



PlanetData

Network of Excellence

FP7 – 257641

D5.1 PlanetData data management tools catalogue and access portal

**Coordinator: Pablo Mendes (FUB), Zoltán Miklós (EPFL),
Freddy Priyatna (UPM), Oscar Corcho (UPM)**

With contributions from: EPFL, UPM, KIT, IJS, UIBK

**1st Quality reviewer: Lyndon Nixon
2nd Quality reviewer: Carolina Fortuna**

Deliverable nature:	Report (R)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	M12
Actual delivery date:	
Version:	1.1
Total number of pages:	28
Keywords:	

Abstract

PlanetData data management tools catalogue and access portal is available as a part of the portal <http://www.planet-data.eu/> (developed in WP7 of the project). This report gives a short overview of the tool catalogue that is available at <http://www.planet-data.eu/results/data-and-toolsets>. The catalogue is continually maintained and updated.

Executive summary

PlanetData data management tools catalogue and access portal is available as a part of the portal developed in WP7. This report gives a short overview of the tool catalogue and the related activities. The PlanetData Lab, that is available at

<http://www.planet-data.eu/results/data-and-toolsets>

is part of this deliverable. The current list of tools include

- GSN, OKKAM (EPFL)
- LDSPider, cumulusRDF (KIT)
- R2O/ODEMapster, S2O Platform, geometry2RDF (UPM)
- monetDB (CWI)
- D2R server, Silk, Pubby, Dbpedia spotlight (FUB)
- LarKC (UIBK)

The deliverable also reports on data provisioning tools.

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	D5.1	Title	PlanetData data management tools catalogue and access portal
Work Package	Number	WP5	Title	PlanetData Lab

Date of Delivery	Contractual	M12	Actual	Mxx
Status	version 1.1		final <input type="checkbox"/>	
Nature	prototype <input type="checkbox"/> report <input checked="" type="checkbox"/> dissemination <input type="checkbox"/>			
Dissemination level	public <input checked="" type="checkbox"/> consortium <input type="checkbox"/>			

Authors (Partner)	FUB			
Responsible Author	Name	Pablo Mendes	E-mail	pablo.mendes@fu-berlin.de
	Partner	FUB	Phone	

Abstract (for dissemination)	
Keywords	

Version Log			
Issue Date	Rev. No.	Author	Change
V0.1		Zoltan Miklos	
V0.2			Input from UPM
V0.3			Update based on the peer reviews
V1.0			Reviewer's comments incorporated
V1.1			Version to submit to EC

Table of Contents

Executive summary	3
Document Information	4
Table of Contents	5
Abbreviations	6
1 Introduction	7
2 Goals of the tool catalogue development	8
Versions	8
Cataloging	8
Uses of the catalogued tools	8
Next steps.....	8
3 Catalogue organisation.....	10
Metadata structure.....	10
Actual list of tools.....	12
4 RDF data provisioning	17
Introduction.....	17
Direct provisioning tools.....	17
Direct provisioning tools from Relational Data	18
Direct provisioning tools from XML and XLS	18
Direct provisioning tools from geometrical data.....	19
Mapping-based provisioning tools.....	19
Mapping-based provisioning tools for relational data.....	20
Mapping-based provisioning tools from sensor data	20
MappingPedia: exploiting mappings from data provisioning tools	20
Purpose.....	20
Vocabulary	21
Current status and future work.....	24
Conclusion	25
5 Conclusion.....	27
References	28

Abbreviations

PlanetData PlanetData Network of Excellence

1 Introduction

Processing and experimenting with sizable amounts of data requires adequate systems and infrastructure. PlanetData has set up a portal and provides access to existing technologies for large-scale data management for the benefit of PlanetData core and associate members, and subsequently the European research community at large.

This deliverable describes a catalogue of tools for large-scale data management, with particular focus on sensor data and linked data. The catalogue helps PlanetData partners in the use of the listed tools, by explicit links to documentation, use cases.

The tool catalog is an organic part of the dissemination platform developed in WP7 and available at:

<http://www.planet-data.eu/results/data-and-toolsets>

The deliverable is organized as follows. Section 2 summarizes the goals of the PlanetData lab and its development. Section 3 discusses the organization of the tool catalogue. Section 4 is dedicated to the tools on data provisioning and finally Section 6 concludes the deliverable.

