1
Introduction

1.1.
Motivation

Statistical data are considered one of the major sources of information and are essential in many fields. In the governmental domain, statistical data offer an “anatomy” of the society and helps identifying the strengths and weaknesses of the government, playing a key role in decision making. In science, statistical data are a fundamental artifact to prove or disprove scientific theories. In the business domain, statistical data about product sales or economic indicators provide crucial input for strategic decisions for management and marketing. However, the elicitation of statistical data is usually quite costly in terms of time and resources, especially in cases involving different organizations (Salas et al., 2012).

Most of the statistical data is produced by official governmental agencies and to ensure quality and accuracy common methodologies, standards and classifications are used in collecting, classifying, processing and publishing statistics. After collecting and processing these data, they are disseminated to the public to allow an overview of the data collected. This audience may be the general public, which usually prefer to display the results in tables and charts, or advanced users, including researchers, analysts and statistical experts, which prefer to get as close as possible to raw data and view data in a format that facilitates digital analysis (Cyganiak et al.,2011).

Statistics are frequently stored in relational databases. The raw data are cleaned and validated and stored in data tables, ensuring the confidentiality of individuals and entities. These data are commonly stored and disseminated as multidimensional structures known as *data cubes* (Cyganiak et al.,2011).

Applications dealing with statistical data usually include Online Analytical Processing (OLAP), a set of tools and algorithms for querying large multidimensional databases (Etcheverry & Vaisman 2012). In OLAP, data are usually perceived as data cubes and the advantage of using such structures is based on the possibility of obtaining different perspectives of the data, transforming the cubes through OLAP operations.

In the process of analyzing statistical data, some characteristics are essential to ensure that data is consumed in a simple but efficient way. The main characteristics are: (i) the data should be published in a simple format, with undue complexity that could become a barrier to their use, and in a standardized way, so that they can be reused and processed by automated tools; (ii) the data should be contextualized with other existing data to enrich the quality of the statistics.

In this context, the Linked Data principles (Heath & Bizer 2011) can be profitably applied to statistical data, in the sense that the principles offer a strategy to provide the missing semantics of the data. Intuitively, if followed, the Linked Data principles will include the data in a context, i.e., will connect statistical data with related data sources, creating a globally interconnected data space that enables a rich analysis of the data (Cyganiak et al. 2011), (Ruback et al. 2013).

To represent data, the Linked Data principles recommend using RDF (Resource Description Framework). This simple and flexible model has features that facilitate data merging, even if the underlying schemas differ, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed (Klyne et al. 2004). The usage of RDF thus allows structured and semi-structured data to be mixed, exposed and shared across different applications, thereby facilitating data interoperability. In fact, RDF inspired interesting mashup tools (Bizer et al. 2007).

The mediation architecture, adopted in this dissertation, helps describing and consuming statistical data, exposed as RDF triples, but stored in relational databases. The architecture features a catalogue of *linked data cube descriptions*, created according to the Linked Data principles. This catalogue will be the subject of study of this dissertation and has a standardized description for each data cube actually stored in each statistical (relational) database known to the mediation environment. The mediator offers an interface to browse the linked data cube descriptions and exports the data cubes as RDF triples, generated on demand from the underlying data sources (Ruback et al. 2013). This mediation architecture follows a pay-as-you-go approach (Bizer 2010), where the conversion of the underlying (relational) data cubes to RDF is performed in real time, as requested by the software agents.

The main motivation for the architecture therefore is to facilitate the consumption of statistical data by software agents in so far as it offers a uniform strategy to describe data cubes and to link their descriptions – especially the dimensions – to other data sources and vocabularies. To make this possible, the architecture features the catalogue of data cubes descriptions, which will be described in detail in this dissertation, containing metadata but not the observations themselves.

Another motivation for developing a uniform strategy to describe data cubes is to facilitate applications that apply mashup operations to data cubes.

