordata.org/schema#UnitMeasure`

External links connect different resources, i.e., when the URIs of the subject and object are in different namespaces. For example, the following triple is an external RDF link:

subject: `http://linkedpordata.org/time#Year1991`
predicate: `owl:sameAs`
object: `http://dbpedia.org/resource/Category:1991`

External RDF links are fundamental to Linked Data. They enable semantic connections from data in one dataset to data described in another dataset. The latter may, in turn, have links to data in yet another dataset, and so on. Therefore, setting external RDF links not only connects one dataset to another, but enables the creation of a single globally distributed data space that can be queried and explored as a single RDF graph. There are two types of external links: *Vocabulary Links* and *Identity Links*.

Vocabulary Links The Web of Data takes a twofold approach to deal with integration of data at the schema level. First, data publishers are encouraged to adopt the widely deployed vocabularies to describe their datasets. If the existing vocabularies are not sufficient, publishers should define their own custom vocabulary and make it as self-descriptive as possible. This means that each custom term should have a definition retrievable via the HTTP protocol (in other words, URIs identifying proprietary terms should be dereferenceable). In Section 4.2.3 we discussed how we chose the existing vocabularies to represent Pordata statistics and the time dimensions. We developed the Pordata Vocabulary reusing terms from the Data Cube Vocabulary and the Timeline Ontology. In Section 4.3.3 we addressed the issue of making Pordata URIs dereferenceable. By doing so, we anticipate easier integration of Pordata with other LODsets that employ the same vocabularies, as well as enable consumption of the Pordata data

[1] Recall that the object of a triple can be either a literal or resource (Section 4.2.3)

by Linked Data applications that are familiar with the widely deployed vocabularies, e.g., Linked Data browsers and search engines.

Second, we, as data publishers, consumers of our data and other interested third parties can define mappings between terms of our and other vocabularies. These mappings take the form of RDF links and are called *vocabulary links*. To introduce vocabulary links the RDFS properties can be used, e.g., `rdfs:subclassOf`, `rdfs:subPropertyOf`, and others (Section 4.2.3). For example, in Pordata we have concepts of *Civil Marriage* and *Catholic Marriage*. Assume that in another dataset somebody talks about *Marriage*. We can define the following vocabulary link between our datasets:

subject: `http://linkedpordata.org/vocabulary/Marriage`
predicate: `rdfs:subclassOf`
object: `http://example.com/Marriage`

Adoption of widely deployed vocabularies to model data and vocabulary links between terms of different vocabularies enables easier integration of data on the Web of Data.

Identity Links Identity links address the issue of connecting data at the semantic level. They are used to introduce aliases on the Web of Data. *Aliases* are concepts that are defined in different datasets but have the same meaning. Aliases play an important social function on the Web of Data. They enable different views of the world to be expressed on the Web of Data. Thus, they naturally reflect the real world situation, where different people often have different opinions or some people have more or less knowledge than other people. For example, when you publish statistics about countries and assign URIs to concepts in your dataset, including countries, it is likely that there are many other data providers on the Web who also defined the same countries in their datasets, e.g., DBpedia [39] that contains data extracted from Wikipedia or Geonames [50] that provides descriptions of millions of geographical locations worldwide. Having one, and only one, URI for every concept in the world would entail the creation of a centralized naming authority to assign URIs. However, it is believed by Linked Data advocates that having centralized naming authority would create a major barrier to the growth in the Web of Data. Indeed, for somebody to start publishing data as Linked Data, first, he/she should get to know all the authorized URIs for the things in the dataset. This would take lots of effort and time before the data could actually be published. Defining different URIs for same things in different custom namespaces lowers the barrier to enter the Linked Data space. Later on, a data publisher or third parties may invest efforts into setting identity links between aliases.

By common agreement, Linked Data publishers use the predicate `owl:sameAs` to introduce aliases. An example of the identity link can be the external link presented above, where we stated that the concept of *Year 1991* defined in Pordata is the same as the concept of *Year 1991* defined in DBpedia.

By setting identity links to URIs in another public dataset, a data publisher obtains two main benefits:

- Target URIs are dereferenceable. This means that the descriptions of the concepts identified by these URIs can be retrieved from the Web and reused by the data publisher to enrich his/her dataset.

- These URIs may already be linked to other public datasets. The data publisher can navigate to these datasets and reuse the data, and this might lead to more and more datasets discovered on the Web of Data.

In the next sections we discuss how we implemented the "Presidential" and "Movie" use cases by linking Pordata to the Web of Data.

5.2 Linking Pordata

The Web of Data collectively provides many potential targets for links. There is a wide variety of datasets from different domains available on the Web of Data (Section 3.3.1). While choosing a target dataset for linking we ask ourselves two questions: "How to locate a dataset that can add value to Pordata?" and "How to search for relevant concepts withing the dataset?".

We need to find datasets containing information about Portuguese presidents and Manoel de Oliveira's movies we used the CKAN repository [26] that maintains a list of publicly available LODsets by domain. We found two potential datasets that could provide the additional information we need to implement our use cases: the cross domain DBpedia dataset and the Linked Movie Database. The next step was to explore these datasets and locate the concepts for linking. The following options exist to explore a Linked Data set:

- explore the dataset using the HTML Web forms (HTML Web forms work as Linked Data browsers that display RDF data in a human-readable way);

- if the owner of the dataset does no provide HTML Web forms, Linked Data browsers can be used to explore the data (e.g., Tabulator [111] or Marbles [78] Section 3.3.2);

- Linked Data sets can be directly queried via a SPARQL-endpoint;

5.2.1 "Presidential" Use Case

Recall, in the presidential use case a journalist is analysing statistics about inflation rate in Portugal. These statistics are provided in the *National Accounts/Prices/Inflation Rate* table, that contains data about inflation rate in Portugal from 1960 till 2010. However, there is no indication of years when the country was ruled by the president whose party was *Socialist Party*.

We explored DBpedia [39] with the help of the HTML Web forms and discovered that DBpedia defines concepts of years using the following pattern:

`http://dbpedia.org/resource/Category:{year}`

For example, `http://dbpedia.org/resource/Category:1991` identifies the year `1991`. Next, we will discuss how we retrieved the information defined in DBpedia about this year by posting SPARQL queries to the DBpedia SPARQL-endpoint[2]

Events that happened in Portugal in 1991 can be extracted by the following SPARQL query:

```
SELECT DISTINCT ?event
WHERE {
<http://dbpedia.org/resource/Category:1991> skos:broader ?years .
?yearsByCountry skos:broader ?years .
?yearsInPortugal skos:broader ?yearsByCountry .
?yearInPortugal skos:broader ?yearsInPortugal .
?event dcterms:subject ?yearInPortugal .
FILTER (regex(str(?yearsInPortugal), "Portugal"))
FILTER (regex(str(?yearInPortugal), "1991"))}
```

The result contains different kinds of events. By adding one more filter to the previous query, we limit the events to presidential elections only:

```
FILTER (regex(str(?event), "presidential_election"))
```

In the result of the extended query the variable `?event` is bound to one term that corresponds to the concept of the presidential election in Portugal in Year 1991. The solution looks as follows:

?event
$dbpedia : Portuguese\_presidential\_election, \_1991$

We extended the query with two more triple patterns to retrieve the president that was elected on this election and his party. The following listing contains these patterns:

[2]The DBpedia SPARQL-endpoint is available at `http://dbpedia.org/sparql`.

```
?event <http://dbpedia.org/property/afterElection> ?pres .
?pres <http://dbpedia.org/ontology/party> ?party .
```

The complete SPARQL query that retrieves the president name and the party of the president that was elected in 1991 in Portugal is given in Appendix E. The result of running this query against the DBpedia SPARQL-endpoint is illustrated by Figure 5.1.

eventName	presName	partyName
"Portuguese presidential election, 1991"@en	"Mário Soares"@en	"Socialist Party (Portugal)"@en

Figure 5.1: The Portuguese president elected in 1991 and his party defined in DBpedia.

Links to DBpedia To be able to reuse this information about Portuguese presidents from DBpedia, we defined identity links from the concepts of years in Pordata to the same concepts of years in DBpedia. Figure 5.2 illustrates these links.

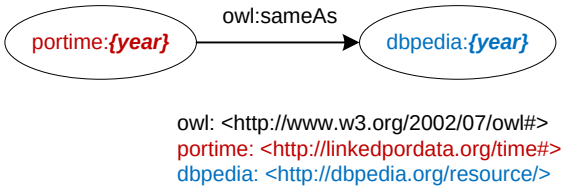


Figure 5.2: Identity links from concepts of years in Pordata to the same concepts of years in DBpedia.