2 **Goals of the tool catalogue development**

The tool catalog was developed to support PlanetData core and associated partners to use the tools available in the consortium. The catalog should foster the collaboration between partners. To achieve this goal we collected useful information about the tools, in particular explicit links to documentations, installation instructions (if applicable). Most importantly, we also included contact persons and their coordinates. These contact persons are available to help, to provide further technical details or put the partners in contact with the local experts.

Versions

The PlanetData lab is available as a part of the dissemination portal from the first release on. We have gradually improved the tool catalogue, with a new release in month 11, together with the new release of the portal.

PlanetData lab is available in early versions also on the project wiki that is the main platform for collecting new content for later versions. The URL of the wiki page is http://wiki.planet-data.eu/web/PlanetData_Lab.

We plan to regularly update the tool catalogue every 3 months of the project or on demand. The project partners continually maintain the wiki pages that are serving always the basis for new releases of the PlanetData lab on the official PlanetData portal.

Cataloging

The PlanetData lab is a collection of tools that we often refer as a tool catalogue. However we did not classify or categorize the tools. We included the relevant tools from the core partners, and we added appropriate labels. Depending on the growth of the PlanetData lab we might adopt a classification or a relevant ontology.

We plan to include further tools from associated members or now partners who join the consortium, for example through the PlanetData programs. The selection criteria for new tools are the willingness of partners to provide the relevant information, metadata and documentation and most importantly support. We do not impose explicit additional requirements for the maturity of the tools, however the included tools should be mature enough such PlanetData partners should be able to benefit from the use of them. Moreover the partners should provide appropriate support in installation and/or use of the tool.

Uses of the catalogued tools

We have a number of collaborative projects. These include cooperation between EPFL and IJS and UPM in the context of WP1 or the uses of monetDB within the consortium by many partners. These joint projects will appear in the tool catalogue as examples. In particular, they will form the basis of best practices in the second year of the project (task T5.2). In the current collaboration efforts, the partners largely benefited from the extensive documentation available through the PlanetData lab, as well as from the personal support, for which PlanetData lab offers a straightforward entry point.

Next steps

Our goal is to continually extend the existing collection. As new partners have joined to the consortium recently, we plan to include the tools they can offer as well. The same ways we will include tools from associated partners as long they are ready and willing to provide support for the PlanetData participants.

Beyond extending the list of tools we plan to add joint projects and best practices that are planned for the second phase of PlanetData. Moreover, we would like to reach the European research community and give an even more broad support in the use of the tools.

3 Catalogue organisation

Metadata structure

The tool catalog contains the following metadata for each tool.

- in a nutshell (short description of the tool)
- Documentation (link to the documentation pages)
- Requirements (requirements for using or installing the tools)
- License (the type of license needed to use it)
- Contact person/ mailing list (the name and contact address of a person who could help with the use. If appropriate, a mailing list of users is also indicated.)
- Organization/person behind (the organization that developed the tool or providing support, for PlanetData partners)
- Publications (link to some representative publication about the tool)
- Events (a link to information about events related to the particular tool)
- Category

An example page of a catalog entry is depicted in Figure 1.

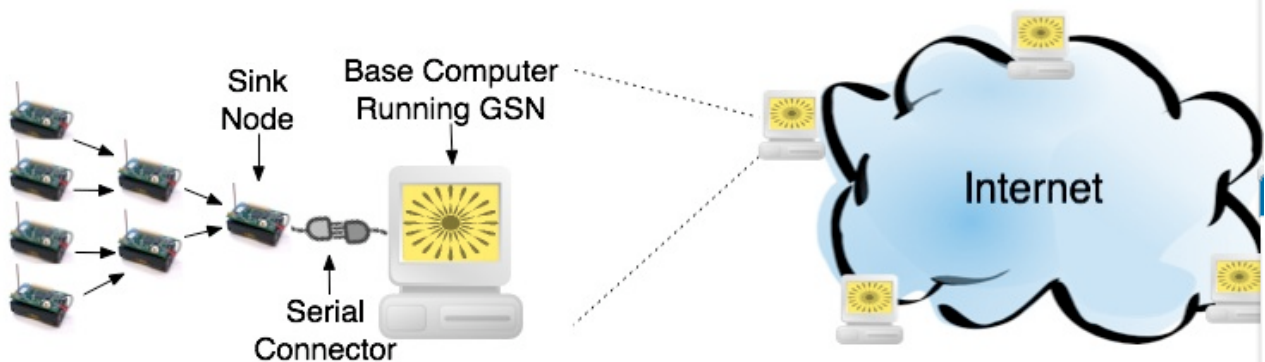
GSN (Global Sensor Networks)

What is GSN ?

