# PlanetData

**Network of Excellence**

**FP7 – 257641**

# D27.1 Call2:The ETIHQ Repository

**Coordinator: Adrian M.P. Braşoveanu**

**With contributions from: Marta Sabou**

**1st Quality reviewer: Jesus Barrera**

**2nd Quality reviewer: Alexandra Moraru**

| | |
|---|---|
| Deliverable nature: | R/O |
| Dissemination level:<br>(Confidentiality) | PU |
| Contractual delivery date: | M42 |
| Actual delivery date: | M44 |
| Version: | 1.0 |
| Total number of pages: | 28 |
| Keywords: | Linked Data, OBDA, RDB2RDF, RDF Data Cube Vocabulary, statistical linked data, tourism indicators |

## *Abstract*

TourMIS is one of the core sources of European tourism statistics that provides regional (city-level) data about five tourism indicators. This deliverable describes the steps taken to create the ETIHQ repository including:

(i) setting up a technical infrastructure for triplifying and publishing tourism indicators as Linked Data;

(ii) using this infrastructure to expose as Linked Data the content of TourMIS;

(iii) creating links within the ETIHQ dataset as well as between the ETIHQ repository and external data sources.

This deliverable describes the process, tools and methodologies we have followed, and relies on the semantic modelling reported in workpackage D26.

# Executive summary

This deliverable describes the process of publishing the TourMIS database, a core source of European statistics, as Linked Open Data. We report on the publishing methodology and tools, including the following main steps:

- Setting up the infrastructure for publishing TourMIS statistical data as linked data (conversion, triple store, SPARQL endpoint, frontend).
- Using OBDA (Ontology Based Data Access) methodologies to perform RDB2RDF (Relational to RDF) conversion of a SQL Server database into triples.
- Integrating a TDD (Test-Driven Methodology) into all the steps of the Linked Data Publishing in order to make sure that the resulting data is of high quality.
- Applying a simple method of collecting, interlinking and consuming similar data from other repositories.

All data and the interfaces that can be used to explore this data are available at: http://data.etihq.eu/.

Next steps will be to create and promote applications that use this data.

# Document Information

| | | | |
|---|---|---|---|
| **IST Project Number** | FP7 - 257641 | **Acronym** | PlanetData |
| **Full Title** | PlanetData | | |
| **Project URL** | http://www.planet-data.eu/ | | |
| **Document URL** | | | |
| **EU Project Officer** | Leonhard Maqua | | |

| | | | | |
|---|---|---|---|---|
| **Deliverable** | **Number** | D27.1 | **Title** | The ETIHQ Repository |
| **Work Package** | **Number** | WP27 | **Title** | Data Publishing and Linking |

| | | | | |
|---|---|---|---|---|
| **Date of Delivery** | **Contractual** | M44 | **Actual** | M44 |
| **Status** | version 1.0 | | final □ | |
| **Nature** | prototype □ report □ dissemination □ | | | |
| **Dissemination level** | public □ consortium □ | | | |

| | | | | |
|---|---|---|---|---|
| **Authors (Partner)** | | | | |
| **Responsible Author** | **Name** | Adrian Brasoveanu | **E-mail** | adrian.brasoveanu@modul.ac.at |
| | **Partner** | MODUL University | **Phone** | |

| | |
|---|---|
| **Abstract (for dissemination)** | TourMIS is one of the core sources of European tourism statistics that provides regional (city-level) data about five tourism indicators. This deliverable describes the steps we took to create the ETIHQ repository which contains a Linked Data representation of the TourMIS information. |
| **Keywords** | Linked Data, OBDA, RDB2RDF, RDF Data Cube Vocabulary, statistical linked data, tourism indicators |

**Version Log**

| Issue Date | Rev. No. | Author | Change |
|---|---|---|---|
| 03/05/2014 | 1 | Adrian Brasoveanu | Template instantiation. Chapter 2 & 3. |
| 09/05/2014 | 2 | Marta Sabou | Comments on v1 |
| 11/05/2014 | 3 | Adrian Brasoveanu | Reorganized Chapters 1,2,3. |
| 12/05/2014 | 4 | Adrian Brasoveanu | Added Introduction and Conclusions, Chapter 4 |
| 20/05/2014 | 5 | Adrian Brasoveanu | Addressed Reviewers Comments. |
| 22/05/2014 | 6 | Marta Sabou | Final review |
| 23/05/2014 | 7 | Adrian Brasoveanu | Final review |
| | | | |
| | | | |
| | | | |
| | | | |

# Table of Contents

# List of figures and/or list of tables

# Abbreviations

| | |
|---|---|
| ETIHQ | Exposing Tourism Indicators as High Quality Linked Data |
| LD | Linked Data |
| DSD | Data Structure Document |
| QB | RDF Data Cube Vocabulary |
| PROV | Provenance |
| DSL | Domain-specific Language |
| LDP | Linked Data Platform |
| ODBA | Ontology Based Data Access |
| RDB2RDF | Relation Database to RDF |
| TDD | Test-Driven Development (also used as TDE – Test-Driven Evaluation) |
| LDQA | Linked Data Quality Assurance |

# 1 Introduction

In order to create a high quality Linked Data ecosystem, several stages are necessary, each stage requiring a different set of skills, tools and vocabularies. A typical sequence of steps needed to produce high quality Linked Data is:

1. *Data Cleaning* – This step is optional. It is not an integral part of all the Linked Data processes, but it is needed especially for legacy databases.
2. *Semantic Modelling* – Selecting vocabularies, creating ontologies and a structure for the datasets.
3. *RDB2RDF conversion* – Databases are transformed into triples using the process defined in the Semantic Modelling step.
4. *Interlinking* – Linked Data offers simple mechanisms, such aslinks, to extend a dataset with connections towards other linked datasets.
5. *Linked Data Interface Publishing* – Because querying data with regular expressions would not be ideal for all developers, providing SPARQL endpoints and Linked Data front-ends is necessary.
6. *Linked Data Quality Assurance* – Quality Assurance should be a part of the process in all the stages.
7. *Creating applications* – Mashups, visualizations, web applications, apps or any other method to consume data represent a proof that the Linked Data process was important to a certain category of users.

This report focuses on steps 1, 3, 4, 5, and 6 from the previous list, as we applied them when transforming tourism data from the TourMIS database [2] into Linked Data.

We have already described how we applied Semantic Modelling to our tourism data (step 2 above) in a previous report [3].

Creating applications that make use of this dataset (Step 7) will be covered in the following stages of the project.

Following the steps described in the above table, we have created 5 datasets, containing 20 indicators, 2 million observations and 15 million triples. We have also collected 120 indicators from external sources like World Bank, Eurostat.

This deliverable is structured as follows. *Section 2* presents an overview of the tools that can be used in order to successfully publish enterprise databases as Linked Data and describes our selection process for each of the tools we have used. *Section 3* offers a glimpse into the design process and methodologies we have followed when publishing the ETIHQ dataset. *Section 4* is dedicated to a problem that is really important for today's Linked Data: interlinking various datasets, and describes the methodology we have followed in order to select relevant indicators from other Linked Data sources (Eurostat, World Bank). *Section 5* presents the conclusions and future work.