1.2. Overview of the Mediation Architecture

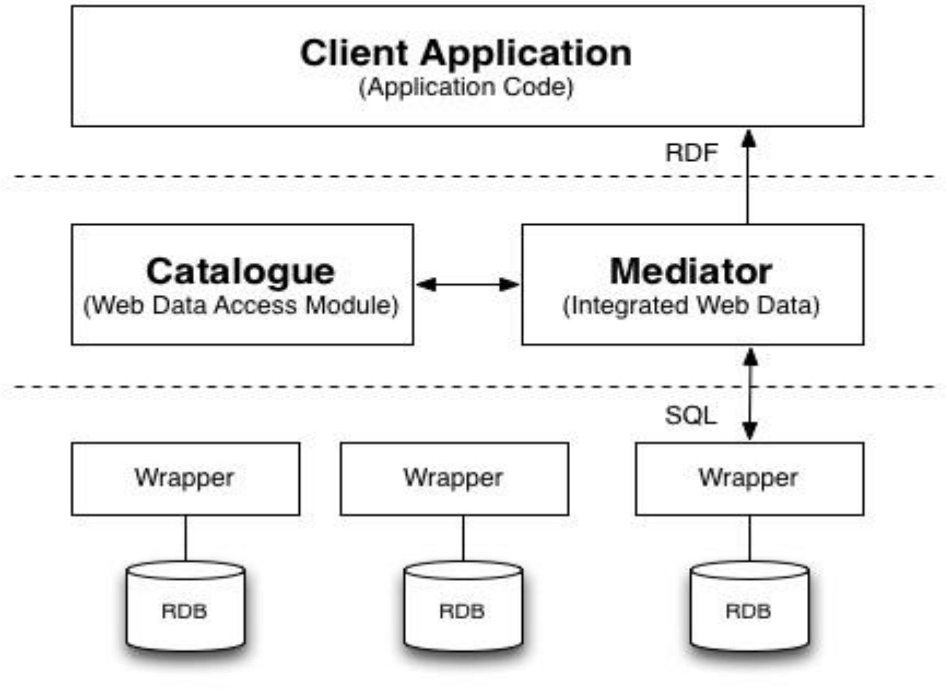

Figure 1 - Overview of the Mediation Architecture (Ruback et al. 2013)

The mediation architecture comprises the following major components: *Wrappers*, *Catalogue*, *Mediator*, and *Client Application*.

A *Wrapper* for an underlying relational database provides star-shaped view schemas describing statistical data stored in the database. It is important to notice that the data cubes may be organized in the underlying database in any way, using several tables. However, the wrapper exposes each data cube through a single star-shaped schema, whose mapping to the underlying tables is internal to the wrapper.

The *Catalogue*, which is going to be studied in this dissertation, contains *public* and *private* data. Public data refers to the data cube descriptions (again, including their dimensions and dimension values) that are exposed to the applications. A data cube description is stored as a set of RDF triples called a *linked data cube description*. A linked data cube description contains triples describing the dimensions and attributes of a data cube, including dimension domain values. However, a linked data cube description does not contain triples that capture the observations, i.e., it is not a complete materialization of a data cube in RDF; the data cube observations still remain in the relational database.

The catalogue also includes public RDF *sameAs* triples that relate resources in linked data cube descriptions with resources located in external data sets, such as DBpedia (Auer et al. 2007).

Private data refers to the information required internally. For each linked data cube description in the catalogue, there is at least one mapping to a star-shaped view schema of an underlying database, which is used to retrieve the observations (of the data cube). Similar mappings are required to retrieve the dimension values.

By assumption, each linked data cube description corresponds to one or more star-shaped schemas, in the sense that the data cube may be redundantly stored in different databases or even in the same database. The mediator is free to choose any one of the star-shaped schemas to materialize the data cube.

The *Mediator* mediates access to the underlying statistical relational databases and exposes catalogue data to the applications. It allows an application to select a linked data cube description, stored in the catalogue, and to apply certain transformations to the cube. It converts the data (i.e., the observations) returned by a wrapper to RDF, passing the triples to the application that submitted the request.

