4
The OLAP2DataCube Catalog On Demand Framework

This chapter presents an architecture developed to describe and consume statistical data, which are 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. The main motivation for the architecture is to facilitate the consumption of statistical data by software agents in so far as to offer a uniform strategy to describe data cubes and to link their description – especially dimensions – to other data sources and vocabularies. To make this possible, the architecture features a catalogue of data cubes descriptions, containing the metadata, but not the observations themselves. It follows a pay-as-you-go approach, where the conversion of the underlying (relational) data cubes to RDF is performed in real time, when requested by the software agents (Ruback et al. 2013).

Section 4.1 presents overview of the *OLAP2DataCube Catalog On Demand* framework, introducing its components. Section 4.2 describes the three stages of consuming a data cube. One of the components of this architecture – the mediator – is the focus of this dissertation and is presented in the next chapter.

4.1
Overview of the OLAP2DataCube Catalog On Demand

The architecture of the *OLAP2DataCube Catalog On Demand* is comprised of the following components (see Figure 20): the *Client Application*, the *Linked Data Cube Enriching*, the *Catalogue*, the *Mediator*, *Data Source Recommendation*, the *Data Cube Discovery* and *Wrappers*.

A *Client Application* is any application that interacts with the mediator to access the Catalogue and the underlying databases. A client application has already been developed, RdXel (Neto 2013), divided into two modules. The first module browses the catalogue with the keyword specified by the user, returning

descriptions that match the keywords. The cubes are then listed so that the user can choose between them. The second module allows the user to select the desired dimensions and metrics of a cube. Then, the application requests the observations of the selected cube to the mediator and displays them. The application also offers slicing, dicing, drilling up, drilling down, filtering and ordering operations.

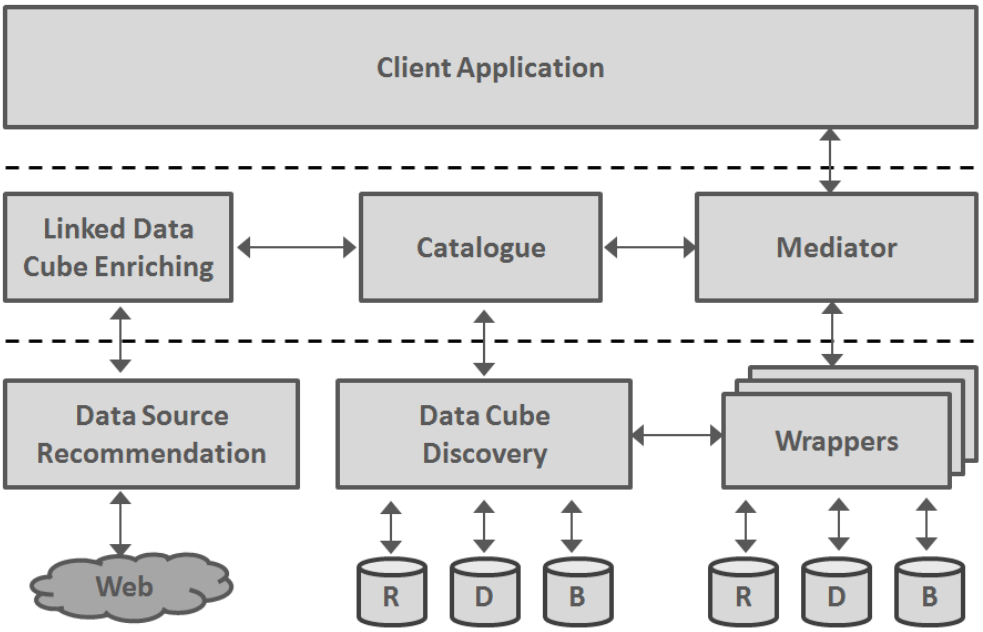

Figure 20: Overview of the OLAP2DataCube Catalog On Demand

A *Wrapper* for an underlying relational database provides star-shaped view schemas describing statistical data stored in the database. 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 (Ruback et al. 2013).

The *Catalogue* contains both *public* and *private* data. Public data refers to the data cube descriptions (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 only contains triples describing the dimensions and attributes of a data cube, sometimes including dimension domain values. Thus, 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 still remains 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 datasets, such as in DBpedia (Auer et al. 2007). For example, if there is a dimension resource representing the City of Rio de Janeiro, there will be a *sameAs* triple relating this resource with the DBpedia entry for the City of Rio de Janeiro, Brazil, i.e., http://dbpedia.org/resource/Rio_de_Janeiro (Ruback et al. 2013).

Private data refers to 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 (Manso 2013).

The *Linked Data Cube Enriching* module aims at enriching Linked Data cube descriptions, interconnecting their components with entities defined by external data sources, according to the Linked Data Principles. It consists of two major components, the automatic enriching component and manual enriching component. The first component automatically generates *owl:sameAs* triples from mappings between local entities and their external sources. The second component allows the user to manually set links between external entities and the entities that the automatic component was unable to process. Together, these components facilitate the definition of data cubes according to the *Linked Data* Principles (Cabrera 2013).

The *Data Source Recommendation* module aims at identifying RDF sources to be used by the Enrichment module (Talavera 2012).

The *Data Cube Discovery* covers the identification of statistical databases stored in a relational database and the synthesis an RDF description for the datasets using the Data Cube Vocabulary, as well as other W3C recommended vocabularies (Ortiga 2013).

The *Mediator* mediates access to the underlying statistical relational databases and exposes catalogue data to the applications. The main role of the mediator is to offer an interface to browse the linked data cube descriptions stored in the Catalogue and to export the data cubes as RDF triples, generated on demand from the underlying databases. This dissertation focuses on the mediator, called *LDC Mediator* (Linked Data Cubes Mediator), which is described in Chapter 5.

4.2

Data cube consumption

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

4.2.1. 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. 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, which are matched against linked data cube descriptions stored in the Catalogue.

The mediator then returns the RDF triples to the calling application that represents a set of possible cubes to be handled. The application then selects one of these cubes. For example, the user selects a cube that contains certain information regarding the employability index for cities in the State of Rio de Janeiro, from 2002 to 2010, by age group. In this case, the data cube contains 3 dimensions - Region, Time and Age Group.

4.2.2. Stage 2: Requesting Data Cube Metadata

The second stage starts when the application sends a request to the mediator for the data cube metadata. The mediator provides a RESTful HTTP method to access the data cube metadata through a RESTful Web service. The method template is:

`GET http://host:port/LDC_Mediator/dataCube/{uriCube}` where `uriCube` is the URI of the requested cube. Section 5.4.3 describes in details this stage of the process.

4.2.3. Stage 3: Triplification of a Data Cube

In the last stage, the mediator triplifies the (relational) data cube, received from the wrapper. It reads the rules in the R2RML mapping files, generates the triples and returns them to the application. Again, the mediator provides a

RESTful HTTP method to return the data cube observations. The method URI is `GET http://host:port/LDC_Mediator/observations/{uriCube}` where `uriCube` is the URI of the requested cube. Section 5.4.4 describes in details this stage of the process.

4.3 Summary

This chapter summarized the *OLAP2DataCube Catalog On Demand* framework, an architecture that facilitates the consumption of linked data cubes. The components of the architecture were also briefly presented: *Client Application*, *Linked Data Cube Enriching*, *Catalogue*, *Mediator*, *Data Source Recommendation*, *Data Cube Discovery* and *Wrappers*. Finally, the process of consuming a data cube was outlined. Understanding the complete architecture helps understand the role of the mediator, which is presented in next chapter.