In order to generate the links automatically we defined the following Virtuoso RDB-to-RDF mappings (Section 4.3.2):

```
pdt:year_iri(years.v_year)
                    owl:sameAs pdt:dbpedia_year_iri(years.v_year) .
```

- `pdt:year_iri` IRI class is used to generate URIs for the Pordata years;
- `pdt:dbpedia_year_iri` IRI class is used to generate corresponding DBpedia URIs of years.

The following listing shows an example of the identity link generated by the mapping defined above. It links the same concept of year `1991` from Pordata to DBpedia:

```
<http://linkedpordata.org/time#Year1991> owl:sameAs
                    <http://dbpedia.org/resource/Category:1991> .
```

5.2.2 "Movie" Use Case

Recall, in the "Movie" use case, the journalist is exploring statistics about movie screenings in Portugal. They reside in the *Screenings: total and by film's country of origin* table that contains data about the number of screenings of movies from different countries in Portugal from 1979 till 2010. However, there is not enough information about these years, in particular, we do not know in which years Manoel de Oliveira released his movies.

The *Linked Movie Database (LinkedMDB)* [71] publishes movie-related information. We explored the dataset via the HTML Web forms and discovered that LinkedMDB defines the concepts of countries as follows:

`http://data.linkedmdb.org/resource/country/{country}`

For example, the URI `http://data.linkedmdb.org/resource/country/PT` refers to the concept of the Portugal country.

The following query obtained the titles, directors and date of releases of the movies produced in Portugal:

```
PREFIX movie: <http://data.linkedmdb.org/resource/movie/>
PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT DISTINCT ?movieTitle ?dirName ?date
WHERE {
SERVICE <http://data.linkedmdb.org/sparql>
{
?movieIri movie:country
   <http://data.linkedmdb.org/resource/country/PT> .
?movieIri rdfs:label ?movieTitle .
?movieIri dcterms:date ?date .
OPTIONAL {?movieIri movie:director ?director .
          ?director movie:director_name ?dirName . }}}
```

We used the federated query service provided by Virtuoso [119] to query the LinkedMDB SPARQL-endpoint[3], since the endpoint does not provide a user interface to construct queries. The result of the query is partially[4] illustrated on Figure 5.3.

Among others, we can see the movies produced by Manoel de Oliveira, their titles and the production dates.

While analysing the LinkedMDB dataset, we discovered that there are identity links from the LinkedMDB concepts of countries to the same concepts of countries defined

[3]The LinkedMDB SPARQL-endpoint is available at `http://data.linkedmdb.org/sparql`.
[4]The complete result contains more than 30 movies.

movieTitle	dirName	date
O PÃ¡tio das Cantigas	Ribeirinho	1942-01-23
The Mahabharata	Peter Brook	1989
Ossos	Pedro Costa	1997
No Quarto da Vanda	Pedro Costa	2000
Casa de Lava	Pedro Costa	1995
Colossal Youth	Pedro Costa	2006
Alice	Marco Martins	2005-10-06
Aniki-BÃ³bÃ³	Manoel de Oliveira	1942-12-18
O Convento	Manoel de Oliveira	1995
The Letter	Manoel de Oliveira	1999
I'm Going Home	Manoel de Oliveira	2001
Abraham's Valley	Manoel de Oliveira	1993-10-15
The Forest	Leonel Vieira	2002-11-01

Figure 5.3: Portuguese movies, directors of the movies and the dates of their releases defined in LinkedMDB.

in Geonames [50]. For example, the concept of Portugal is linked to the same concept of Portugal in Geonames through the following link:

```
<http://data.linkedmdb.org/resource/country/PT>
        owl:sameAs <http://sws.geonames.org/2264397/> .
```

`http://sws.geonames.org/2264397/` is the URI identifying Portugal in Geonames. Geonames does not provide a SPARQL-endpoint to directly query the data. However, each URI that is used in the dataset can be dereferenced into the corresponding description. The Virtuoso Sponger functionality [121] allows to dereference URIs within a SPARQL query during the query execution. We can use Sponger to retrieve the RDF description of the Geonames's Portugal as follows:

```
define input:grab-all "yes"
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
SELECT distinct *
WHERE
{
<http://sws.geonames.org/2264397/> geo:long ?long ;
                                   geo:lat ?lat.}
```

The `grab-all` parameter in the query above enables the Sponger mechanism. The query retrieves the latitude and longitude of Portugal from Geonames, `39.5` and `-8` correspondingly.

Links to LinkedMDB

In order to be able to reuse data from both LinkedMDB and Geonames it is enough to link Pordata to LinkedMDB only. We defined the followwing identity links that connect the concepts of countries in Pordata with the same concepts of countries in LinkedMDB:

```
porvocab:Portugal owl:sameAs
                    <http://data.linkedmdb.org/resource/country/PT> .
porvocab:Spain owl:sameAs
                    <http://data.linkedmdb.org/resource/country/ES> .
porvocab:France owl:sameAs
                    <http://data.linkedmdb.org/resource/country/FR> .
porvocab:United_Kingdom owl:sameAs
                    <http://data.linkedmdb.org/resource/country/GB> .
porvocab:USA owl:sameAs
                    <http://data.linkedmdb.org/resource/country/US> .
```

The whole picture of how we exploited the identity links in different datasets is illustrated by Figure 5.4.

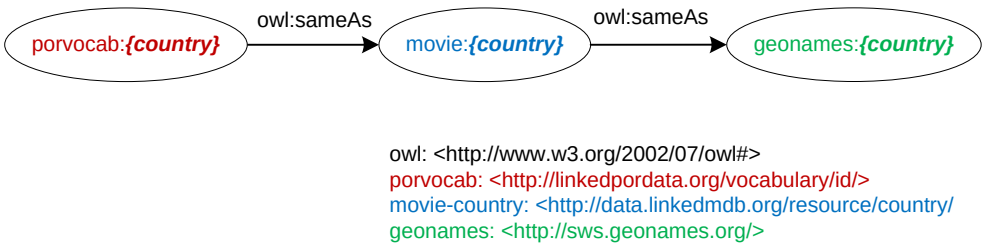


Figure 5.4: Identity links from Pordata to LinkedMDB and from LinkedMDB to Geonames.

5.3 Demo application

As a proof-of-concept, we developed a Web application that provides facilities to analyse Pordata in new ways, including the ones that are defined in the "Presidential" and "Movie" use cases. The demo application is a mashup application that is solely based on results of SPARQL queries evaluated over the Linked Pordata, DBpedia, LinkedMDB and Geonames.

There are different approaches to consume Linked Data from the Web. In the following we provide the overview of them.

5.3.1 Querying Linked Data

In database theory there are two classic approaches to query distributed data: **data warehousing** and **query federation** [55]. Both approaches were adopted to query Linked Data. These traditional approaches face different challenges, we outline them

below. However, they do not take into account that Linked Data queries are posted over the globally distributed data space, and it is not possible to know in advance all data sources that might be relevant for query answering. Hence, to address the full potential of the Web it is necessary to have a novel mechanism to automatically discover and perform "on-the-fly" integration of data on the Web. The **automated link traversal** mechanism was proposed to handle this issue.

Data Warehousing Data warehousing is an approach where data is collected and stored in a central database, called *data warehouse*. Queries are executed over this central database. In Linked Data this idea is realized by setting up a single SPARQL endpoint over a collection of Linked Data sets copied from multiple sources on the Web. Such a collection is built by loading RDF dumps of relevant Linked Data sets into a single RDF store.

One may build his/her own RDF store for data warehousing, or use available public SPARQL endpoints, that provide access to copies of major Linked Data sets. For example, the endpoint by *OpenLink SoftWare* provides an access to copies of the majority of data sets from the LOD cloud[5].

This approach to query the Web of Linked Data offers the best performance. However, setting up such collections of copies of data sets is not a trivial task. As we mentioned before, RDF dumps are an alternative way of serving Linked Data on the Web. Therefore, this is not always the case, that a Linked Data set of interest has an RDF dump for download. Furthermore, this approach is not suitable in situations when having up-to-date data is essential, since updates to the original sources are not immediately reflected in the created data warehouse.

Federated queries Similarly to accessing a single Linked Data set by querying its SPARQL endpoint, applications may access multiple data sets by querying relevant SPARQL endpoints. This approach refers to the *query federation* in the database theory. The function of analyzing a query and decomposing it into several sub-queries is done by a *query federator*. The sub-queries are then executed over the relevant SPARQL endpoints, and the returned intermediate results are joined together to form an answer to the original query.

Query federation approach does not require synchronization of the data, as well as the additional storage space, required by data warehousing. However, in addition to the demand of having a SPARQL-endpoint for each data set of interest, the downside is, that query execution might be very slow. This is especially true when many

[5]The *OpenLink SoftWare* SPARQL-endpoint is available at `http://lod.openlinksw.com/sparql`.

distributed sources are involved.