GSN is a software middleware designed to facilitate the deployment and programming of sensor networks.

How it works

GSN is a Java environment that runs on one or more computers composing the backbone of the acquisition network. A set of wrappers allow to feed live data into the system. Then, the data streams are processed according to XML specification files. The system is built upon a concept of sensors (real sensors or virtual sensors, that is a new data source created from live data) that are connected together in order to build the required processing path. For example, one can imagine an anemometer that would send its data into GSN through a wrapper (various wrappers are already available and writing new ones is quick), then that data stream could be sent to an averaging mote, the output of this mote could then be split and sent for one part to a database for recording and to a web site for displaying the average measured wind in real time. All of this example could be done by editing only a few XML files in order to connect the various motes together.



GSN (Global Sensor Networks)	
Short description:	Software middleware designed to facilitate the deployment and programming of sensor networks
Website:	GSN official website
Requirements:	Java
Download:	http://sourceforge.net/apps/trac/gsn/wiki/Download
Documentation:	http://sourceforge.net/apps/trac/gsn/wiki/Documentation
Publications:	http://sourceforge.net/apps/trac/gsn/wiki/Publications
Events:	http://sourceforge.net/apps/trac/gsn/wiki/Workshop
License:	GNU GPL
Contact person:	Hoyoung Jeung
Partner:	EPFL
sameAs:	none

Figure 1 Metadata for GSN

Currently we use the following categories: *Produce/Publish/Consume/Provisioning* that refers to the data management phase in which one can apply the given tool. In fact, often these categories are not exclusive; a tool might fall into multiple categories. Instead of defining hierarchical or disjoint categories, we consider these as labels. We have also introduced a special category “*data management*”, as the monetDB database

is a complex data management system, rather than a simple tool to support only one special task. While the use of this simple categorization is sufficient for the current set of tools, we might reconsider the use of other categorization, if the catalogue grows larger.

Actual list of tools

The current list of tools includes:

cumulusRDF

RDF store for cloud-based architectures. Cumulus provides a REST-based API with CRUD operations to manage RDF data. The current version uses Apache Cassandra as storage backend.

D2R

D2R Server is a tool for publishing the content of relational databases on the Semantic Web, a global information space consisting of linked data.

Data on the Semantic Web is modelled and represented in RDF. D2R Server uses a customizable D2RQ mapping to map database content into this format, and allows the RDF data to be browsed and searched – the two main access paradigms to the Semantic Web.

DBpedia Spotlight

DBpedia Spotlight is a tool for annotating mentions of DBpedia resources in text, providing a solution for linking unstructured information sources to the Linked Open Data cloud through DBpedia. DBpedia Spotlight performs named entity extraction, including entity detection and Name Resolution (a.k.a. disambiguation). It can also be used for building your solution for Named Entity Recognition, amongst other information extraction tasks.

Text annotation has the potential of enhancing a wide range of applications, including search, faceted browsing and navigation. By connecting text documents with DBpedia, our system enables a range of interesting use cases. For instance, the ontology can be used as background knowledge to display complementary information on web pages or to enhance information retrieval tasks. Moreover, faceted browsing over documents and customization of web feeds based on semantics become feasible. Finally, by following links from DBpedia into other data sources, the Linked Open Data cloud is pulled closer to the Web of Documents.

You can try out DBpedia Spotlight through our Web Application or Web Service endpoints. The Web Application is a user interface that allows you to enter text in a form and generates an HTML annotated version of the text with links to DBpedia. The Web Service endpoints provide programmatic access to the demo, allowing you to retrieve data also in XML or JSON.

GSN (Global Sensor Networks)

GSN is a software middleware designed to facilitate the deployment and programming of sensor networks.

Geometry2RDF

A tool that generates RDF triples from geometrical information, which can be available in GML or WKT.