# 2 Linked Data Tools

As mentioned in the Introduction, each stage of the Linked Data process, needs separate tools. Figure 1 explains how these stages fit into our architecture. Between each stage there has to be a process (modelling, enrichment) or the result of a process (URI patterns, DSDs, triples, data formats). Almost all arrows should be bidirectional, but we chose to emphasize only one of them this way, as the process of Semantic Modelling typically requires more effort.

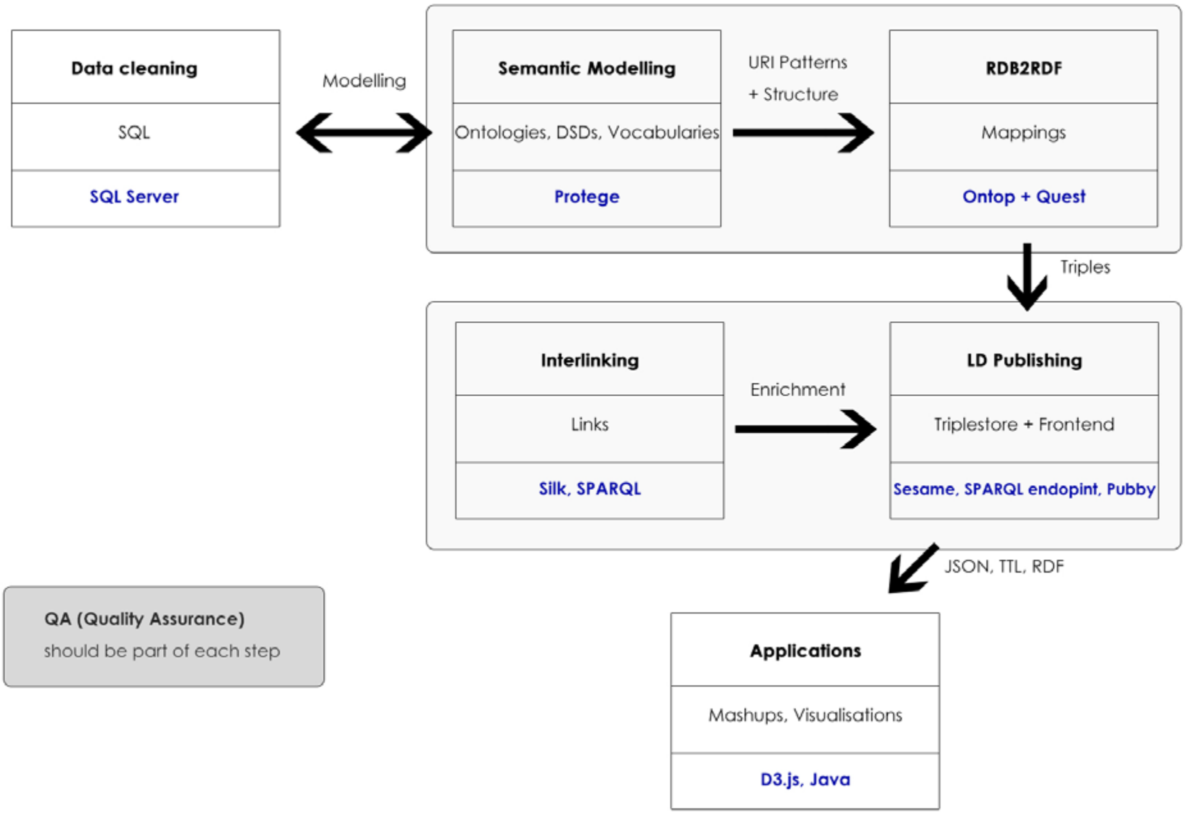


**Figure 1 ETIHQ Architecture**

Defining this architecture, as well as finding the tools was an iterative process. This section explains some of the reasoning behind choosing the tools that appear in the previous figure.

While it was not our intent to make yet another survey of linked data tools, we put significant effort into identifying a suitable tool or tool set to support us in the process of generating linked data from the SQL Server database used by TourMIS. Most of the current products support generation of linked data from open source databases (MySQL, PosgreSQL), but not from enterprise developers (Oracle, Microsoft SQL Server). Some products suffice for generating triples, but have no reasoning capabilities or the possibility to create linked views for the data. Other issues that we faced were the lack of documentation, lack of user groups for exchanging best practices and lack of support for modern standards (Direct Mapping, R2RML, modern vocabularies).

*Data cleaning* is a step that can be performed using either SQL or dedicated tools, therefore it is not strictly a part of the Linked Data process. We found that although OpenRefine is a suitable tool for data cleaning, its scalability is limited and does not support databases that have millions of rows. An ideal tool for data cleaning should work with both databases and triple stores.

*Semantic Modelling* is typically done using ontology editors. Vocabularies are selected, imported into these tools, and sometimes extended. Protégé or TopBraid are typically the tools of the trade. A good overview can be found in [4].

*Data cleaning* and *Semantic Modelling* can be performed in any sequence, with a preference towards performing data cleaning first.

*RDB2RDF conversion* is the actual transformation of the database into triples. It is typically followed by the *Linked Data publishing* step. There are many tools that can be used for these steps, but even so, finding a tool that is suitable for a particular task can be challenging. This is why we considered that some notes on the tools that can be used for these steps are always welcome and therefore included them in the current report.

*Linked Data Quality Assurance* should be included in each stage of the Linked Data generation and publishing process. Unfortunately this is a relatively recent topic, as the first years of Linked Data were rather dedicated to understanding how to produce such data. There are however two tools that are worth mentioning: Sieve [5] and Databugger [6], which also comes with a methodology inspired from TDD [7].

*Interlinking* is a step which can be realized through many options (writing code, using some similarity measures, SPARQL queries, dedicated software, etc). We found that Silk [8] and LIMES can be successfully used for most interlinking tasks.

For *applications*, the tool that needs to be present in a programmer's toolbox is a visualization library. Tables are not enough to understand the insights that are hidden in data.

Taking these considerations into account, we decided to focus our short survey on the tools that could potentially help us for the RDB2RDF conversion and for the Linked Data publishing steps.

## 2.1 RDB2RDF Tools

The most important step towards the publication of relational data as Linked Data is to generate or describe mappings between databases and RDF via special languages or source code. This step is often called triplification, especially if the data to be published as linked data does not come from a database. If the data comes from a database, it is typically called RDB2RDF (Relational Database to RDF).

There are several options for performing triplification:

- mapping generators (scripts or tools that bootstrap the database and the ontology)
- tools that have their own mapping language
- tools developed specifically for converting particular formats (Any23, Excel to LOD converters)
- tools that support RDB2RDF: Direct Mapping [11] [12] and R2RML [10][13]
- writing code (typically in Java, Python, PHP or Ruby).

Some products will also provide all these options in order to help generate quality linked data (D2RQ, Virtuoso, Ontop).

For RDB2RDF tools, we were interested if they:

- Offer support for both Direct Mapping and R2RML;
- Work well with both open source and enterprise databases;
- Offer multiple routes for triplification, reasoning or publishing the data;
- Offer a fast reasoner;
- And are updated frequently.