A *Client Application* is any application that interacts with the mediator to access the catalogue and the underlying databases.

One particular client application would be a *Catalogue Browser* that offers a user interface that helps discovering the cubes described in the catalogue.

A client application had already been developed, RdXel (Pesce 2012), which browses the catalogue with the keyword given by the user, requests and displays the observations of the selected cube to the mediator.

1.3. The Three Stages of Data Consumption

The Data Cube consumption goes through three stages: selection, fetching and triplification. This section outlines these three stages.

1. **Stage 1: Selection of a Data Cube**

   This first stage of the process begins with an application sending a *search request* to the mediator.

   The mediator offers two search interfaces for the selection of data cubes. The first interface is a SPARQL endpoint to the catalogue, through which an application may submit SPARQL queries to locate linked data cube descriptions. The second is a keyword search interface that an application may use to submit keywords that are matched against linked data cube descriptions stored in the catalogue. The mediator then returns to the calling application the RDF triples that represent a set of possible cubes to be handled. The application then selects one of these cubes and the description of the cube selected is retrieved.

   After choosing a cube, the application may also decide that it needs the data cube after a certain transformation, such as a slice of the cube (recall that a *slice* of a data cube eliminates a dimension from the cube).

2. **Stage 2: Fetching a Data Cube**

   The second stage starts when the application sends a *fetch request* to the mediator.

   A fetch request specifies a data cube *C* and a transformation *T* on the cube. The mediator queries the catalogue to retrieve information about the database ***d*** where *C* is stored and how it is stored, which is expressed as an SQL query *Q* over the wrapper interface for ***d***. Next, the mediator modifies *Q* to account for *T*.

   The mediator then sends the modified query to the wrapper for ***d*** to fetch the data cube.

   Finally, the wrapper sends the modified query to the underlying database and returns the (relational) query results to the mediator.

3. **Stage 3: Triplification of a Data Cube**

   In the last stage, the mediator triplifies the (relational) data cube, received from the wrapper, and sends the triples to the application.

1.4. Contributions

This dissertation has two major contributions. First, it defines and prototypes a catalogue component covering the following features:

- The catalogue must store the descriptions of the data cubes in RDF.
- The catalogue must store OWL sameAs links to other databases, specially the dimensions of the data cubes.
- The catalog must enable the reuse of the dimensions and their descriptions that are common to different data cubes.
- The catalogue must offer SPARQL queries as well as keyword queries over the data cube descriptions it stores.
- The catalogue must offer ways to upload data cube descriptions (this last feature is not discussed in this dissertation).

The second contribution of this dissertation is to discuss the various problems and design decisions that must be faced when triplifying data cubes stored in relational databases. Briefly, this dissertation emphasizes and illustrates that a linked data cube description must:

- Use standard RDF vocabularies, recommended by W3C, whenever possible.
- Include the mappings from the star schema of the underlying relational databases to the data cube description in RDF.
- Include OWL sameAs statements that link data cube descriptions, specially dimensional data, to external data sources.

In summary, this dissertation contributes to the description and storage of data cubes in RDF – following the Linked Data principles – thereby helping making the semantics of the data cubes explicit.

1.5. Related Work

Related work can be roughly divided into two topics: RDF triplification approaches and statistical data publishing.

Currently, most of the work in the area of triplification focuses on generating RDF from relational database content. There is a wide range of approaches developed in this regard, ranging from very simple scripts, such as Triplify, to standalone solutions, such as D2R , up to integrated tools, such as Virtuoso RDF Views.

Triplify (Auer et al. 2009) is a simplistic approach to convert and publish relational databases to RDF triples and Linked Data. Triplify is based on mapping HTTP-URI requests onto relational database queries. To perform the conversion, a mapping is defined involving SQL queries, using a table-to-class and column-to-predicate approach for transforming SQL query results to RDF. The generation of these semantic representations can be performed on demand or in advance (ETL). Triplify is widely used in the generation of content in RDF and Linked Data.

