1
Introduction

The Open Data paradigm is defined as the process by which information is produced, archived and distributed in open raw formats and stored in ways it is accessible and readily available online. The dissemination of Open Data promotes data analysis and allows the reuse of the stored information in new ways, such as the creation of data mashups, i.e., the merging of data from different data sources, in order to produce comparative views of the combined information [Accar & Novak 2009].

Pure HTML sites, database dumps or zipped packages for bulk data downloads are the traditional approaches for publishing Open Data. Often consumers have to use additional tools to separate and extract relevant data from the HTML code or text files, converting it to reusable format, and then mashing the content up with other sources. However, this approach requires a large effort from the data consumer. There are also cases in which data producers provide access to information through specific APIs. This means that consumers have access to data only in the way the producer thinks it should be accessed, i.e., through certain methods and specific formats. The consumer does not have access to the raw data, or to a holistic view of it.

The Semantic Web community provides a common framework that allows data to be shared and reused across applications, enterprises, and community boundaries [Herman 2011]. It offers technologies to describe, model and query data. With the adoption of the Semantic Web standards, organizations are able to publish datasets annotated with domain-specific vocabularies, and offer query interfaces for applications to access public information in a non-predefined way [Accar & Novak 2009]. Particularly, for representing Open Data, W3C recommends the Linked Data standard [Bizer et al. 2007a], which is based on the representation of data in the form of a set of RDF triples [Bizer et al. 2009].

1.1 Problem Setting

The task of representing Open Data as Linked Data requires the conversion of a myriad of information datasets — represented by relational database schemas and their instances — to RDF datasets. A key issue in this process, known as triplification[1], is deciding how to represent relational database schema concepts in terms of RDF classes and properties. This is done by mapping relational database concepts to an RDF vocabulary, to be used as the base for generating the RDF triples. The definition of this vocabulary is extremely important, because the more one reuses well known standards, the easier it will be to interlink the results to other existing dataset [Breitman et al. 2006].

There are triplifying engines that provide support to the mechanical process of transforming relational data to RDF triples, such as Triplify [Auer et al. 2009], D2R Server [Bizer & Seaborne 2004] and Virtuoso RDF View [Erling & Mikhailov 2009]. However, they offer very little support to users during the conceptual modeling stage, producing semantically poor triple sets.

1.2 Goal

In this work, we propose the StdTrip process, which aims at guiding the users in the triplification task, providing support in the stage of creating a conceptual model of the RDF datasets. Based on an *a priori* design approach, StdTrip promotes the reuse of standard — W3C recommended — RDF vocabularies, whenever possible, and suggesting the reuse of vocabularies already adopted by other RDF datasets otherwise.

1.3 Contributions

The main contributions of this thesis are the following:

- The StdTrip process, an approach to guide users in the RDB-to-RDF process, promoting the reuse of standard, RDF vocabularies.
- The StdTrip tool, an implementation of the respective approach that serves to demonstrate the feasibility of the proposed process.

[1] In this thesis we use the terms *triplification* and *RDB-to-RDF* interchangeably, to denote any technique that takes a relational data (schema and tuples) as an input and produces RDF triples.

1.4 Organization

The rest of this thesis is organized as follows. In Chapter 2, we discuss the background concepts involved in the process of interlinking newly produced datasets to existing ones. In Chapter 3, we discuss related work. In Chapter 4, we introduce the StdTrip process to be used in the conceptual modeling stages of the RDB-to-RDF process. In Chapter 5 we describe the StdTrip tool implementation. Finally, in Chapter 6, we discuss some limitations of our approach and the challenges to be met in the future.