Previous surveys ([9] and [14]) lack mentioning the practical things that developers should know about a conversion tool when creating Linked Data such as supported databases or reasoning. We therefore believe that our criteria are also important for those who want to use this class of products.

One of the surveys [9] dedicates a lot of space to the R2RML tools. It also introduces a simple classification of RDB2RDF tools into 2 large classes of software packages:

- Non R2RML (a good review of these can be found in a Git[1])
- R2RML enabled

We were mostly interested in the second class of tools, as they seem to be more mature. The following table can be seen as a continuation of the survey from [9] by complementing it with previously missing information.

**Table 1 RDB2RDF conversion tools that use R2RML and Direct Mapping**

| **Product /**<br>**(Creator) /**<br>**URL** | **Supported Databases** | **RDB2RDF Transformation** | **Linked Data Materialization, Publishing, UI** | **Reasoning** |
|---|---|---|---|---|
| D2R Server<br>(DERI)<br>http://d2rq.org/ | MySQL, Oracle, SQL Server, PostgreSQL,, HSQLDB, Interbase | R2RML (Experimental),<br>Direct Mapping,<br>D2RQ Language | Yes (Velocity Templates) | No |
| Virtuoso<br>(OpenLink Software)<br>http://virtuoso.openlinksw.com/ | MySQL, Oracle, DB2, SQL Server, PostgreSQL , HSQLDB, Interbase<br>Any SQL-92, SQL-99, ODBC, JDBC | R2RML,<br>Meta Schema Mapping Language,<br>SPARQL 2 SQL Query Rewriting | Yes (Faceted Views, Linked Data Views) | RDFS,<br>OWL Inference |
| Ontop<br>(Free University of Bozen - Bolzano) http://ontop.inf.unibz.it/ | MySQL, PostgreSQL, H2, DB2 Oracle , SQL Server, Teiid | R2RML,<br>Direct Mapping,<br>SPARQL 2 SQL Query Rewriting | Materialization<br>UI via 3rd party tools | Quest engine- RDFS, OWL 2 QL |
| Morph (UPM)<br>https://github.com/fpriyatna/morph | PostgreSQL, MySQL, Monet | R2RML,<br>SPARQL 2 SQL Query Rewriting | No | No |
| SparqlMap<br>(AKSW Leipzig)<br>http://aksw.org/Projects/SparqlMap.html | PostgreSQL, MySQL, HSQL | R2RML<br>SPARQL 2 SQL | No | No |
| Information Workbench<br>(Fluid Operations)<br>http://www.fluidops.com/information-workbench/ | Oracle, DB2, SQL Server, MySQL, PostgreSQL, CSV<br>other sources | R2RML<br>XML2RDF | Yes<br>(Faceted Views, Linked Data Views, Visualizations) | N/A |

[1] https://github.com/timrdf/csv2rdf4lod-automation/wiki/Alternative-Tabular-to-RDF-converters

| | | | | |
|---|---|---|---|---|
| Talend Open Studio<br>(Talend)<br>http://www.talend.com/download | Oracle, DB2, SQL Server, MySQL, PostgreSQL, CSV<br>other sources | R2RML plug-ins via TalendForge (AllegroGraph, Talend4SW) | Plug-ins | No |
| Oracle 12c<br>(Oracle)<br>http://www.oracle.com | Oracle<br>Other sources | R2RML,<br>Direct Mapping | No | OWLSIF,<br>OWLPRIME, RDFS++<br>3rd party (Pellet) |
| Ultrawrap<br>(Capsenta)<br>http://capsenta.com/ultrawrap/ | PostgreSQL, DB2, SQL Server, Oracle | R2RML,<br>Direct Mapping | Yes (via D2R) | N/A |
| Spyder<br>(Revelityx)<br>http://www.revelytix.com/content/spyder | Oracle, SQL Server, PostgreSQL, MySQL, DB2, CSV | R2RML,<br>Direct Mapping | No | No |
| db2triples<br>(Antidot)<br>https://github.com/antidot/db2triples/ | PostgreSQL, MySQL | R2RML<br>Direct Mapping | No | No |
| XSPARQL<br>(DERI)<br>http://xsparql.deri.org/rdb2rdf | PostgreSQL, MySQL | R2RML<br>Direct Mapping | No | No |
| RBA-R2RML<br>(Federal University of Ceara, CE, Brazil & PCU, Brazil)<br>Not public | MySQL, Oracle, SQL Server, PostgreSQL, HSQLDB, Interbase | R2RML<br>Direct Mapping | No | No |
| R2RML Parser<br>(Nick Konstantinou)<br>https://github.com/nkons/r2rml-parser | PostgreSQL, MySQL | R2RML | No<br>(Except if used with D2RQ) | No |

We only focused on the tools that support RDB2RDF standards. Only one of the reviewed tools was not available online at the time of writing (RBA-R2RML), but we included it for reference, as their approach seems to be promising.

Besides the tools reviewed here, the only one that caught our attention was **OpenRefine**[2]. It is not an R2RML tool, but it provides extensions such as:

- RDF Extension - for publishing datasets with the QB vocabulary;
- Named Entity Extension - for highlighting named entities;
- DBpedia Extension - for linking with DBpedia.

While Refine is a good tool, it is also challenging to use for large datasets (the cleaning step, for example, would require you to look at millions of columns).

Almost all of the reviewed products support PostgreSQL and MySQL, the free databases with the largest user bases. They all offer a SPARQL endpoint, either directly or via a plug-in. The huge number of tools suggest that this trend of using R2RML and Direct Mapping together with some custom mapping languages (either open-source or proprietary) for RDB2RDF conversions will continue. No product has a detailed documentation, but D2RQ, Virtuoso, Ontop have useful guides and tutorials to get started. We found that particularly useful are the products that offer at least some basic scaffolding or data wizards so that we get a feeling for how the linked data will look like: D2RQ, Virtuoso, Ontop, Ultrawrap, etc, even if in most cases these assistants are for their own mapping languages.

Enterprise solutions such as Talend typically offer performant GUIs for data integration, and can connect to lots of data sources: almost any database (including SQL Server), Office, CSV, text, HTML, and many others. Their cost is, however, prohibitively expenssive for small research projects.

While there are many solutions out there that allow publishing Relational Data as Linked Data, only 3 of them are free, open-source and support SQL Server:

- D2RQ
- Virtuoso
- Ontop

**D2RQ** connects to SQL Server, but unfortunately it still has no official release with support for R2RML (just a preview release from 2012). Current D2RQ solutions that use R2RML use either the 2012 preview or their custom forks of the GitHub code. It supports both Transient and Persistent Views.

**Virtuoso** also has a Linked Data interface for publishing Linked Data and connectors to almost all important databases. It offered RDB2RDF conversions even before R2RML became a de-facto standard. It supports federation and publishing billion triples datasets such as DBpedia. The documentation is scarce, and it has a steep learning curve. That would have been however minor inconveniences, considering what it offers. Considering the fact that the R2RML support in Virtuoso fails to pass a lot of the test cases[3], it is much easier to recommend Virtuoso as a triple store or as a Linked Data publishing solution, than as a RDB2RDF converter that uses R2RML. It supports both Transient and Persistent Views. Another problem for us was that the RDB2RDF editor is not available in the free version of the tool.