Automated link traversal The traditional query approaches discussed above implies *a-priori* knowledge and selection of the potentially relevant data sources. By working with certain selected data sources, these approaches ignore the great potential of the Web of Data. In fact, they do not explore all the possibilities of the huge data space created by a large number of interlinked datasets. RDF links take the form of RDF triples (Section 5.1.1), where the subject is a URI in the namespace of one dataset, while the object is a URI in the namespace of another dataset. Thus, the Web of Linked Data can be seen as a single, globally distributed, potentially infinite RDF graph.

New opportunities of querying data also pose new challenges: due to the openness of the Linked Data space it is not possible to know all data sources that might be relevant for answering a query in advance. The *automated link traversal* algorithm, which was proposed to solve this problem [54], is based on the idea to intertwine query evaluation with the dereferencing of the URIs in a query. The algorithm requires a query to contain URIs as a starting point. The query is evaluated over the data retrieved from looking up these URIs. The intermediate results, in turn, may contain URIs referring to other datasets, which may also provide relevant information for answering the query. Thus, these URIs should also be traversed. Data retrieved by dereferencing these URIs is added to the previously retrieved data, and the query is evaluated against the enlarged RDF graph bringing new URIs for traversing.

The link traversal based query execution naturally supports the idea of the Linked Open Data space. Usually, not all the URIs, that are looked up during the query execution, are in the same namespace controlled by a single data publisher. Instead, the Linked Data principles suggest to set links to external data sources. Thus, new data can be discovered and used to answer queries. It is only necessary to define a starting URI in a query to begin exploring the Web of Data. The queried data is up-to-date, as with the federated queries approach. But in contrast to the previously described approaches, the automated link traversal mechanism does not require selecting the data sources in advance to answer a query; nor does it require existence of RDF dumps or SPARQL-endpoints for datasets of interest.

Demo solution We know all the datasets we want to query in advance. DBpedia and LinkedMDB have SPARQL-endpoints. Thus, we will use them to retrieve up to date data. Geonames does not have the SPARQL-endpoint, however, the data can be downloaded as RDF dumps. For our demo application we need to retrieve data about five countries, thus, downloading and setting up large RDF dumps from Geonames is unreasonable. We adopted the link traversal mechanism to query Geonames.

1. Local Virtuoso SPARQL-endpoint for Linked Pordata;
2. DBpedia and LinkedMDB SPARQL-endpoints;
3. Automated link traversal via the Virtuoso Sponger for "SPARQL-less" Geonames.

5.3.2 Demo architecture

The demo application was written in Python using the Django framework[6]. To run SPARQL queries and process the results we used the Python wrapper around a SPARQL service, *SPARQLWrapper*[7]. The visualization was powered by the *SIMILE Exhibit* widgets[8], that along with interactive maps, timelines and timeplots provides advanced filtering functionalities. The Demo system architecture is illustrated in Figure 5.5.

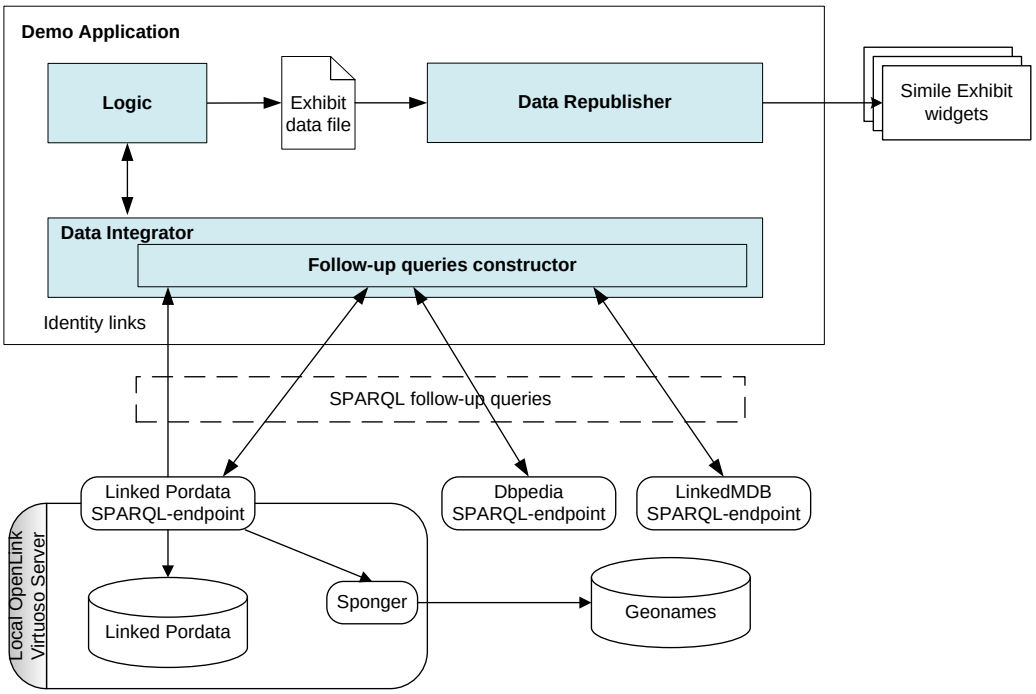


Figure 5.5: Demo system architecture.

The data integrator component is fetching Linked Data. It starts by exploring the identity links in Linked Pordata, then it constructs follow-up SPARQL queries, i.e. substitutes place holders in query templates based on results from previous queries. The results of fetching Linked Data is then processed by the Logic component that builds

[6]https://www.djangoproject.com/
[7]http://sparql-wrapper.sourceforge.net/
[8]http://www.simile-widgets.org/exhibit/

a data file for the Exhibit widgets. The data republisher component calls the SIMILE Exhibit widgets and passes the data file to them.

The demo is available at `http://linkedpordata.dyndns.org/demo/`. The first page presents the available Pordata sets and series for visualization. One has to choose series from the "Inflation Rate" dataset in order to perform the "Presidential" use case or "Screening by country of film" to perform the "Movie" use case.

Let us illustrate the work of the application by example of the "Movie Use Case". Select the series "Country of Origin - Portugal" and hot the *visualize it* button. The visualization page offers three different views: Timeplot, Map and Timeline. For the "Movie" use case we need the first two. By default, the Timeplot represents all the observations of the selected statistical series. The filters on the right side allow us to manipulate the visualisations by selecting specific properties of the data. For example, in the *Movies' filters* we can select the director Manoel de Oliveira, and the Timeplot will represent statistics for years, when Manoel de Oliveira released his movies (see Figure 5.6). By hovering over the Timeplot's points we can see the number of screenings in these years.

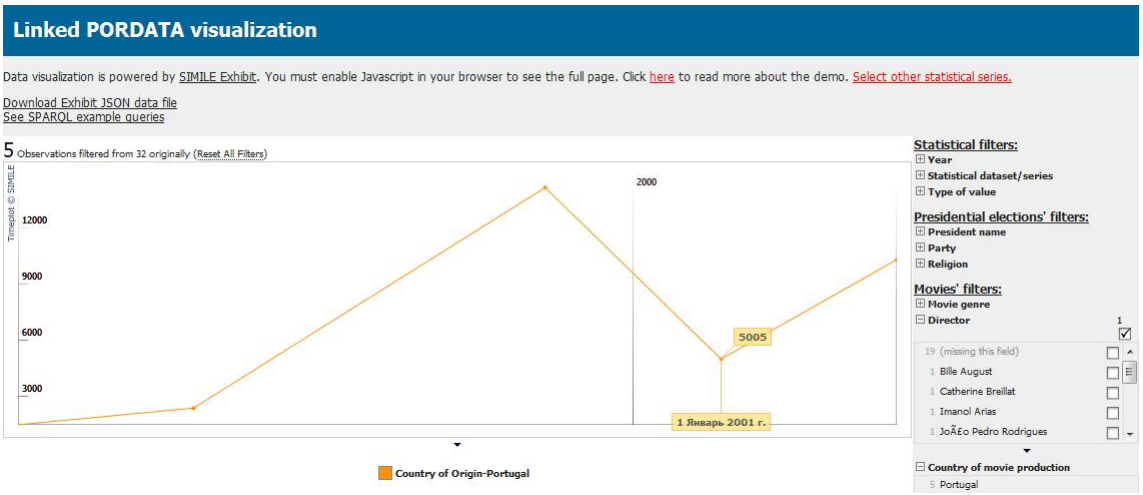


Figure 5.6: Screenings of Portugal movies when Manoel de Oliveira released his movies.

In order to see exactly what movies were released by Manoel de Oliveira, one has to switch to the Map view (see Figure 5.7) and click to the corresponding bubble.