LDIF

<p>The Web of Linked Data grows rapidly and contains data from a wide range of different domains, including life science data, geographic data, government data, library and media data, as well as cross-domain datasets such as DBpedia or Freebase. Linked Data applications that want to consume data from this global data space face the challenges that:</p>

- | |
|--|
| <ol style="list-style-type: none"> 1. data sources use a wide range of different RDF vocabularies to represent data about the same type of entity. 2. the same real-world entity, for instance a person or a place, is identified with different URIs within different data sources. |
|--|

<p>This usage of different vocabularies as well as the usage of URI aliases makes it very cumbersome for an application developer to write SPARQL queries against Web data which originates from multiple sources. In order to ease using Web data in the application context, it is thus advisable to translate data to a single target vocabulary (vocabulary mapping) and to replace URI aliases with a single target URI on the client side (identity resolution), before starting to ask SPARQL queries against the data.</p>
--

<p>Up-till-now, there have not been any integrated tools that help application developers with these tasks. With LDIF, we try to fill this gap and provide an open-source Linked Data Integration Framework that can be used by Linked Data applications to translate Web data and normalise URI while keeping track of data provenance.</p>
--

<p>LDIF provides an expressive mapping language for translating data from the various vocabularies that are used on the Web into a consistent, local target vocabulary. LDIF includes an identity resolution component which discovers URI aliases in the input data and replaces them with a single target URI based on user-provided matching heuristics. For provenance tracking, the LDIF framework employs the Named Graphs data model.</p>
--

LDSpider (Linked Data Spider)

LDSpider is a web crawling framework for the Linked Data web.

LarKC (Large Knowledge Collider)

<p>The overall aim of LarKC is to build an integrated platform for semantic computing on a scale well beyond what is currently possible.</p>
--

<p>We develop the Large Knowledge Collider, a pluggable algorithmic framework implemented on a distributed computational platform. This will allow reasoning at Web scale by trading quality for computational cost and by embracing incompleteness and unsoundness.</p>
--

- | |
|---|
| <ul style="list-style-type: none"> • Plug-in Architecture: Instead of being built only on logic, the Large Knowledge Collider allows to exploit a large variety of methods from other fields: cognitive science (human heuristics), economics (limited rationality and cost/benefit trade-offs), information retrieval (recall/precision trade-offs), and databases (very large datasets). A pluggable architecture ensures a coherent integration of various components. |
|---|

- **Distributed and Parallel Computing:** The Large Knowledge Collider makes use of parallel hardware using cluster computing techniques. In this way it aims to leverage large-scale, distributed computational resources in order to meet the scalability requirements of current, data-driven applications in semantic computing.

MonetDB

A relational database management system for high-performance data warehouses for business intelligence and eScience.

R2O and ODEMapster

The UPM framework to Upgrade Relational Legacy Data to the Semantic Web consists of

- R2O, a fully declarative, XML-based language that allows the description of arbitrarily complex mapping expressions between ontology elements (concepts, attributes and relations) and relational elements (relations and attributes).
- The ODEMapster processor, which generates Semantic Web instances from relational instances based on the mapping description expressed in an R2O document
- ODEMapster plugin provides users a Graphical User Interface that allows to create, execute, or query mappings between ontologies and databases

OKKAM

The OKKAM project aims at enabling the Web of Entities, namely a virtual space where any collection of data and information about any type of entities (e.g. people, locations, organizations, events, products, ...) published on the Web can be integrated into a single virtual, decentralized, open knowledge base (like the Web did for hypertexts). OKKAM contributes to this vision by supporting the convergence towards the use of a single and globally unique identifier for any entity which is named on the Web. The intuition of the project is that the concrete realization of the Web of Entities requires that we enable tools and practices for cutting to the root the proliferation of unnecessary new identifiers for naming the entities which already have a public identifier (the OKKAM's razor). Therefore, OKKAM makes available to content creators, editors and developers a global infrastructure and a collection of new tools and plugins which support them to easily find public identifiers for the entities named in their contents/services, use them for creating annotations, build new network-based services which make essential use of these identifiers in an open environment (like the Web or large Intranets).

Pubby

Much Semantic Web data lives inside triple stores and can be accessed only by sending SPARQL queries to a SPARQL endpoint. It is hard to connect information in these stores with other external data sources.

Linked Data is a style of publishing data on the Semantic Web that makes it easy to interlink, discover and consume data on the Semantic Web. It allows a wide variety of existing RDF browsers (e.g. Disco, Tabulator, OpenLink Browser), RDF crawlers (e.g. SWSE, Swoogle), and query agents (e.g. SemWeb Client Library, SWIC) to access the data.