**Ontop** supports both Virtual RDF Graphs and triples materialization. It also comes with a bundled Sesame which is custom-made for accessing Virtual RDF Graphs and which can be used in conjunction with Pubby or Elda to publish the data. If the SPARQL 2 SQL query rewriting (Transient Views) is too slow, there is always the option of just materializing the triples and storing them into any triple store (Persistent Views). One of the interesting features is federation from multiple databases via Teiid. While we do not need this feature, it can offer an interesting method for performing alignment between large databases. The Protege plug-in offers a good mapping assistant. It also offers a fast reasoner (Quest), which supports powerful inference mechanisms for RDFS, OWL 2 QL. It has the fastest query processor available today (10x-500x times faster than other engines)[4]. Based on the above considerations, we decided to use Ontop, as it was one of the best solutions available. The reasoning behind our choice is explained in the next chapter (Section 3.3).

[2] http://openrefine.org/

[3] http://www.w3.org/TR/rdb2rdf-implementations/

[4] http://ontop.inf.unibz.it/?page_id=74

## 2.2 Linked Data Publishing Tools

Exposing data as triples or offering a SPARQL endpoint is an important part of the Linked Data publishing process.

When reviewing these tools, we took into consideration the following criteria:

- The support for various SPARQL and data formats;
- Ease of use;
- Functional interface;
- Update frequency.

We considered two large categories:

- SPARQL GUIs (the equivalent of command line tools for Linked Data)
- Linked Data Frontends (the equivalent of traditional database interfaces for Linked Data).

### 2.2.1 SPARQL GUIs

A SPARQL endpoint is the minimal interface that is required for Linked Data exploration, but it is typically aimed at advanced users with working knowledge of SPARQL. It is the equivalent of the SQL browsers offered by most of the databases today. It is however hard to keep it online sometimes, and even maintainers of large datasets like PubMed or Bio2RDF have problems with making them available at all times[5].

A SPARQL endpoint is often featured with Relational to RDF converters (D2RQ, Virtuoso, etc), with Linked Data Platforms, or with triple stores (Virtuoso, Sesame, etc). In many cases it is however better to install a separate interface for the SPARQL endpoint, especially if it offers more advanced features such as SPARQL highlighting, saved queries (query examples), autocompletion, debugging capabilities.

**Table 2 SPARQL GUIs**

| **Product / (Creator) / URL** | **SPARQL** | **Formats** | **Notes** |
|---|---|---|---|
| SNORQL (Richard Cyganiak) https://github.com/kurtjx/SNORQL | SPARQL 1.0<br>Not clear if it supports SPARQL 1.1 and Extensions | RDF/XML, JSON, TXT, Turtle, N-Triples, HTML | Old but effective GUI |
| Flint (TSO OpenUp Labs) https://github.com/TSO-Openup/FlintSparqlEditor | SPARQL 1.0, 1.1, Extensions | RDF/XML, JSON, TXT, Turtle, N-Triples, HTML | Advanced GUI |
| YASGUI (Laurens Rietveld) http://laurensrietveld.nl/yasgui/ | SPARQL 1.0, 1.1, Extensions | RDF/XML, JSON, TXT, Turtle, N-Triples, HTML | Advanced GUI based on Flint |

We only considered the tools that are currently updated. Other tools that were not included because we have not been able to find clear and updated information regarding their status are Twinkle, Glint and SPARQLer.

All of the tools included in the table are ready for production and are worth using. SNORQL has been around for a few years and was tested in large projects such as DBpedia, therefore it is really a good option for enterprise use cases. Flint and YASGUI [15] represent the next generation of SPARQL tools. While they are not yet widely deployed, they offer significant improvements in usability that could help both new and experienced SPARQL users.

[5] http://daverog.wordpress.com/2013/06/04/the-enduring-myth-of-the-sparql-endpoint/

### 2.2.2 Linked Data Frontends

Linked Data Frontends that follow Linked Data principles (URI design, redirects, etc) are typically presented together with the triple stores. They offer a direct way of browsing the resources and ontologies as HTML pages. Without such frontends, large datasets would be hard to navigate for most users. Some frontends are extensible through a system of plug-ins.

**Table 3 Linked Data Frontends**

| **Product / (Creator) / URL** | **Linked Views (Browser)** | **LD API** | **Read/ Write** | **Data Formats** | **Notes** |
|---|---|---|---|---|---|
| Pubby (Richard Cyganiak) https://github.com/cygri/pubby | Yes | No | No | Turtle, RDF/XML | Still the most popular |
| D2RServer (Richard Cyganiak) http://d2rq.org/d2r-server | Yes | No | No | Turtle, RDF/XML, JSON | Popular in OGD circles |
| ELDA(Epimorphics)<br>https://github.com/epimorphics/elda | Yes | Yes | Yes | Turtle, RDF/XML, JSON, CSV, Text, HTML | Popular in OGD circles in UK, Australia |
| Virtuoso (OpenLink Software)<br>http://virtuoso.openlinksw.com/ | Yes | Yes | Yes | Turtle, RDF/XML, JSON, CSV, Text | Useful for large datasets |
| Graphity (Graphity) https://github.com/Graphity/ | Yes | Yes | Yes | Turtle, RDF/XML, JSON | Components are split into multiple products |
| Node LDP (AKSW)<br>https://github.com/AKSW/node_ldp | Yes | Yes | Yes | Turtle, RDF/XML, JSON | Works only with Node.js in JavaScript |
| Callimachus (3 Round Stones)<br>http://callimachusproject.org/ | Yes | Yes | Yes | Turtle, RDF/XML, JSON | Typically used as LD CMS |
| Marmotta (Apache)<br>http://marmotta.apache.org/ | Yes | Yes | Yes | Turtle, RDF/XML, JSON | Offers transactions, versioning and rule-based reasoning |

While for historical purposes it is good to also list software that is not actively developed, we decided to list only the publishing tools that we considered suitable for today's needs, and that are still under development.

Node LDP is a good solution for providing Linked Data to JavaScript applications, and Callimachus is great as a CMS that makes use of Linked Data.

Virtuoso and Apache Marmotta are all-in-one solutions. They offer all the functionalities needed for supporting Linked Data publishing: frontend, views, versioning, reasoning. They are suitable for projects where a single product should be used for implementing the whole Linked Data Lifecycle (as described in this deliverable, or following other methodologies), but they are not necessarily the best choice for providing Linked Data frontends.

While Pubby is old, it is effective and widely deployed. It also comes with some plug-ins for metadata extension, maps, and a richer interface, even though some of these features might not yet be merged into its master branch. D2RQ frontend is similar to Pubby.

ELDA has a considerable learning curve, but offers a much better user interface than Pubby in the end. It is also one of the first examples of a Linked Data Platform. ELDA is a good choice for teams of developers who create Linked Data APIs. Graphity is a recent project which, similarly to ELDA, proposes another model of Linked Data Platform [16].