5.4 Chapter Summary

In this chapter we presented the demo application that demonstrates how one can benefit from the Linked Data version of Pordata. We proposed and developed two use

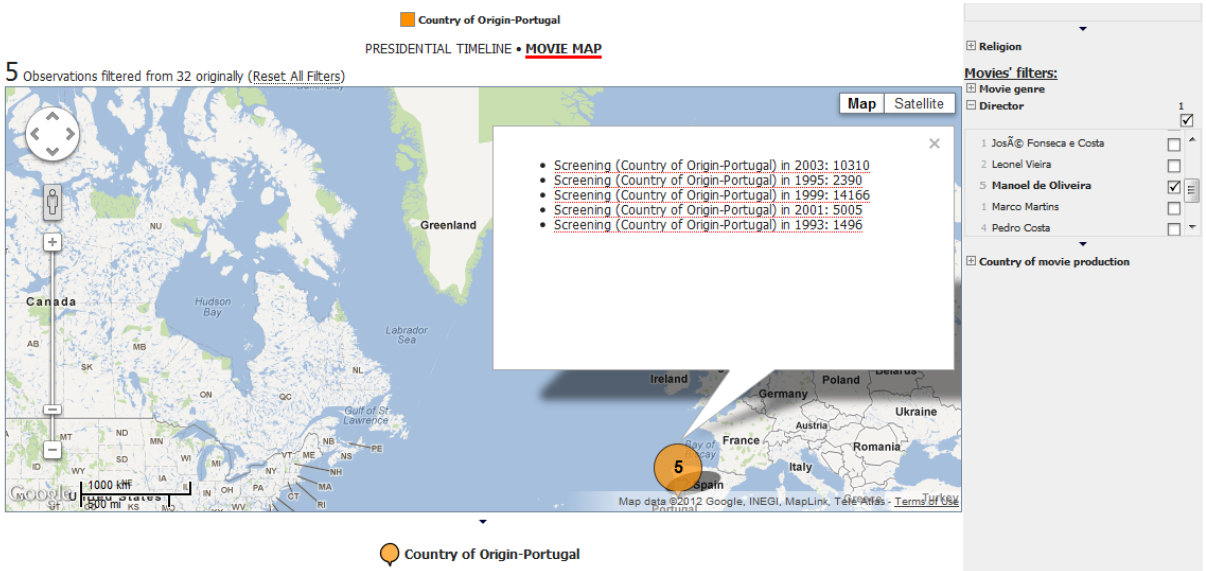


Figure 5.7: What movies were released by Manoel de Oliveira.

cases of analysing the Pordata statistics in context. In the "Presidential" use case the journalist wanted to examine the inflation rate in years when Portugal was ruled by the presidents from the Socialist Party. In the "Movie" use case the same journalist was interested in the number of screenings of Portuguese movies in years when Manoel de Oliveira released his movies. We argued that Pordata data on its own does not allow to accomplish these use cases, for the lack of additional information about specific years. To add this information one has to solve the data integration issue. Linked Data proposes a novel approach to integrate data on the Web. It is based on setting vocabulary links to integrate data on the schema level and identity links to connect the same concepts from different data sources. We explained how we explored DBpedia and Linked Movie Database, the data sources that provide information about Portuguese presidents and movies of Manoel de Oliveira. We demonstrated how we obtained the required information from DBpedia and LinkedMDB by querying them. The analysis of LinkedMDB also revealed that there are identity links from this datasets to Geonames. We followed these links and obtained the Geonames data as well. By doing so, we aimed to show one of the great advantages of the Web of Data of exploring and discovering more datasets while setting links to one of them. Further, we discussed a Web application that was developed on top of the Linked Pordata enriched with the identity links. We talked about the existing approaches to query the Web of Data: data warehousing, federated queries and the automated link traversal mechanism, a novel approach that was introduced to query Linked Open Data. We argued that the link traversal mechanism allows to answer complex queries on the Web of

Data by discovering new data from potentially many datasets that are linked together. One of the main advantage of the link traversal mechanism over the traditional data warehousing and federated queries is that it does not require selecting datasets in advance to answer a query, nor does it require additional work for installing federated systems or downloading and setting RDF dumps. It even does not assume existence of SPARQL-endpoints to query datasets. We demonstrated it in our demo application on the example of Geonames, that does not provide a SPARQL-endpoint. To query LinkedMDB and DBpedia we used their SPARQL-endpoints. The demo application integrates Linked Pordata with the data from DBpedia, LinkedMDB and Geonames and visualises it on a Timeline, Timeplot and Map. The application provides facilities to analyse the Pordata statistics in new ways, including the "Presidential" and "Movie" use cases.

6

Conclusion and Future Work

In this work we introduced the Linked Data principles that define how to publish and interlink structured machine-readable data on the Web. We presented our proposal for publishing statistical dataset Pordata as Linked Data and developed a Web application that demonstrates the benefits of the Linked Data version of Pordata. In Chapter 2 we analysed the Pordata project, that was established by the Francisco Manuel dos Santos Foundation with the aim to promote and disseminate Portuguese statistics to the population. Statistics are currently published in the form of HTML documents, i.e., raw statistical tables embedded into Web pages. We argued that, when presented as raw data, statistics are mainly of interest for expert users and difficult for ordinary people to understand. When put in context, i.e., combined with other data, statistics become more interesting and comprehensible for ordinary people. Pordata can also be exported as `xls` files, that represent the data in structured format so that machines can process the data and extract its meaning by using the columns' and rows' names. However, data in `xls` remains locked in documents. Thus, for example, if there are changes to the original data (e.g., new observations were added for the past year) we need to re-load the corresponding document completely and figure out what was changed. When we are interested in a part of the document, i.e., in a particular observation or in observations that were collected in a certain year, we can not download just that part of the document.

In Chapter 3 we introduced Linked Data, a new data publishing paradigm. The idea of Linked Data was formulated by Tim Berners-Lee as four Linked Data principles [15]. Linked Data extends the scope of the traditional Web from documents to

encompass entities of the real world (e.g., people, buildings and countries) that make up data. To identify the entities, HTTP URIs are used. The Linked Data principles recommend to use the RDF data model to publish and interlink structured machine-readable data on the Web. When links connect data from one dataset to data from another, the links can be used to enrich the original dataset with information from the target dataset. Additionally, one can explore the links in the target dataset and discover more data, and so on. Published and interlinked Linked Data sets on the Web form a global space of semantically rich interlinked machine-readable data, which is also referred to as the Web of Data. We discussed the Linking Open Data (LOD) community project that was created to bootstrap the Web of Data. The enthusiasts of the LOD project are republishing Open Data sets as Linked Data. We introduced major LOD participants from different domains, including main linking hubs on the Web of data such as DBpedia, Geonames and Freebase. We argued that many data providers from different domains became interested in the new publishing paradigm. They seek to gain benefits from exposing their data on the Web of Data, which already contains plenty of interesting information. The fact that all data on the Web of Data is represented using a common data model makes it easier to combine and reuse information from different datasets. Moreover, the semantics of Linked Data is understandable for machines. We presented an overview of new kinds of semantic applications that take advantage of the explicit semantics of data and links on the Web of Data, including applications that were built upon government data.

In Chapter 4 we described our proposal for publishing Pordata as Linked Data. We presented the methodology that we followed in order to develop the proposal that consists of two main steps: data modelling and data publication. In the data modelling step we explained how we extracted the Pordata entities and identified them using of HTTP URIs. We adopted the existing vocabularies - Data Cube and Timeline Ontology, to develop the Pordata Vocabulary. Thus, we ensured better interoperability of Pordata with other Linked Data sets that employed the same vocabularies. Technical realisation of Linked Pordata was based on the Virtuoso Universal Server. To keep the existing Pordata data management infrastructure, we based our proposal on transferring relational data into RDF. We generated RDF representation of Pordata in terms of Data Cube and Timeline Ontology using the Virtuoso RDB-to-RDF mappings. To deploy the Linked Pordata we described how to make Pordata URIs dereferenceable by means of the Virtuoso URL-rewriter. Even though the work we described in our proposal relates to a particular dataset, Pordata, we believe that it can be used as a case study for modelling other statistical datasets as well as publishing relational data as Linked Data.

In Chapter 5 we demonstrated the practical usage of our proposal. We suggested two use cases of analysing Pordata statistics in context: "Presidential" and "Movie' use cases. We argued that these use cases are not supported by the current Pordata data format. We demonstrated how we set identity links between the same concepts of years and countries in Pordata and in DBpedia and LinkedMDB. By using these links we could enrich Pordata statistics with additional information. We also illustrated the main advantage of the Web of Data: when links are set to one dataset on the Web of Data, from that dataset by exploring its links one can discover potentially many other data sources on the Web of Data. While studying the LinkedMDB dataset, we found out that there are links from its concepts of countries to the same concepts of countries in Geonames. By following these links, we reused information from Geonames to enrich Pordata without even linking to it explicitly.