Pubby makes it easy to turn a SPARQL endpoint into a Linked Data server. It is implemented as a Java web application.

R2R

Data is represented on the Web of Linked Data using terms from a wide range of different vocabularies. The R2R Framework enables Linked Data applications which discover data on the Web, that is represented using unknown terms, to search the Web for mappings and apply the discovered mappings to translate Web data to the application's target vocabulary. The R2R Framework is aimed to be used by Linked Data publishers, vocabulary maintainers and Linked Data application developers. The tool supports these tasks by:

- providing the R2R Mapping Language for publishing fine-grained term mappings on the Web
- defining best-practices on how mappings can be discovered by Linked Data applications
- providing an open-source implementation of the R2R Mapping Engine.

This document gives a short overview of the R2R Framework, describes its installation and configuration and gives several mapping examples.

S2O

A framework for providing access to streaming data based on ontologies, consists of

- SparqlStream : a sparql extension for rdf streams.
- S2O : an extension to R2O for expressing mappings from streaming sources to an ontology.
- SNEE : a query processing engine over relational data streams.

Silk

The Web of Data is built upon two simple ideas: first, to employ the RDF data model to publish structured data on the Web; second, to set explicit RDF links between data items within different data sources. The Silk Link Discovery Framework supports data publishers in accomplishing the second task. Using the declarative Silk - Link Specification Language (Silk-LSL), developers can specify which types of RDF links should be discovered between data sources as well as which conditions data items must fulfill in order to be interlinked. These link conditions may combine various similarity metrics and can take the graph around a data item into account, which is addressed using an RDF path language. Silk accesses the data sources that should be interlinked via the SPARQL protocol and can thus be used against local as well as remote SPARQL endpoints. Silk offers the following features:

- Flexible, declarative language for specifying link conditions
- Support of RDF link generation (owl:sameAs links as well as other types)
- Employment in distributed environments (by accessing local and remote SPARQL endpoints)
- Usable in situations where terms from different vocabularies are mixed and where no consistent RDFS or OWL schemata exist
- Scalability and high performance through efficient data handling (speedup factor of 20 compared to Silk 0.2):

- Reduction of network load by caching and reusing of SPARQL result sets
- Multi-threaded computation of the data item comparisons (3 million comparisons per minute on a Core2 Duo)
- Optional blocking of data items

4 RDF data provisioning

Introduction

This section is dedicated to the RDF data provisioning tools in the PlanetData lab. They are tools which allow transforming various types of data sources available in different data modalities into RDF. They are labelled as “*Provisioning*” tool in our catalogue.

The reason for the existence of these types of tools is the fact that a large amount of the data that is available on the Web is in relational databases, spreadsheet documents, etc., and hence this type of tools have appeared and evolved constantly during the last decade.

While catalogues of these types of tools exist, such as the one available at <http://www.w3.org/wiki/ConverterToRdf> these catalogues are not necessarily up-to-date and the information available in them is not enough for practitioners to understand how these tools should be used, nor is there any hint about whether some tools are still maintained or not, or where they have been used. That is, these catalogues only act as a set of pointers to technology and their description pages.

As part of the PlanetData Lab, we maintain a set of data provisioning tools, which contains enough metadata about the tools that can be used for the purpose of data conversion to RDF and which contains examples about how they can actually be used by practitioners, so as to facilitate their uptake by the community. Besides, we aim at providing a catalogue of existing mappings used in these types of tools, so that these can be reused for different purposes or can be used in global query rewriting processes to identify where relevant data is available when trying to answer a query. This effort is what we call mappingpedia.

This will be a sustained effort during the execution of PlanetData, and work has been planned accordingly. In this period we have concentrated in providing a first categorisation of the types of tools that we will be providing in the PlanetData Lab. Next steps will be to generate appropriate metadata for their description, reusing efforts done and lessons learned in other parts of the NoE, generate the corresponding documentation in the form of best practices, and make these tools available in the PlanetData Lab, beyond providing URLs to their descriptions. (in deliverable D5.2)

In this deliverable we start providing a brief analysis of the state of the art in this type of tools, providing exemplar descriptions of some of these tools, from those that will be made available through the PlanetData Lab, and starting with those that are owned by PlanetData partners.