For our first implementation we used Pubby because of its ease of use. Later versions will switch to ELDA.

# 3 Publishing Tourism Statistics as Linked Data

A subset of the TourMIS system was published as Linked Data several years ago. It contained approximately one million statements and focused only only on a few types of indicators. The triples were produced by writing Java code using the Jena library [1]. As part of ETIHQ, our goal is to publish the entire content of TourMIS in terms of the RDF Data Cube Vocabulary [17]. To that end, we examined current best practices [18] [19], and created the methodology described in our previous deliverable, D26.1, in order to publish tourism data from TourMIS [2] as linked data.

This chapter describes all the steps we took in order to publish the new datasets and make sure that they will comply with the recent developments in the industry.

It is worth mentioning that in addition to the methodology that we followed for creating Semantic Models, we have also incorporated two methodologies that ensure creating high quality linked data:

- OBDA – Ontology Based Data Access [21].
- TDD – Test-Driven Development (our SPARQL queries are not yet formalized as those from [7], but we follow their guidelines).

While these methodologies are not necessarily new (first ideas about OBDA can be found in the mid '90s [20], for example), they have only recently started to be applied for Semantic Web developments.

## 3.1 Cleaning TourMIS

In order to expose the TourMIS database as Linked Data we have first performed a refactoring process, including the following steps:

- We removed tables and columns that were not needed. The database is already more than a decade old, therefore it had some cases of redundant or unnecessary data. For smaller datasets, this step can be done with OpenRefine, but we used SQL instead to cater for the complexity of this operation.
- We checked the links between various tables in the database schema to make sure the deletion of tables or columns did not affect them. This was particularly important as some of these links between tables are also reflected in the Linked Data structure.
- We changed some keys in order to have clean URIs later.
- We removed duplicates, whenever possible, as they are not needed in Linked Data. This was particularly important because of the fact that in SPARQL (as it is currently implement in most triple stores), we have no guarantees that there won't be duplicates, even if we use the DISTINCT clause.
- We checked the various applications that use the database to see if they still work (the web application that is used for inserting data, for example).

This step is important because errors will cost much more to fix at a later stage. It is also important for maintaining high quality datasets.

## 3.2 Semantic Modelling

This step was already described in our previous deliverable, D26.1 [3]. We have done the following:

- Selected vocabularies (QB, PROV-O, DCT, Reference interval service, etc).
- Built ontologies (Points of Interests, Shopping, Base).
- Created the basic URI structure.
- Created DSD for the various datasets.

## 3.3 RDB2RDF

We decided to follow an **OBDA** (Ontology Based Data Access) methodology as implemented in Ontop, a plug-in for Protégé.

### 3.3.1 OBDA Methodology

Some of the advantages of OBDA over other triplification methodologies are:

- **OBDA** (Ontology Based Data Access) presents several approaches towards linked data generation:
    - **ETL** (Extract Transform Load) – generates triples, enriches them and adds them to a triple store (Persistent Views);
    - **On-the-fly** – Virtual RDF Graphs (Transient Views) that contain all data that we need from the database and are updated on-the-fly.
    - **API** – Ontop is a Java library, so it can also be called from within programs.
- **Test Driven Development** works well with an OBDA methodology, because:
    - Database to Linked Data mappings can be tested while being written. The editor indicates via highlighting if a mapping is syntactically correct. Only correct mappings are saved.
    - Linked Data can be tested via SPARQL queries during development directly from Protégé.
    - On-the-fly testing can also be tried with Sesame.
    - Test cases for linked data can also be implemented.
- Support for **multiple mapping languages**:
    - The SPARQL to SQL query rewriting algorithms implemented by Ontop work best with its own mapping language.
    - The Ontop mapping language is a version of Direct Mapping.
    - R2RML - can import or generate R2RML mappings as well.
- **Good integration between ontology, data and queries**.
    - In many cases when publishing linked data we discover we need to add classes or properties. This makes the process of linked data publishing tedious, requiring several iterations. Some iterations can be reduced because Ontop allows to easily switch between the ontology, mappings and SQL results, and the SPARQL results and modify them when needed.
- **Easy refactoring**.
    - All mappings are saved in an OBDA file (that has extension .obda) and all queries in a query file (with extension .q), which allows changing namespaces as needed even without starting Protégé.
    - Namespaces can be changed via Protégé.

It is also worth mentioning that OBDA is on the road towards industry adoption [22].

### 3.3.2 RDB2RDF with Ontop

While the ideas behind OBDA methodology are not new, the product that implements it (Ontop) is new. It is worth mentioning that:

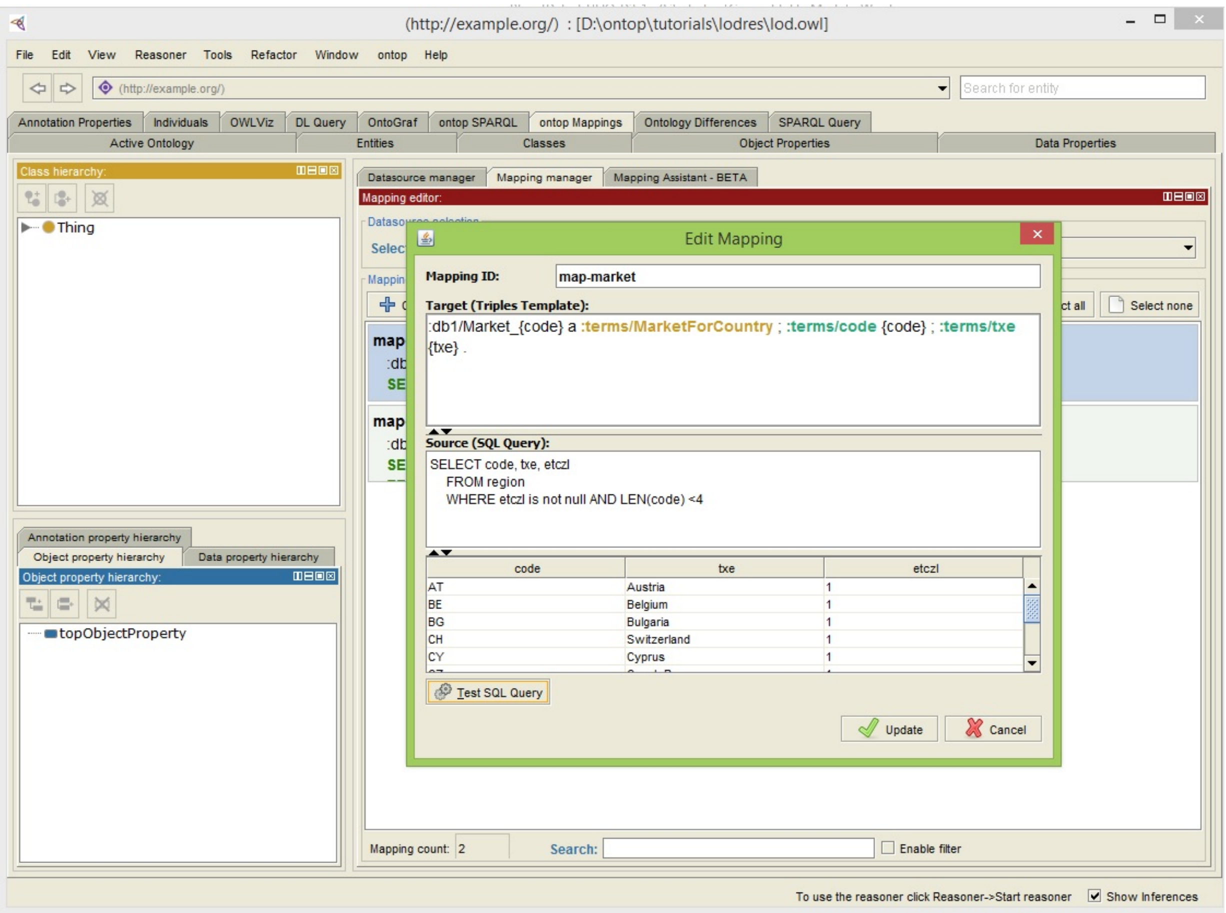
- It has one of the fastest reasoners: Quest.
- Offers support for both Direct Mapping and R2RML. Ontop mapping language typically uses Direct Mapping.
- It integrates very well with Protégé.
- Its mapping language uses graph patterns as formalized by Perez, Arenas and Gutierrez [23].
- It supports federation from multiple databases via Teiid.
- It scales well, as it can easily be seen from some of the examples provided.
- It integrates well with Sesame.