D2RQ Platform (Bizer et al. 2009a) treats non-RDF Databases as Virtual RDF Graphs. Its aim is to expose RDBs on the Semantic Web to provide access via SPARQL queries and Linked Data. This approach allows relational databases to offer their contents as virtual RDF graphs without having to replicate the whole RDB into RDF triples. It also allows existing RDF vocabularies to be reused. Its mappings use the table-to-class and column-to-predicate approach. The D2RQ platform consists of the D2RQ Mapping Language, D2R Engine and D2R Server. The D2RQ Mapping Language is a declarative language for mapping relational database schemas to RDF vocabularies and OWL ontologies. The D2RQ Engine, a plug-in for the Jena and Sesame Semantic Web toolkits, which uses the mappings to rewrite Jena and Sesame API calls to SQL queries against the database and passes query results up to the higher layers of the frameworks. D2R Server, an HTTP server that can be used to provide a Linked Data view, a HTML view for debugging and a SPARQL Protocol endpoint over the database.

The Virtuoso's RDF Views (Erling & Mikhailov 2009) map relational data into RDF, exposing the relational data as virtual RDF graphs. RDF Views transform the result set of a SQL SELECT statement into a set triple. The mapping follows the table-to-class approach for automatic generation of the mapping file. The mapping file, also called RDF View, is composed by several declarations called “quad map patterns”. A quad map pattern defines one particular transformation from one set of relational columns into RDF triples that match one SPARQL graph pattern. Collectively, these quad map patterns constitute an *RDF meta schema.* Virtuoso´s RDF View allows mapping arbitrary collections of relational tables, into RDF without having to convert the whole data into RDF triples and without physical regeneration of relational data. The mapping file can be stored as triples, and therefore is Available for querying through SPARQL.

One of the few works in the area of transforming statistical data to RDF explores the opposite direction to the approach presented in this dissertation, i.e., the transformation of statistical Linked Data for use in OLAP systems (Kämpgen & Harth 2011).

The *Statistical Data and Metadata eXchange* (SDMX) (Sdmx 2009) is an initiative started in 2001 to foster standards for the exchange of statistical information. The SDMX sponsoring institutions are the Bank for International Settlements, the European Central Bank, Eurostat, the International Monetary Fund (IMF), the Organization for Economic Co-operation and Development (OECD), the United Nations Statistics Division and the World Bank. The SDMX message formats have two basic expressions, SDMX-ML (using XML syntax) and SDMX-EDI (using EDIFACT syntax and based on the GESMES/TS statistical message). Experiences and best practices regarding the publication of statistics on the Web in SDMX have been published by the United Nations (UNSC & ECFE 2000) and the Organization for Economic Cooperation and Development (OECD 2006).

The representation of statistics in RDF started with SCOVO (Hausenblas et al. 2009), (Cyganiak et al. 2010) and continued with its successor, the RDF Data Cube Vocabulary (Cyganiak et al. 2013). The Data Cube Vocabulary is closely aligned with SDMX (Cyganiak et al. 2010). Examples of statistics published as RDF adhering to the Data Cube vocabulary and visualized for human consumption include the EC's INFSO Digital Agenda Scoreboard12 and the LOD2 Open Government Data stakeholder survey (Martin et al. 2011).

OLAP2DataCube (Salas et al. 2012) is an Ontowiki plug-in for statistical data publishing for extracting and publishing statistical data on the Web. The approach here proposed was based on the experience obtained during the development of OLAP2DataCube and aimed at complementing it and is focused on mitigating the problems created by redundantly maintaining both the relational data cubes and their triplifications. Comparing with OLAP2DataCube, data integration is performed in an evolutionary way and data cubes are triplified on demand, so that the problems created by data redundancy are bypassed.

1.6.
Organization

The remainder of this dissertation is structured as follows. Chapter 2 discusses the problem of describing data cubes in RDF. Chapters 3 and 4 respectively detail the proposed architecture of the Catalogue of Linked Data Cube Descriptions and describe its implementation. Finally, Chapter 5 contains the conclusions and directions for future work.