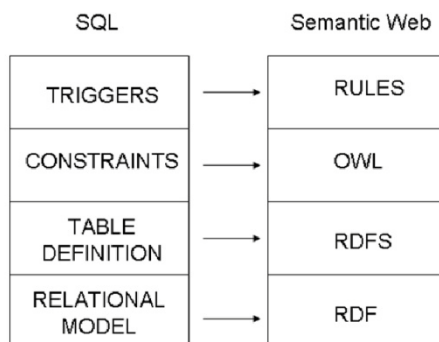
We will make first an analysis of the types of tools that we can find. According to the W3C RDB2RDF Working Group (<http://www.w3.org/2009/08/rdb2rdf-charter.html>), there are two types of RDF provisioning tools: direct translation and mapping-based. In the following sections, we will give brief description of those tools.

Direct provisioning tools

These tools are defined as those that do not use mappings for defining the relationship between the source schema in which the data is available (e.g, the relational data schema, the XML Schema, etc.) and the set of ontologies according to which the data will be generated. In fact, these tools do not normally consider an existing set of ontologies for the transformation. On the contrary, they generate their own vocabulary, which is a representation of the source schema in an ontology language, with not many transformations. We will briefly discuss about some of the existing direct provisioning tools for relational data, XML, spreadsheets and geometrical information in geospatial data-bases.

Direct provisioning tools from Relational Data

Sequeda et al.[12] provide a comparison of a set of tools for transforming re-lational data into RDF without the needs of mappings using the relationship between SQL DDL and the Semantic Web layer as the main driver for discussion.



**Figure 2 Layer Cake
Correspondence between SQL and
the Semantic Web**

As we can see from Figure 1, there are four SQL DDL properties that can be aligned with the Semantic Web layer. The first layer is the relational model layer, which consists of database records, to be aligned with RDF instances. The second layer is the table definition layer, which can be aligned to the RDFS <http://www.w3.org/TR/rdf-schema/> language. The third layer is the constraints layer, which is aligned to OWL <http://www.w3.org/TR/owl-ref/>. The fourth layer is the triggers layer and it is aligned with rules. Furthermore, the mappings between the first three layers are discussed in more detail, as can be seen on Figure 2. It is mentioned that no work has been done on the detailed mapping on the fourth layer. More detailed transformation regarding database normalization, inheritance modeling, symmetric and transitive relationships are out of the scope of this document and the reader is referred to the paper.

Relational Database	RDFS Ontology	OWL Ontology
Non-binary Relation	RDFS Class	OWL Class
Binary Relation	RDFS Property	OWL Object Property
Column	RDFS Property	OWL Datatype Property
Foreign Key	RDFS Property	OWL Object Property

**Figure 3 Summary of Relational Database direct
mapping to RDFS and OWL**

Several tools adopting this approach are studied on the paper such as Ultrawrap[11], Qualeg DB[2], and others[14, 1, 5, 9, 13]. Interested readers are referred to the mentioned survey for the translations rules of each tool to RDF.

Direct provisioning tools from XML and XLS

Another type of data source commonly used in this direct translation approach are XML les, and spreadsheets available as XLS or CVS les. For this data types, several tools have been made available on the state of the art, such as RDF123[7] or XLWrap[8]. The PlanetData Lab will include a tool that is called NOR2O[15], and which allows transforming these type of data into RDF. Currently, it can be downloaded at

<http://mccarthy.dia.fi.upm.es/nor2o/>. Figure 3 depicts the modules of this tool. The NOR Connector loads classification schemes, thesauri, and lexicons modeled with their corresponding data models, and implementations. The Transformer performs the transformations by implementing the sequence of activities included in the patterns. This module interacts with the Semantic Relation Disambiguator module for obtaining the suggested semantic relations of the NOR elements. The Semantic Relation Disambiguator is in charge of obtaining the semantic relation between two NOR elements. Basically, the module receives two NOR elements from the Transformer module and returns the semantic relation between them. The module connects the external resource through the External Resource Service module to get the relation. The External Resource Service is in charge of interacting with external resources for obtaining the semantic relations between two NOR elements. At this moment the module interacts with WordNet. We are implementing the access to DBpedia. The OR Connector generates the ontology in OWL. To this end, this module relies on the OWL API.