- It can be used for TDD: you can test a mapping via a SPARQL query immediately after you added it to the mappings list.
- It is one of the few tools that have good support for SQL Server.

The product itself also has some disadvantages:

- It is still an experimental product, subject to several changes between versions.
- The support for more complex SQL queries or views is still preliminary. Currently only basic SQL operators are supported.
- Since it is a Protégé plug-in, it shares some of its disadvantages.
    - Not all ontologies can be loaded.
    - The handling of namespaces sometimes causes various errors which require restarting Protégé.
    - Some of the Ontop forms would really need more screen real-estate (it would be nice to also see the SPARQL view in the Mapping Editor, for example), but that is not always possible in Protégé even in Full HD or larger resolutions.

Another disadvantage is that Ontop does not come with a Linked Data Frontend solution such as its direct competitors (Virtuoso, D2RQ). We do not see this as a disadvantage, as there are several other tools that provide this functionality, as already discussed in Section 2.1.


(http://example.org/) : [D:\ontop\tutorials\lodres\lod.owl]
File Edit View Reasoner Tools Refactor Window ontop Help
Annotation Properties Individuals OWLViz DL Query OntoGraf ontop SPARQL ontop Mappings Ontology Differences SPARQL Query
Active Ontology Entities Classes Object Properties Data Properties
Class hierarchy:
Thing
Datasource manager Mapping manager Mapping Assistant - BETA
Mapping editor:
Annotation property hierarchy
Object property hierarchy Data property hierarchy
topObjectProperty
Select none
Mapping count: 2 Search: Enable filter
To use the reasoner click Reasoner->Start reasoner Show Inferences

Edit Mapping

Mapping ID: map-market

Target (Triples Template):

```
:db1/Market_{code} a :terms/MarketForCountry ; :terms/code {code} ; :terms/txe
{txe} .
```

Source (SQL Query):

```
SELECT code, txe, etczl
   FROM region
   WHERE etczl is not null AND LEN(code) <4
```

| code | txe | etczl |
|---|---|---|
| AT | Austria | 1 |
| BE | Belgium | 1 |
| BG | Bulgaria | 1 |
| CH | Switzerland | 1 |
| CY | Cyprus | 1 |

Test SQL Query

Update Cancel

**Figure 2 Ontop Mappings Editor**

Since it is under heavy development, Ontop will likely improve in its future versions.

### 3.3.3 Mappings

In order to transform relational data into RDF, Ontop uses its own mapping language. This language is related to Direct Mapping and R2RML. A screenshot of the mapping editor is depicted in Figure 2.

The mapping language enables defining mapping axioms as pairs of *source* and *target*. The source is the SQL query, and the target is a triple template that defines the look of the URIs. Placeholders are used instead of the source columns in the target.

Multiple prefixes can be used in a query. The format for target templates is the classic S-P-O (Subject-Predicate-Object) Turtle format that is already familiar to the Semantic Web Community. The type of the object data can be inferred by the Quest reasoner or declared by the user.

A simple mapping for TourMIS data is:

```
mappingId  map-market
target          :db1/Market_{code} a :ebo/MarketForCountry ; rdfs:label
{txe}.
source          SELECT code, txe, etczl
                     FROM region
                     WHERE etczl is not null AND LEN(code) <4
```

The mapping is for a reduced set of data (only the type and the label) that could describe a market. Such a mapping would generate 2 triples for each row returned by the SQL query. Here are the triples returned for the row that contains data for Austria.

```
eds:market_AT a ebo:MarketForCountry;
     rdfs:label “AT”.
```

### 3.3.4 Datasets

The datasets we published are those modelled in our previous deliverable D26.1 and displayed in Table 4.

**Table 4 Overview of Datasets**

| Dataset | Components | | | Observation frequency |
|---|---|---|---|---|
| | Dimensions | Measures | Attributes | |
| Arrivals | Time<br>Market<br>City | Arrivals | unitMeasure | Monthly<br>Yearly |
| Bednights | Time<br>Market<br>City | Bednights | unitMeasure | Monthly<br>Yearly |
| Capacity | Time<br>city | Capacity | unitMeasure | Yearly |
| Arrivals At POIs | Time<br>PointOfInterest<br>City | ArrivalsAtPOIs | unitMeasure | Monthly<br>Yearly |
| Shopping Items | Time<br>ShoppingItem<br>City | ShoppingItemPrice | Currency | Yearly |

## 3.4 Linked Data Quality Assurance

We decided to use a TDD approach towards LDQA (Linked Data Quality Assurance). Some of the steps we have followed to make sure that our data meets the current QA standards are:

- We queried the Virtual RDF Graphs created by Ontop, using the Ontop SPARQL endpoint integrated in Protégé every time we created new mappings.
- We checked data dumps of the datasets to see if they contain valid Turtle.
    - Validate using Apache Jena (riot command line tool contains an option for validation).
    - A second test was performed by uploading dumps of the data to the Sesame repository since Sesame rejects files with invalid syntax.
- We checked if there were matches to the entity data (regions, authorities) in other datasets.
- We randomly checked if certain links from our datasets or sameAs links still exist.

Future versions will also contain test cases such as those described in [7] and [8], because maintaining valid data online is also an important part of the Social Contract of Linked Data[6] (in order to support others to make use of a dataset, it is the data publishers responsibility to ensure that the data access point is available online and that the data source contains valid data which is constantly updated).

## 3.5 Interlinking

We used Silk [8] to create links between the ETIHQ datasets and classic datasets such as DBpedia and Geonames.

We compiled Geonames using a Python script, and used the dumps for DBpedia.