We used this interlinked Pordata version to develop a Web application[1] that integrates data from DBpedia, LinkedMDB and Geonames with Pordata statistics. In order to query the datasets on the Web of Data we employed different approaches. We used federated queries to obtain data from DBpedia and LinkedMDB, as we knew beforehand what datasets we wanted to query and that they had SPARQL-endpoints. The enriched statistics were visualised on Timeplot and Timeline. However, Geonames does not provide a SPARQL-endpoint to directly query its data. To be able to obtain Geonames information we exploited the native Linked Data query mechanism, the automated link traversal. This mechanism allows to query Linked Data on the Web without selecting a priori which dataset we want to query (as it is required in traditional federated queries and data warehousing). We simply took advantage of the basic requirement of any Linked Data set, namely, that each HTTP URIs should be dereferenceable. We dereferenced URIs identifying countries in Geonames and reused the resulting descriptions to add geographical coordinates to the Pordata countries. With this information we plotted statistics about movies on a Map. Our application provides new ways to analyse Pordata statistics, including filtering by political parties, presidents or directors of movies. There is also a possibility to combine different statistical tables, as well as single series, and make comparative analysis of different statistical variables. The application we developed is just an example of how one can take advantage of Pordata published as structured machine-readable data and how to benefit from the information on the Web of Data. One of the main benefits of the Web of Data is that data may be reused in ways unexpected by the original publisher. As soon as the Linked Data version of Pordata is accessible, it is up to other people and their creativity to implement their interesting ideas.

[1] The demo application is available at `http://linkedpordata.dyndns.org/demo/`

6.1 Future work for the proposal

As the best practices suggest, we adopted the existing widely deployed vocabularies to describe Pordata: the Data Cube vocabulary to model statistics and the Timeline Ontology to model the time dimension. By doing so, we increased the potential uptake of the Pordata data by other people or machines, which are most likely familiar with these widely deployed vocabularies. There are several other ways to increase the reusability of the data on the Web of Data. We can consider them as a future work to improve our proposal. They come down to adding various kinds of metadata:

- Provenance metadata.

  The ability to track the origin of the data enables data consumers to determine whether they can trust this data or not. Basic provenance metadata may include, for example, information about the data creator and the creation date. The *Dublin Core* vocabulary [41] provides the required terms to attach this information to the dataset, particularly, the `dc:creator`, `dc:publisher` and `dc:date`. In addition, the information about the Linked Data creation methods can be included as metadata. For this, the *Open Provenance Model Vocabulary* [62] provides means for describing data transformation workflows in terms of `opmv:Agent`, `opmv:Artifact` and `opmv:Process`.

- Dataset-level metadata.

  Technical metadata supports data consumers in choosing the most efficient way to access the data for the specific task they want to perform. The de facto standard for describing Linked Data sets is the *Vocabulary of Interlinked Datasets* [33]. It can be used to define information about such alternative means of data access as RDF dumps and SPARQL-endpoints (via `void:dataDump` and `void:sparqlEndpoint`), logical partition of the dataset (via `void:subset`) and links defined to other LODsets (via `void:Linkset`).

- Licensing metadata.

  The absence of a data licensing statement does not automatically make the data Open. It rather makes unclear for data consumers if they can reuse the data for their purposes or not. If a data publisher wants to publish the data and share it with everybody on the Web he/she must explicitly state this by including a proper data license. In order to include licensing information about the dataset, the Creative Commons licenses [25] can be expressed in RDF. For example,

`cc:license <http://creativecommons.org/licenses/by-sa/3.0/>` attached to a URI identifying a document specifies the CC-BY-SA license for it.

Adopting SDMX terms It is often the case that in a domain of interest, there are already standards and guidelines that define vocabularies, thesauri and code lists to communicate information across different institutions. It is a good practice to reuse common concepts from these sources to describe Linked Data sets. This enables better interoperability between different datasets that support these standards.

Statistical Data and Metadata eXchange (SDMX) is the standard for statistical data exchange. SDMX includes a set of content oriented guidelines (COG)[2], a set of common statistical concepts and associated code lists. There is an ongoing work related to expressing the SDMX concepts in terms of RDF [34].

As a future work for our proposal we can examine the possibility of extending the Pordata Vocabulary with SDMX terms. This will increase interoperability of Pordata with other datasets that already adopted SDMX. Data Cube enables reuse of general statistical concepts via the `qb:concept` property, which links a `qb:ComponentProperty` to the concept it represents. Assuming, the prefix `sdmx-concept` refers to the document that describes COG derived concepts, the following are the examples of reusing the SDMX terms[3]:

- *Temporal dimension* component.

  Currently, we use `porschema:refPeriod` to attach the year to an observation. There is a suitable predefined concept in the SDMX-COG, REF_PERIOD, which indicates the period of time or point in time to which the measured observation is intended to refer. We could extend our vocabulary with the following triple that encodes the meaning of `porschema:refPeriod` through the standardized concept:

  ```
  porschema:refPeriod qb:concept sdmx-concept:refPeriod.
  ```

- *Unit of measure* component.

  We specify the unit of measure of observations via `porschema:hasUnitMeasure`. There is a suitable SDMX concept UNIT_MEASURE, so we could reuse `sdmx-concept:unitMeasure` to define the meaning of our custom unit of measure component:

  ```
  porschema:hasUnitMeasure qb:concept sdmx-concept:unitMeasure
  .
  ```

[2]The guidelines can be downloaded from [103].
[3]The terms are taken from [103].

- *Type of value* component.

  We use `porschema:hasValueType` to clarify the type of an observation's value. There is the SDMX code list CL_OBS_STATUS that provides coded information about the "status" of an observation. This code list defines such statuses as "B-break", "F-forecast" and "P-provisional value". We could reuse the code list to restrict the possible values of the types. Data Cube for this provides the predefined property `qb:codeList`:

```
por:hasValueTypeValue qb:codeList sdmx-code:clObsStatus .
```

6.2 Future work for Linked Pordata

In this work we proposed our idea of how to use the Linked Pordata. We enriched it with information from DBpedia, LinkedMDB and Geonames. The application we built visualises Pordata statistics on a Timeline, Timeplot and Map using temporal and geographical dimensions and provides new ways of analysing Pordata data. For example, inflation rate can be filtered by political parties or presidents. Thus we can analyse inflation rate that was during presidency of one president and compare it with another. Screening statistics were enriched with data about particular movies that were released in corresponding years. This allows to view the statistics and see what movies were released that time, or see the statistics in years when drama movies were screening in Portugal. The application we developed is just an example of how the Linked Pordata can be used. When introduced to the Web of Data Pordata statistics will become accessible to other people who will employ their ideas of how to make use of Pordata. In Section 3.3.2 we highlighted works that were developed based on statistical data that were published to the Web of Data by the U.K. and U.S. governments. They are only few examples of how statistical Linked Data can be used by other people.

Currently, Pordata publishes statistics about Portugal and European countries. These statistics typically do not have the geographical dimension (European countries are presented by one record as *EU27*). However, in the third phase of the project statistics about Portuguese regions will be released. We believe that these statistics can underlie many novel mashup applications. For example, it might be interesting to combine statistics about government spending on education and the number of educational institutions in different regions of Portugal. Represented on a map, such mashup will be easier to analyse and reveal dependencies between the statistical variables and different regions.

The proposal we presented in this work aims at adding a new technical layer to the current Pordata management infrastructure. It will allow to publish Pordata statistics as Linked Data. Adoption of the Linked Data principles to the Pordata statistics will transform them into a structured format with explicitly defined semantics. Moreover, with Linked Data standards, Pordata statistics become a part of the Web of Data, which already provides a big collection of diverse semantically rich data that can be used to enrich Pordata statistics. The common data model (RDF) initialised by all LODsets will ease integration of Pordata with other datasets on the Web of Data and make it easier for others to reuse Pordata data. We believe that, published as Linked Data, Pordata statistics will give rise to many interesting applications and, thus, become more valuable.