Direct provisioning tools from geometrical data

Geometrical information, which can be represented in GML or WKT, plays an important role in the context of the Web of Linked Data, as most entities can be related to geographical information, such as places or area. The PlanetData Lab will contain a tool called geometry2rdf, that transforms geometrical information

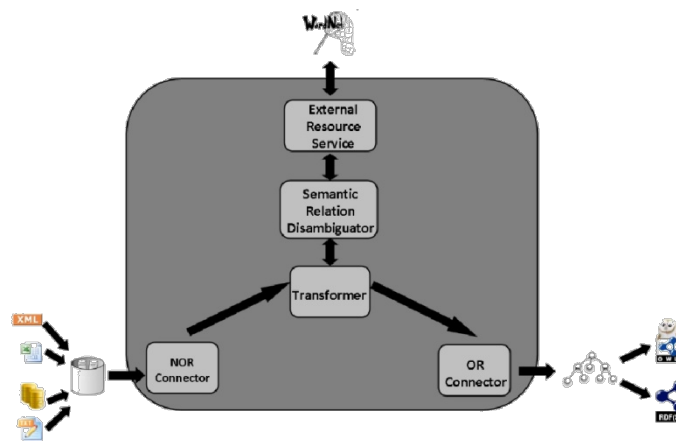


Figure 4 NOR2O Modules

available in Oracle and mySQL spatial extensions to these databases, to RDF instances. geometry2rdf defines a set of RDF triples for geometrical information (which could be available in GML or WKT). geometry2rdf can be currently downloaded at <http://mccarthy.dia.fi.upm.es/geometry2rdf/>.

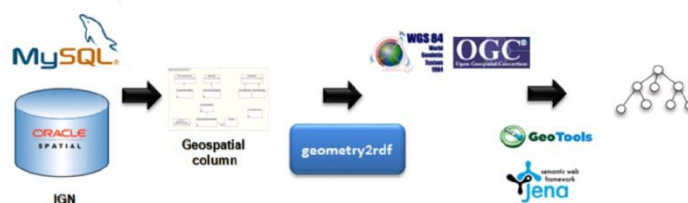


Figure 5 Geometry2RDF Modules

Mapping-based provisioning tools

Another group of RDF provisioning tools are those that use mappings to define the relationships between the source schema and the set of ontologies according to which the RDF data will be generated. The most popular use for these tools is to define mappings between relational databases and ontologies, although recent work has adopted this approach for mapping from sensor schemas to ontologies as well as described in Deliverable D1.1.

Mapping-based provisioning tools for relational data

Some research work[10, 6] have been done in recent years in the context of providing surveys of tools that transform relational data into RDF using map-pings. Hence we will not provide a deep discussion in this document about these tools. This type of tools is very important since most of the data that can be transformed into RDF reside in relational database management systems (DBMSs) and in many cases users have to define manually mappings between the relational schema to ontology elements.

Two of these tools, D2R[4] and ODEMapster[3], have been developed by PlanetData partner. D2R and ODEMapster are part of PlanetData Lab and can be downloaded at <http://www4.wiwiw.fu-berlin.de/bizer/d2r-server/> and <http://neon-toolkit.org/wiki/ODEMapster> respectively.

The aforementioned surveys classify and categorize these tools in terms of their mapping language, ontology language, data access, query translation, commercial status, and many others. It is worth noting that the W3C RDB2RDF Working Group is currently putting effort in order to have a standard mapping language of this kind of tools, named R2RML (<http://www.w3.org/TR/r2rml/>).

Mapping-based provisioning tools from sensor data

In the context of Deliverable D1.1, we have reported our work on the transformation of sensor data into RDF using mappings. More specifically, two approaches have been explored and described. The first one is transforming GSN sensor data into RDF instances and taking into account its metadata repository when performing queries. The second one is transforming Pachube sensor output into RDF instances and annotating its sensors metadata with external knowledge, such as DBpedia and Cyc. We refer the reader to that deliverable in order to get more details of these approaches. These tools will be also included in the PlanetData Lab in the following period.

MappingPedia: exploiting mappings from data provisioning tools

Purpose

One of the goals of PlanetData WP5 is to identify missing functionalities in current tools. In this section, we provide our analysis of the missing pieces in the mapping-based RDF data provisioning tools and report our work. In order to do so, we will first discuss the need of organizations for mappings in the context of a data integration system.