We then split DBpedia by the type of entities we were interested in (countries and cities, for Geonames; countries, cities, organizations for DBpedia). We decided to use RDFSlice because it also returned all the triples related to a certain object regardless of its position as subject or object in a statement (inverse functional dependency). RDFSlice typically uses simple SPARQL queries such as the next one when searching for matches in the dumps:

```
select *
where {
{?s a http://dbpedia.org/ontology/City.?s ?p ?o.}
union
{?s1 a http://dbpedia.org/ontology/City.?o1 ?p1 ?s1.}
}
```

We used the rdfs:label property and the recommended settings for Silk in order to generate the links:

- Equality/Inequality
- Levensthein Distance
- Jaccard Distance
- wgs84

Another aspect of the interlinking step, mainly the links with other statistical datasets that might contain interesting or related indicators, is covered in Section 4 of this report.

## 3.6 Linked Data Publishing

We consider Linked Data Publishing to contain the following sub-stages:

[6] http://www.w3.org/TR/ld-bp/#SOCIAL-CONTRACT

- Setting up a triple store.
- Setting up a SPARQL endpoint.
- Setting up a Linked Data Frontend.

We think that this stage to be the equivalent of a Long Term – Relationship in humans life. Sometimes it is considered also a Social Contract, especially in the Open Government circles.

When you publish the data online, you have to provide the following:

- Clear URIs and structure (this is why Semantic Modelling is probably the hardest part when creating new datasets);
- Long-Term Support;
- Good documentation.

For the moment we consider that our previous deliverable does a good job of providing the initial documentation, and as soon as the users will start creating applications we will talk with them and provide them with an extended documentation.

### 3.6.1 Tripe store

The ontologies and the triples that resulted from the RDB2RDF and Interlinking stages were uploaded to a Sesame repository. While not the best triple store available, Sesame is a good solution for maintaining datasets that reach up to 100-150 million triples[7].

For testing the generated triples we splat them by dataset and uploaded them to a Sesame repository.

The production version will use an In Memory Store. Should our dataset exceed 100-150 million triples we will consider switching to a more scalable triple store such as Fuseki with TDB or Virtuoso.

The size of the RAM will increase with the size of the triple store.

### 3.6.2 SPARQL endpoint

A SPARQL endpoint is important in order to be able to query a triple store. It is recommended to use a SPARQL 1.1 endpoint if the triple store supports SPARQL 1.1, as UPDATE queries could be good solutions for improving data quality.

Sesame already offers a SPARQL endpoint[8].

In addition to the Sesame SPARQL endpoint, we have also added anotherSPARQL interface: YASGUI.[9]

### 3.6.3 Linked Data Frontend

We used Pubby to publish a Linked Data frontend to our datasets. While not an LDP solution yet, the latest Pubby[10] does offer several good options for a Linked Data frontend:

- CONSTRUCT queries to offer more control on what is displayed on screen;
- Metadata extension;
- Responsive design;
- It allows generating HTML pages for resources that are outside of the datasetBase;
- Uses IRIs.

Future versions of Pubby will also offer a plug-in system. One of the proposed plug-ins is dedicated to geo visualizations, which will play an important role on the decision support systems planned to use the TourMIS Linked Data

[7] http://rivuli-development.com/further-reading/sesame-cookbook/loading-large-file-in-sesame-native/

[8] http://data.etihq.eu/

[9] http://data.etihq.eu/yasgui/

[10] https://github.com/cygri/pubby

It is likely that when the size of the dataset will increase (more than 150-200 million triples would already be more than Geonames, for example) we will also switch to ELDA, as it offers a much better solution for publishing large scale datasets. We have already started experimenting with it. An advantage of ELDA or other LDPs when publishing statistical data is the extended support for slicing the data (observation groups, slices) and support for maps, which is something that users might need.

## 3.7 Linked Data Applications

A good use case for Linked Data applications that use TourMIS data is represented by geographic visualizations. They could be implemented either as mashups, as it seems to be the case today, or in a separate interface.

Another good use case is represented by classical statistical visualizations.

Applications are not however part of the current package, so they will be described in our next work package.

# 4 Linking Indicators

TourMIS data consists of a collection of tourism related indicators. As we have seen in the previous section, these indicators can be published as Linked Data using the RDF Data Cube Vocabulary. We consider this work to be the foundation of future tourism research, therefore we consider that indicator modelling is a topic worth addressing.

We have two types of indicators:

- *Internal indicators* – indicators that are already present in our database (or triple store, after publishing them as Linked Data);
- *External indicators* – indicators that come from external sources like World Bank[11], Eurostat[12].

Most of our work is focused on publishing and consuming the internal indicators from TourMIS, but if there are similar indicators from external sources, then we are interested to access them, collect them, and plot them together with TourMIS data as part of the envisioned decision support system which will be built within ETIHQ.

For collecting such data, we follow the next steps:

- Query external data sources for external indicators;
- Decide if the external indicators are valuable for us;
    - Automatic detection of such indicators is well beyond the scope of our project, but it is an interesting research problem that we intend to address in the future.
- Collect the external data:
    - Use dumps (where available and if the dataset is small and of interest);
    - Slice it using RDFSlice if the dataset is too big (some of the datasets might have several TBs, but the data related to tourism will probably be only a small part);
    - Use CONSTRUCT queries - there is no need to use something more complex than a CONSTRUCT query if only one indicator from an external dataset would provide yearly data that is similar or complementary to data from TourMIS.
- Create links in our current data to the external datasets.
- Process the information about these indicators:
    - Calculate various aggregate functions (AVG, SUM, COUNT, MAX, MIN, etc);
    - Find a method to present it with the current information.

It is often the case that the information presented in other sources is complementary to the data we provide. Here are some examples for external indicators that we have found:

- World Bank – the data comes typically from the WDI (World Development Indicators).
    - International tourism, number of arrivals[13] – it is provided for countries, and it therefor complements city level statistics offered in TourMIS.. We can compare country totals with the total of important cities from a country (typically the cities covered by TourMIS).
    - International tourism, number of departures[14] – provided as yearly measures, this indicator can be used together with our market information from TourMIS in order to understand how many times per year people from a certain country travel to a certain destination (it would not help us to get the actual number of people travelling, as some of these people might make multiple trips).

[11] http://worldbank.270a.info/.html

[12] http://eurostat.linked-statistics.org/

[13] http://worldbank.270a.info/classification/indicator/ST.INT.ARVL.html

[14] http://worldbank.270a.info/classification/indicator/ST.INT.DPRT.html

    - Other indicators that might be valuable are GDP
- Eurostat – the data comes from Eurostat[15], and was published in SDMX format. A conversion to Linked Data was made in the Linked Statistics project.
    - Nights spent at tourist accommodation establishments by coastal and non-coastal area (from 2012 onwards) (tour_occ_ninatc)
    - Number of establishments, bedrooms and bed-places by coastal and non-coastal area (from 2012 onwards) (tour_cap_natc)
    - Arrivals by type of accommodation (med_to12)
    - Nights spent by type of accommodation (med_to13)
    - Accommodation establishments (med_to21)
    - Number of bed-places (med_to22)
    - Monthly data on tourism industries (tour_indm)
    - Annual data on tourism industries (tour_inda)

It is clear that there is a lot of tourism data in the Eurostat datasets, but it was measured differently than TourMIS in some cases, and in other cases the collecting methodology has been changed since 2012, leaving us with split datasets (up to 2011, after 2012).