Conventional prefix names for the well-known namespaces

prefix	namespace URI	vocabulary
rdf	`http://www.w3.org/1999/02/22-rdf-syntax-ns#`	*RDF core*
rdfs	`http://www.w3.org/2000/01/rdf-schema#`	*RDF Schema*
owl	`http://www.w3.org/2002/07/owl#`	*Web Ontology Language*
skos	`http://www.w3.org/2004/02/skos/core#`	*Simple Knowledge Organization System*
foaf	`http://xmlns.com/foaf/0.1/`	*Friend Of A Friend*
void	`http://rdfs.org/ns/void#`	*Vocabulary of Interlinked Datasets*
dc	`http://purl.org/dc/elements/1.1/`	*Dublin Core*
dcterms	`http://purl.org/dc/terms/`	*Dublin Core terms*
scovo	`http://purl.org/NET/scovo#`	*Statistical Core Vocabulary*
qb	`http://purl.org/linked-data/cube#`	*The Data Cube vocabulary*
tl	`http://purl.org/NET/c4dm/timeline.owl#`	*The Timeline ontology*
xsd	`http://www.w3.org/2001/XMLSchema#`	*XML Schema*
dbpedia	`http://dbpedia.org/resource/`	*Linked Data version of Wikipedia*

In addition there is a handy online service available at `http://prefix.cc` that allows to check already existing widely used prefixes for the vocabulary namespaces.

B

RDF: Structural concepts

An encoding of the Pordata structural concepts: component properties, component classes and component specifications. The syntax used for the encoding is Turtle. Turtle is a plain text human friendly format for representing RDF data. It allows URIs to be abbreviated with prefixes. The W3C submission document [11] gives a detailed introduction to the Turtle syntax.

```
@PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@PREFIX owl: <http://www.w3.org/2002/07/owl#> .
@PREFIX skos: <http://www.w3.org/2004/02/skos/core#> .
@PREFIX qb: <http://purl.org/linked-data/cube#> .
@PREFIX tl: <http://purl.org/NET/c4dm/timeline.owl#> .
@PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> .
@PREFIX porschema: <http://linkedpordata.org/schema#> .
#-------------------------------------------------------------
#-------------------- component classes --------------------
#-------------------------------------------------------------
porschema:UnitMeasure rdf:type rdfs:Class ;
                      rdfs:label "unit of measure" .
porschema:Measure     rdf:type rdfs:Class ;
                      rdfs:label "measure" .
porschema:ValueType   rdf:type rdfs:Class ;
                      rdfs:label "value type" .
#-------------------------------------------------------------
```

```
#------------------- component properties -------------------
#------------------------------------------------------------
porschema:refPeriod rdf:type rdf:Property, qb:DimensionProperty ;
                    rdfs:label "reference period" ;
                    rdfs:range tl:Interval .
porschema:hasMeasure rdf:type rdf:Property, qb:MeasureProperty ;
                     rdfs:label "has measure" ;
                     rdfs:range porschema:Measure .
porschema:hasUnitMeasure rdf:type rdf:Property, qb:AttributeProperty ;
                         rdfs:label "has unit of mesure"@en ;
                         rdfs:range porschema:UnitMeasure .
porschema:hasValueType rdf:type rdf:Property, qb:AttributeProperty ;
                       rdfs:label "has value type"@en ;
                       rdfs:range porschema:ValueType .
#------------------------------------------------------------
#----------------- component specifications -----------------
#------------------------------------------------------------
porschema:CSrefPeriod rdf:type qb:ComponentSpecification ;
     rdfs:label "reference period component specification" ;
     rdfs:comment "Specification of the refPeriod component
         to be mandatory with the attachment to an observation." ;
     qb:dimension porschema:refPeriod ;
     qb:componentAttachment qb:Observation .
porschema:CShasMeasure rdf:type qb:ComponentSpecification ;
     rdfs:label "has measure component specification" ;
     rdfs:comment "Specification of the hasMeasure component
         to be mandatory with the attachment to a data set" ;
     qb:measure porschema:hasMeasure ;
     qb:componentAttachment qb:DataSet .
porschema:CShasUnitMeasure rdf:type qb:ComponentSpecification ;
     rdfs:label "has unit of measure component specification" ;
     rdfs:comment "Specification of the hasUnitMeasure component
         to be mandatory with the attachment to a data set" ;
     qb:attribute porschema:hasUnitMeasure ;
     qb:componentAttachment qb:DataSet .
porschema:CShasValueType rdf:type qb:ComponentSpecification ;
     rdfs:label "has value type component specification" ;
     rdfs:comment "Specification of the hasValueType component
         to be mandatory with the attachment to an observation" ;
     qb:attribute porschema:hasValueType ;
```

```
    qb:componentAttachment qb:Observation .
```

RDF: “Screening by country of origin”

An encoding of the Pordata data concepts for “Screening by country of origin”. The syntax used for the encoding is Turtle. Turtle is a plain text human friendly format for representing RDF data. It allows URIs to be abbreviated with prefixes. The W3C submission document [11] gives a detailed introduction to the Turtle syntax.

```
@PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@PREFIX owl: <http://www.w3.org/2002/07/owl#> .
@PREFIX skos: <http://www.w3.org/2004/02/skos/core#> .
@PREFIX qb: <http://purl.org/linked-data/cube#> .
@PREFIX tl: <http://purl.org/NET/c4dm/timeline.owl#> .
@PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> .
@PREFIX porschema: <http://linkedpordata.org/schema#> .
@PREFIX porvocab: <http://linkedpordata.org/vocabulary/> .
@PREFIX portime: <http://linkedpordata.org/time#> .
#-------------------------------------------------------------
#----------------- custom terms to encode               -----
#----------------- "Screening by country of origin"     -----
#-------------------------------------------------------------
porschema:Country_of_Origin rdf:type rdfs:Class ;
                            rdfs:label "Country of Origin" .
porschema:refCountry_of_Origin rdf:type rdf:Property ;
                    rdf:type qb:DimensionProperty ;
                    rdfs:label "reference country of origin" ;
```

```
                    rdfs:range porschema:Country_of_Origin .
portime:Year1979 rdf:type tl:Interval ;
        rdfs:label "Year 1979" ;
        rdfs:comment "Time interval corresponding to year 1979" ;
        tl:start "1979-01-01" ;
        tl:durationXSD "P1Y"^^xsd:duration ;
        owl:sameAs <http://dbpedia.org/resource/Category:1979> .
porvocab:Portugal rdf:type skos:Concept ;
        rdf:type porvocab:Country_of_Origin ;
        rdfs:label "Portugal" ;
        owl:sameAs <http://data.linkedmdb.org/resource/country/PT> .
porvocab:Normal rdf:type rdfs:Resource ;
                rdf:type porschema:ValueType ;
                rdfs:comment "normal value" .
porvocab:Number rdf:type rdfs:Resource ;
                rdfs:type porschema:UnitMeasure ;
                rdfs:label "Number" .
porvocab:Screening rdf:type rdfs:Resource ;
                   rdf:type porschema:Measure ;
                   rdfs:label "Screening" .
```

D

RDB-to-RDF: IRI and literal classes to map *Tables*

```
prefix pdt: <http://pordata/schemas#>

create iri class pdt:dataset_iri
  using function DB.PORDATA.DATASET_IRI (in id integer not null)
                                                  returns varchar ,
 function DB.PORDATA.DATASET_IRI_INVERSE (in dataset_iri varchar)
                             returns integer option (bijection) .

create iri class pdt:dsd_iri
  using function DB.PORDATA.DSD_IRI (in table_id varchar not null)
                                                    returns varchar ,
  function DB.PORDATA.DSD_IRI_INVERSE (in dsd_iri varchar)
                       returns varchar option (bijection) .

create iri class pdt:measure_iri
  using function DB.PORDATA.MEASURE_IRI
    (in measure varchar not null) returns varchar,
  function DB.PORDATA.MEASURE_IRI_INVERSE
    (in measure_iri varchar) returns varchar option (bijection) .

create iri class pdt:unit_measure_iri
```

```
  using function DB.PORDATA.UNITMEASURE_IRI
    (in unit_measure varchar not null) returns varchar,
  function DB.PORDATA.UNITMEASURE_IRI_INVERSE
    (in unit_measure_iri varchar) returns varchar option (bijection) .

create literal class pdt:dataset_comment
  using function DB.PORDATA.DATASET_COMMENT
    (in id integer not null) returns varchar ,
  function DB.PORDATA.DATASET_COMMENT_INVERSE
    (in dataset_comment varchar) returns integer option (bijection) .
```

E

SPARQL: retrieve president related information from DBpedia.

The complete SPARQL query that retrieves the president name (`?presName`) elected in 1991 in Portugal and the party of the president (`?partyName`) from DBpedia[1]:

```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?eventName ?presName ?partyName
WHERE {
<http://dbpedia.org/resource/Category:1991> skos:broader ?years .
?yearsByCountry skos:broader ?years .
?yearsInPortugal skos:broader ?yearsByCountry .
?yearInPortugal skos:broader ?yearsInPortugal .
?event dcterms:subject ?yearInPortugal .
FILTER (regex(str(?yearsInPortugal), "Portugal"))
FILTER (regex(str(?yearInPortugal), "1991"))
FILTER (regex(str(?event), "presidential_election"))
?event <http://dbpedia.org/property/afterElection> ?pres .
?event rdfs:label ?eventName .
?pres rdfs:label ?presName .
?pres <http://dbpedia.org/ontology/party> ?party .
```

[1]The DBpedia SPARQL-endpoint is available at `http://dbpedia.org/sparql`.

```
?party rdfs:label ?partyName .
FILTER langMatches( lang(?presName), "EN" )
FILTER langMatches( lang(?partyName), "EN" )
FILTER langMatches( lang(?eventName), "EN" )} LIMIT 1
```

Bibliography

[1] Library of congress subject headings info page. http://lcsh.info/ Last-accessed: February 2012.

