3
Related Work

A survey on existing RDB-to-RDF mapping approaches [Sahoo et al. 2009], points out that researchers and practitioners have provided different mechanisms with which to tackle the RDB-to-RDF mapping process. It is important to note, however, that all current RDB-to-RDF approaches listed provide different, proprietary, mapping languages for the mapping process.

Due to this fact, some initiatives are being taken to establish standards by which to govern this process. That is the case of the W3C RDB2RDF Working Group[23] which is currently working on a standard language to express relational database to RDF mappings called R2RML [Das et al. 2010]. A standardized mapping between RDB to RDF may allow the use of a single mapping specification. This feature will allow vendors to compete on functionality and features, rather than forcing database administrators to rewrite their entire relational data to a specific RDF mapping when they want to migrate their data from one database to another.

In what follows we review the RDB-to-RDF approaches that are most relevant to the work presented in this thesis.

3.1 Triplify

Auer et al. [Auer et al. 2009] describe Triplify, a simplified approach based on mapping HTTP-URI requests onto relational database queries. Triplify motivates the need for a simple mapping solution through using SQL as mapping language, for transforming database query results into RDF triples and Linked Data. The mapping is done manually. It uses the table-to-class and column-to-predicate approach for transforming SQL query results to the RDF data model. This transformation process can be performed on demand through HTTP or in advance (ETL). The approach promotes the reuse of mapping files, through a collection of configurations files for common relational schemata. It can be easily integrated and deployed with the numerous widely installed Web

[23] http://www.w3.org/2001/sw/rdb2rdf/

applications such as WordPress, Gallery, and Drupal. Triplify also includes a method for publishing update logs to enable incremental crawling of linked data sources. The approach was tested with 160 GB of geo data from the OpenStreetMap project and exhibited a high flexibility and scalability.

3.2 D2RQ

D2RQ [Bizer & Seaborne 2004] generates the mapping files automatically, using the table-to-class and column-to-predicate approach. D2RQ use a declarative language, implemented as Jena graph [Carroll et al. 2004], to define the mapping file. The approach allows relational databases to offer their contents as virtual RDF graphs without replication of the RDB in RDF triples. The tool can also provide the RDF dump of the relational database if required. In the virtual access the mapping file is largely used for translating SPARQL to SQL queries. The mapping file may be customized by the user, thereby allowing the user to define the reuse of standard ontology terms in the mapping process.

3.3 Virtuoso RDF View

Erling et al. [Erling & Mikhailov 2009] describe the virtuoso RDF View, which uses 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”, which specify how the column values of tables are mapped to RDF triples. Similarity to D2RQ [Bizer & Seaborne 2004], Virtuoso RDF View allows mapping arbitrary collections of relational tables, into RDF without having to convert the whole data into RDF triples. The data returned by the process is presented as virtual RDF graphs without physical creation of RDF datasets, according to the mapping file represented in quad map patterns. It is important to note that, the mapping file can be stored as triples, and therefore queryable via SPARQL.

3.4 DB2OWL

In [Cullot et al. 2007] the authors present the DB2OWL tool, based on mapping a relational database to a single, local ontology. The DB2OWL mapping file uses the XML based language R2O [Barrasa et al. 2004] to describe relationships between database components and a local ontology. This mapping language is used to either execute the transformation in response to a query or in batch mode, to create a RDF dump. The DB2OWL tool uses the table-to-class and column-to-predicate approach with some improvements, the more significant of which is the correct identification of object properties.

3.5 RDBtoOnto

In [Cerbah 2008] Cerbah propose the RDBtoOnto tool which take advantage of both the schema and the data stored in relational databases to obtain more accurate ontologies. The RDBtoOnto is a tool that supports the complete transitioning process from access to the input databases to generation of populated ontologies. The RDBtoOnto uses the table-to-class and column-to-predicate approach to create an initial ontology schema, which is later refined through the identification of taxonomies inside of the RDB data.

3.6 Ultrawrap

Sequeda et al. [Sequeda et al. 2009] present the automatic wrapping system called Ultrawrap, which provides SPARQL querying over relational databases. The Ultrawrap tool defines a triple representation as a SQL view in order to take advantage of the optimization techniques provided by the SQL infrastructure. The ontology, which is the basis for SPARQL queries, is generated following the table to class approach with First Order Logic, introduced by [Tirmizi et al. 2008].

3.7 Automated Mapping Generation for Converting Databases into Linked Data

Polfliet et al. [Polfliet & Ichise 2010] propose a method that automatically associates database elements with ontology entities in the mapping generation process. The method uses schema matching approaches, mostly string-based ones, to align RDB elements with ontology terms. D2RQ [Bizer & Seaborne 2004] is used to creating the initial ontology schema. This approach provides a rudimentary method for linking data with other datasets, based on SPARQL queries and ***rdfs:****label* tags.

Table 1, as follows, provides an overview comparison of the related work described in this chapter and the StdTrip approach, that will be introduced in the next chapters.

	Triplify [Auer et al. 2009]	**D2RQ** [Bizer & Seaborne 2004]	**Virtuoso RDF View** [Erling & Mikhailov 2009]	**DB2OWL** [Cullot et al. 2007]	**RDBtoOnto** [Cerbah 2008]	**Ultrawrap** [Sequeda et al. 2009]	**Polfliet et al.** [Polfliet & Ichise 2010]	**StdTrip**
Mapping implementation (ETL or Virtual)	Both	Both	Both	Virtual*	ETL	Virtual	Both	Both
Query Implementation (SPARQL, SPARQL to SQL)	None	Both	Both	SPARQL to SQL	SPARQL to SQL*	SPARQL to SQL	Both	Both
Mapping Language	SQL	DR2	Quad map patterns	R2O	Rules		D2R	SQL, D2R, R2RML
Support Ontology reuse	No	No	No	No	No	No	Yes	Yes
R2RML Specification	No	No	No	No	No	No	No	Yes
Schema Triples (Output)	No	No	No	Yes	Yes	No	No	Yes
Mapping approach	table-to-class	table-to-class	table-to-class	table-to-class	table-to-class	table-to-class	table-to-class	RDB-ER ER-RDF

Table 1: Comparison of RDB-to-RDF approaches (*)Partial