For all external indicators we also collect the information that describes them (data structure documents, code lists, and so on).[16]

We consider this topic to be relevant for our next package, as people are always interested to compare data from various sources.

[15] There is more tourism data in Eurostat than the data discussed in this paragraph. A list can be browsed using this link: http://epp.eurostat.ec.europa.eu/NavTree_prod/everybody/BulkDownloadListing?sort=1&file=table_of_contents_en.pdf

[16] See: http://data.etihq.eu/

# 5 Conclusions and Future Work

We have hereby described the process that we followed in order to collect data for the ETIHQ repository, as well as the data that is contained in this repository. The concrete outcomes of this work are:

- A blueprint for publishing statistical tourism data using QB vocabulary that details all the stages and tools that might be used.
- The five datasets resulted from the RDB2RDF conversion in Turtle format.
- Various frontends to the Linked Data: SPARQL GUIs, an HTML interface.

The data and interfaces are available at http://data.etihq.eu.

We can draw several conclusions from our work. Firstly, publishing the data was not a complicated process, but finding the tools that could help us do it efficiently while also incorporating some Quality Assurance steps was a complicated task (most tools did not support SQL Server well at the time when we started to develop these datasets). Secondly, OBDA is an efficient methodology for RDB2RDF conversions, and can help in reducing the learning curve of Linked Data publishing if the tools mature. Thirdly, no clear methodologies exist for supporting the whole chain for collecting, describing and using data from other datasets. There are mechanisms to describe the data (void, DCAT, PROV-O) that help dataset discovery and detecting provenance, but since SPARQL is not able to perform complex computations in the same manner as SQL, combining and aggregating values from multiple datasets is not straightforward. The real problem lies in the fact that the structure of linked data statistics datasets is easy to read by both humans and machines, but the real hints regarding the dataset usability lie in the comments that only people (typically experts) can interpret. There is a need for logic-based formalisms or DSLs that could help in this matter.

Future work will include development of mashups or applications that use this data. First stages of this process might require revisiting the current data, ontologies or interfaces. Once the early stages of the applications are done we plan to invite more users to use our applications.

# References

[1] Sabou, M., Arsal, I., Brașoveanu, A.M.P. 2013. TourMISLOD: a Tourism Linked Data Set. Semantic Web Journal 4(3): 271-276.

[2] Wöber, K. 2003. Information supply in tourism management by marketing decision support systems. Tourism Management. 24(3):241-255.

[3] Sabou, M., Brașoveanu, A.M.P. 2014. D26.1 Call2: Semantic Modelling of Tourism Indicators. PlanetData Deliverable.

[4] Myrseth, P., Øverby, E., Yang, J.J. 2013 Survey on Vocabulary and Ontology Tools. Available at http://www.semicolon.no/wp-content/uploads/2013/09/Semicolon_Vocabulary-tools-survey_v1.0.pdf.

[5] Mendes, P.N., Mühleisen, H., Bizer, C. 2012. Sieve: linked data quality assessment and fusion. EDBT/ICDT Workshops 2012: 116-123.

[6] Kontokostas, D., Westphal, P., Auer, S., Hellmann, S., Lehmann, J., Cornelissen, R.. 2014. Databugger: a test-driven framework for debugging the web of data. WWW (Companion Volume) 2014: 115-118.

[7] Kontokostas, D., Westphal, P., Auer, S., Hellmann, S., Lehmann, J., Cornelissen, R., Zaveri, A. 2014. Test-driven evaluation of linked data quality. WWW 2014: 747-758.

[8] Isele, R., Jentzsch, A., Bizer, C. 2010. Silk Server - Adding missing Links while consuming Linked Data. COLD 2010.

[9] Michel, F., Montagnat, J., Faron-Zucker, C. 2014. A survey of RDB to RDF translation approaches and tools. Under Review. Semantic Web Journal.

[10] Das, S.., Sundara, S., Cyganiak, R. 2012. R2RML: RDB to RDF Mapping Language. Available at: http://www.w3.org/TR/r2rml/.

[11] Arenas, M., Bertails, A., Prud'hommeaux, E., Sequeda, J. 2012. A Direct Mapping of Relational Data to RDF. W3C Recommendation. Available at: http://www.w3.org/TR/rdb-direct-mapping/.

[12] Sequeda, J., Arenas, M., Miranker, D.P. 2012. On directly mapping relational databases to RDF and OWL. WWW 2012: 649-658.

[13] Sequeda, J. 2013. On the Semantics of R2RML and its Relationship with the Direct Mapping. International Semantic Web Conference (Posters & Demos) 2013: 193-196.

[14] Sequeda, J., Tirmizi, S.J., Corcho, O., Miranker, D.P. 2011. Survey of directly mapping SQL databases to the Semantic Web.Knowledge Eng. Review 26(4): 445-486.

[15] Rietveld, L., Hoekstra, R. 2013. YASGUI: Not Just Another SPARQL Client. ESWC (Satellite Events) 2013: 78-86.

[16] Speicher, S., Arwe, J., Malhotra, A. 2014. Linked Data Platform 1.0, W3C Reccomendation, Available at: http://www.w3.org/TR/ldp/.

[17] Cyganiak, R., Reynolds, D., Tennison, J. 2013. The RDF Data Cube Vocabulary. W3C Reccomendation. Available at: http://www.w3.org/TR/vocab-data-cube/.

[18] Mendes, P. N., Bizer, C., Miklos, Z., Calbimonte, J., Moraru, A., Flouris, G. 2012. D2.1 Conceptual model and best practices for high-quality metadata publishing. PlanetData Deliverable.

[19] Hyland, B., Atemezing, G., Villazón-Terrazas, B. 2014. Best Practices for Publishing Linked Data. W3C Working Group Note. Available at: http://www.w3.org/TR/ld-bp/.

[20] Mena, E., Kashyap, V., Sheth, A.P., Illarramendi, A. 1996. OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies. CoopIS 1996: 14-25.

[21] Rodriguez-Muro, M., Kontchakov, R., Zakharyaschev, M. 2013. Ontology-Based Data Access: Ontop of Databases. International Semantic Web Conference (1) 2013: 558-573.

[22] Kharlamov, E., Jiménez-Ruiz, E., Zheleznyakov, D., Bilidas, D., Giese, M., Haase, P., Horrocks, I., Kllapi, H., Koubarakis, M., Özçep, O.L., Rodriguez-Muro, M., Rosati, R., Schmidt, M., Schlatte, R., Soylu, A., Waaler, A. 2013. Optique: Towards OBDA Systems for Industry. ESWC (Satellite Events) 2013: 125-140.

[23] Pérez, J., Arenas, M., Gutierrez, C. 2006. Semantics and Complexity of SPARQL. International Semantic Web Conference 2006: 30-43

[24] Marx, E., Shekarpour, S., Auer, S. 2013. Large-Scale RDF Dataset Slicing. ICSC 2013: 228-235.