[2] Soren Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. *The Semantic Web*, 4825(Springer):722–735, 2007.

[3] Australian government data portal. http://data.gov.au/.

[4] Statistics austria. http://www.statistik.at/.

[5] Thomas Bandholtz and Maria Ruther. Linked environment data : Scovo-fying the environment specimen bank. *ESWC 2009*, 2009.

[6] Bbc - music home page. http://www.bbc.co.uk/music Last-accessed: February 2012.

[7] Bbc - programmes home page. http://www.bbc.co.uk/programmes Last-accessed: February 2012.

[8] Bbc nature home page. http://www.bbc.co.uk/nature/wildlife Last-accessed: February 2012.

[9] Christian Becker and Chris Bizer. Flickr wrappr. http://www4.wiwiss.fu-berlin.de/flickrwrappr/ Last-accessed: February 2012.

[10] Christian Becker and Christian Bizer. Dbpedia mobile : A location-enabled linked data browser. *World*, 44(16):6–7, 2008.

[11] D. Beckett and T. Berners-Lee. Turtle - terse rdf triple language - w3c team submission, 2008. http://www.w3.org/TeamSubmission/turtle/.

[12] Francois Belleau, Marc-Alexandre Nolin, Nicole Tourigny, Philippe Rigault, and Jean Morissette. Bio2rdf: towards a mashup to build bioinformatics knowledge systems. *Journal of Biomedical Informatics*, 41(5):706–716, 2008.

[13] M. Bergman. More structure, more terminology and (hopefully) more clarity., July 2007. http://www.mkbergman.com/391/more-structure-more-terminology-and-hopefully-more-clarity/ Last-accessed: February 2011.

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

[15] Tim Berners-Lee. Linked data - design issues, 2006.

[16] Tim Berners-Lee. Putting government data online., 2009. http://www.w3.org/DesignIssues/GovData.html.

[17] Tim Berners-lee, Yuhsin Chen, Lydia Chilton, Dan Connolly, Ruth Dhanaraj, James Hollenbach, Adam Lerer, and David Sheets. Tabulator: Exploring and analyzing linked data on the semantic web. In *In Proceedings of the 3rd International Semantic Web User Interaction Workshop*, 2006.

[18] Ncbo bioportal. http://bioportal.bioontology.org/resources Last-accessed: February 2011.

[19] A. Bizer, C. Jentzsch and R. Cyganiak. State of the lod cloud. http://www4.wiwiss.fu-berlin.de/lodcloud/state/ Last-accessed: February 2011.

[20] Christian Bizer. The emerging web of linked data. *IEEE Intelligent Systems*, 24(5):87–92, 2009.

[21] Christian Bizer and Richard Cyganiak. D2r server - publishing relational databases on the semantic web, 2006.

[22] Christian Bizer, Tom Heath, Kingsley Idehen, and Tim Berners-Lee. *Linked Data: Evolving the Web into a Global Data Space*, volume 1. ACM Press, 2008.

[23] D. Brickley and R.V. Guha. Rdf vocabulary description language 1.0: Rdf schema - w3c recommendation, 2004. http://www.w3.org/TR/rdf-schema/.

[24] Open government data - canda. http://www.data.gc.ca/.

[25] Describing copyright in rdf. http://creativecommons.org/ns Last-accessed: February 2012.

[26] Ckan data portal. http://ckan.org/ Last-accessed: February 2012.

[27] K. G. Clark, L. Feigenbaum, and E. Torres. Sparql protocol for rdf - w3c recommendation, 2008. http://www.w3.org/TR/rdf-sparql-protocol/.

[28] Combined online information system database., December 2011. http://data.gov.uk/dataset/coins Last-accessed: February 2011.

[29] Gianluca Correndo, Alberto Granzotto, Manuel Salvadores, Wendy Hall, and Nigel Shadbolt. A linked data representation of the nomenclature of territorial units for statistics. In *Future Internet Assembly*, December 2010.

[30] Gianluca Correndo, Manuel Salvadores, Ian Millard, and Nigel Shadbolt. Linked timelines: Temporal representation and management in linked data. *Simile*, 2010.

[31] R. Cyganiak and A. Jentzsch. The interactive linking open data cloud diagram webpage., September 2011. http://richard.cyganiak.de/2007/10/lod/lod-datasets_2011-09-19_colored.html Last-accessed: February 2011.

[32] R. Cyganiak and A. Jentzsch. The linking open data cloud diagram webpage., September 2011. http://richard.cyganiak.de/2007/10/lod/ Last-accessed: February 2011.

[33] Alexander K. Cyganiak R., Zhao J. and Hausenblas M. Vocabulary of interlinked datasets (void), March 2011. http://vocab.deri.ie/void#.

[34] Dollin C. Cyganiak R. and Reynolds D. Expressing statistical data in rdf with sdmx-rdf, May 2010. http://publishing-statistical-data.googlecode.com/svn/trunk/specs/src/main/html/index.html.

[35] Reynolds D. Cyganiak R. and Tennison J. The rdf data cube vocabulary., July 2010. http://publishing-statistical-data.googlecode.com/svn/trunk/specs/src/main/html/cube.html.

[36] An official website of the united states government. http://www.data.gov/ Last-accessed: February 2011.

[37] D2r server publishing the dblp bibliography databases start page. http://www4.wiwiss.fu-berlin.de/dblp/ Last-accessed: February 2012.

[38] Faceted wikipedia search. http://dbpedia.neofonie.de/browse/ Last-accessed: February 2011.

[39] Dbpedia community effort webpage. http://dbpedia.org/ Last-accessed: February 26 2012.

[40] Alexander De Leon, Victor Saquicela, Luis. M. Vilches-Blazquez, Boris Villazon-Terrazas, Freddy Priyatna, and Oscar Corcho. Geographical linked data: a spanish use case. In *I-SEMANTICS 6th International Conference on Semantic Systems*, 2010.

[41] The dublin core metadata specification webpage., October 2010. http://dublincore.org/documents/dcmi-terms/ Last-accessed: February 2012.

[42] Orri Erling and Ivan Mikhailov. Mapping relational data to rdf with virtuoso, 2007.

[43] D2r server for eurostat start page. http://www4.wiwiss.fu-berlin.de/eurostat/ Last-accessed: February 2012.

[44] D2r server for the cia factbook start page. http://www4.wiwiss.fu-berlin.de/factbook/ Last-accessed: February 2011.

[45] Falcons search webpage. http://ws.nju.edu.cn/falcons/objectsearch/index.jsp Last-accessed: February 2011.

[46] M. Fernandez-Lopez and A. Gomez-Perez. *Searching for a Time Ontology for Semantic Web Applications*, volume 114, pages 331–341. IOS Press, 2004.

[47] et al Fielding. Part of hypertext transfer protocol – http/1.1: 10 status code definitions. http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html.

[48] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext transfer protocol – http/1.1. request for comments: 2616., June 1999. http://tools.ietf.org/html/rfc2616.

[49] Freebase project's webpage. http://www.freebase.com/ Last-accessed: February 2012.

[50] Geonames project's webpages. http://www.geonames.org/ Last-accessed: February 2012.

[51] Linked data.gov.uk project webpage. http://data.gov.uk/ Last-accessed: February 2011.

[52] Linked data.gov.uk applications. http://data.gov.uk/apps Last-accessed: February 2011.

[53] Wolfgang Halb, Yves Raimond, and Michael Hausenblas. Building linked data for both humans and machines. *WWW 2008 Workshop Linked Data on the Web LDOW2008 Beijing China*, 2007.

[54] Olaf Hartig, Christian Bizer, and Johann-christoph Freytag. Executing sparql queries over the web of linked data. *The Semantic WebISWC 2009*, 5823:293–309, 2009.

[55] Olaf Hartig and Andreas Langegger. A database perspective on consuming linked data on the web. *Group*, 14(2):1–10, 2010.

[56] Michael Hausenblas. Linked data applications. *Access*, pages 1–27, July 2009.

[57] Michael Hausenblas, Danny Ayers, Lee Feigenbaum, Heath Tom, Wolfgang Halb, and Yves Raimond. The statistical core vocabulary (scovo), August 2011. http://vocab.deri.ie/scovo.

[58] T. Heath and E. Motta. Revyu: Linking reviews and ratings into the web of data. *Web Semantics Science Services and Agents on the World Wide Web*, 6(4):266–273, 2008.

[59] Jerry R Hobbs and Feng Pan. An ontology of time for the semantic web. *Acm Transactions On Asian Language Information Processing*, 3(1):66–85, 2004.

[60] Statistics netherlands website. http://www.cbs.nl/.

[61] Tauberer J. The 2000 u.s. census webpage., August 2007. http://www.rdfabout.com/demo/census/ Last-accessed: February 2011.

[62] Zhao J. Open provenance model vocabulary specification - revision: 1.0., October 2010. http://open-biomed.sourceforge.net/opmv/ns.html.

[63] Ian Jacobs and Norman Walsh. Architecture of the world wide web, volume one, January 2004. http://www.w3.org/TR/webarch/.

[64] Anja Jentzsch, Matthias Samwald, and Bo Andersson. Linking open drug data. *Library*, pages 3–6, 2009.

[65] Sheridan John and Tennison Jeni. *Linking UK Government Data*, pages 1–4. ACM Press, 2010.

[66] G. Klyne and J. J. Carroll. Resource description framework (rdf): Concepts and abstract syntax - w3c recommendation, 2004. http://www.w3.org/TR/rdf-concepts/.

[67] The cyc public domain ontology., August 1997. http://www.cs.auckland.ac.nz/compsci367s1c/resources/cyc.pdf Last-accessed: February 2011.

[68] Libris homa page. http://libris.kb.se/ Last-accessed: February 2012.

[69] Linkedbrainz project's webpage. http://linkedbrainz.c4dmpresents.org/ Last-accessed: February 2012.

[70] Linkedgeodata.org. http://linkedgeodata.org/ Last-accessed: February 2012.

[71] Linked movie data base project's webpage. http://linkedmdb.org/ Last-accessed: February 2012.

[72] Linksailor. http://linksailor.com/nav Last-accessed: February 2011.

[73] Ckan - linking open data cloud. http://thedatahub.org/group/lodcloud Last-accessed: February 2011.

[74] Linked open data italia webpage. http://www.linkedopendata.it/ Last-accessed: February 2011.

[75] Linked scotland webpage. http://linkedscotland.org/doc/dataset/geography/sns Last-accessed: February 2011.

[76] Linking open data project webpage. http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData Last-accessed: February 2011.

[77] F. Manola and E. Miller. Rdf primer - w3c recommendation, 2004. http://www.w3c.org/TR/rdf-primer/.

[78] Marbles, the linekd data browser. http://www5.wiwiss.fu-berlin.de/marbles/.

[79] Varish Mulwad, Tim Finin, Zareen Syed, and Anupam Joshi. Using linked data to interpret tables. *Technology*, 2010.

[80] Musicbrainz project's webpage. http://musicbrainz.org/ Last-accessed: February 2012.

[81] Joachim Neubert. Bringing the "thesaurus for economics" on to the web of linked data. *Most*, 25964, 2009.

[82] New zealand government data online. http://data.govt.nz/.

[83] D. Norfolk. Structured data is boring and useless., June 2006. http://www.theregister.co.uk/2006/06/23/unstructured_data/ Last-accessed: February 2011.

[84] New york times - linked open data home page. http://data.nytimes.com/ Last-accessed: February 2012.

[85] Open government movement principles. http://www.opengovdata.org/home/8principles.

[86] Open knowledge foundation webpage. http://okfn.org/ Last-accessed: February 2012.

[87] Open knowledge definition project webpage. http://www.opendefinition.org/ Last-accessed: February 2012.

[88] Open government movement. http://www.opengovdata.org/.

[89] Open library home page. http://openlibrary.org/ Last-accessed: February 2012.

[90] Open definition: Conformant licenses. http://www.opendefinition.org/licenses/ Last-accessed: February 2012.

[91] The open graph protocol webpage., January 2012. http://ogp.me/ Last-accessed: February 2012.

[92] Openstreetmap project's webpage. http://www.openstreetmap.org/ Last-accessed: February 2012.

[93] Linked data for ordnance survey project's webpage. http://data.ordnancesurvey.co.uk/ Last-accessed: February 2012.

[94] E. Prud'hommeaux and A. Seaborne. Sparql query language for rdf - w3c recommendation, 2008. http://www.w3.org/TR/rdf-sparql-query/.

[95] Lewis R. Dereferencing http uris – draft tag finding., May 2007. http://www.w3.org/2001/tag/doc/httpRange-14/2007-05-31/HttpRange-14.html.

[96] Rdb2rdf working group webpage. http://www.w3.org/2001/sw/rdb2rdf/ Last-accessed: February 2011.

[97] Rdf core terms. http://www.w3.org/2000/01/rdf-schema#.

[98] Rdfs core terms. http://www.w3.org/1999/02/22-rdf-syntax-ns#.

[99] Resist home page. http://www.resist-noe.org/ Last-accessed: February 2012.

[100] Rdfizing and interlinking the eurostat data set effort webpage. http://riese.joanneum.at/ Last-accessed: June 2011.

[101] Maria Ruther, Joachim Fock, and Joachim Hubener. Linked environment data. *Group*, pages 1–10, 2010.

[102] Satya S Sahoo, Wolfgang Halb, Sebastian Hellmann, Kingsley Idehen, Ted Thibodeau Jr., Soren Auer, Juan Sequeda, and Ahmed Ezzat. A survey of current approaches for mapping of relational databases to rdf. *w3org*, 2009.

[103] Sdmx initiative webpage. http://sdmx.org/?page_id=11 Last-accessed: February 2012.

[104] Semantic mediawiki project webpage. http://semantic-mediawiki.org/wiki/Semantic_MediaWiki Last-accessed: February 2012.

[105] National statistics institute. http://www.ine.es/.

[106] Claus Stadler, Jens Lehmann, Konrad Hoffner, and Soren Auer. Linkedgeodata : A core for a web of spatial open data. *Framework*, 0(0), 1900.

[107] Thomas Steiner and Michael Hausenblas. How google is using linked data today and vision for tomorrow. *Search*, 700, 2010.

[108] Semantic web education and outreach (sweo) interest group webpage. http://www.w3.org/2001/sw/sweo/ Last-accessed: February 2011.

[109] Swoogle search webpage. http://swoogle.umbc.edu/ Last-accessed: February 2011.

[110] R. Fielding T. Berners-Lee and L. Masinter. Uniform resource identifier (uri): Generic syntax. request for comments: 3986., January 2005. http://tools.ietf.org/html/rfc3986.

[111] The tabulator project webpage. http://www.w3.org/2005/ajar/tab Last-accessed: February 2012.

[112] Talis aspire webpage. http://www.talisaspire.com/ Last-accessed: February 2011.

[113] Olivier Thereaux. Common http implementation problems - w3c note., January 2003. http://www.w3.org/TR/2003/NOTE-chips-20030128/#uri.

[114] Twc linking open government data webpage. http://tw.rpi.edu/web/project/LOGD Last-accessed: February 2011.

[115] Twc logd demos. http://logd.tw.rpi.edu/demos Last-accessed: February 2011.

[116] Uriburner, the linekd data browser. http://uriburner.com/.

[117] Deploying linked data - part 2: Deploying linked data using virtuoso. http://virtuoso.openlinksw.com/whitepapers/VirtDeployingLinkedDataGuide_UsingVirtuoso.html Last-accessed: February 2012.

[118] Pierre-Yves Vandenbussche. Lod vocabularies. http://labs.mondeca.com/dataset/lov/ Last-accessed: February 2012.

[119] Openlink virtoso documentation: 16.2.3. sparql web services & apis. http://docs.openlinksw.com/virtuoso/rdfsparql.html#rdfsparqlprotocolendpoint Last-accessed: February 2012.

[120] Virtuoso open-source wiki main page. http://www.openlinksw.com/wiki/main/Main Last-accessed: February 2012.

[121] Openlink virtoso documentation: 16.9. rdfizer middleware (sponger). http://docs.openlinksw.com/virtuoso/virtuososponger.html Last-accessed: February 2012.

[122] Sweo community project: Lod on the semantic web - vocabularies. http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/CommonVocabularies Last-accessed: February 2011.

[123] Helen K R Williams. Linked data and libraries. *Knowledge Management*, 160(20):1–12, 2010.

[124] Schema central webpage. http://www.schemacentral.com/sc/xsd/t-xsd_duration.html Last-accessed: February 2011.

[125] Raimond Y. and Abdallah S. The timeline ontology., October 2007. http://motools.sourceforge.net/timeline/timeline.html.

[126] Zemanta project's webpages. http://www.zemanta.com/ Last-accessed: February 2012.