In distributed and open settings such as those that can be found on the Web of Data, data sources may be available in an RDF format or not, and may belong to the organization making use of it or not. In this context, each data source may have its own schema and integrity constraints, and the use of multiple data sources introduces problems related to the integration of the data according to these different schemas. Data integration approaches have introduced global schemas(which may be represented as ontologies) as single views over these heterogeneous data sources and mappings between the source schema of each of the sources and the global schema.

It is obvious that both global and source schemas contain domain knowledge. The source schemas reflect the current structure of the data while the global schema reflects the ideal view of the data. Linking those schemas, the mappings to some extent contain domain knowledge as well and the ability to analyze mappings is very important. However, as it has normally happened in the use of data integration, the mappings are often considered less important than the schemas and rarely stored or collected.

We propose the generation of a mapping repository for these types of map-pings, so that this knowledge can be maintained and exploited later. A mapping repository might contribute the following benefits

- Mapping pattern discovery. A mapping document may contain mapping patterns, which are commonly used mapping expressions. Some examples of mapping patterns are: mappings to a popular ontology concept (such as foaf:Person), mappings from self-join tables, mappings with

transformation expressions for URI generation and so on. A mapping repository hosting multiple mapping documents can be seen as a library of mapping patterns. In the same way people go to a library to find some books they could go to a mapping repository to find mappings and study how the mappings are used.

- Mapping collection based query answering. Ontology elements may be mapped to several mappings, for example an ontology concept City might be mapped in a mapping document MD1:politics and another concept River might be in a mapping document MD2:Geography. Furthermore, MD1 might have different mapping language with MD2. Thus, a mapping repository is needed in order to answer queries involving multiple concepts defined in several documents.
- Mappings evolution. As the global schema and the source schema evolve, the original mappings may not be used anymore, as they may produce errors either syntactically, semantically or pragmatically. Syntactic errors may occur due to the mappings defined from/to not existing schema components. Semantics errors may occur when some restrictions (such as primary keys on database columns) are dropped or moved to other columns. Pragmatic errors may occur due to the introduction of new results produced that is different to the expected result by the time that the mappings were created.
- Mapping suggestions. A collection of mappings can be analyzed, for example, to discover common patterns of a certain mappings. The result of this analysis might provide suggestions to the user of what other mappings should be created/used. For example, the creation of river or lake concepts might be suggested to a user creating a mapping of sea concept.
- Mapping translation. Global schema may be used by multiple users, possibly speaking different languages. It would be more convenient if a collection of mappings can be used to assist the creation of new mappings. For example, when English ontologies has been mapped to a certain database schema, then Spanish users might use Spanish ontologies and get assistance of what mappings to be created based on the existing mappings. The same situation might happen on the reverse, when dealing with same ontologies, but different natural language of database schema.

Vocabulary

The first step to be done in order to realize the mapping repository is the creation of common vocabulary to describe those mappings independently of the mapping language in which they are available. This vocabulary enables users to query a collection of mapping documents, such as finding certain usages of the mappings based on their database pattern, common use of how to generate URI, and so on. Some of the important concepts of this mapping vocabulary are:

Mapping Document Related Elements. The central concept of this Ontology is the MappingDocument class and it can be seen on Figure 5. It is used to describe a document that specifies the ontology schema(Ontology), the database schema(DataSource), and a set of mappings between them (MapSet). The Ontology class represents the ontology and it will be aligned to the Ontology class defined in Ontology Metadata Vocabulary(OMV), as OMV deals with the evolution and management of ontologies. DataSource class represents an abstraction of the database. This abstraction is made for the future work of the ontology, as to provide a means to make mappings with different type of datasource, for example xml or spreadsheet documents. For the time being, the only defined data-source type is relational database, which is represented as Database class. Several properties are attached to MappingDocument class, such as

- Properties related to the maintenance of mapping documents. Some examples of these properties are used for identifying the identifier of the mapping documents(hasMappingDocumentURI), keeping track the version of the mapping document(hasMappingDocumentVersion), describing the mapping language used(hasMappingLanguage), describing the purpose and tasks related to the mapping documents (hasMappingPurpose and hasMappingTask).

- Properties that enable the exploitation of a collection of mapping documents. hasMappingDocumentTag, hasRelatedMappingDocument.
- Properties related to the provenance of mapping documents. hasCreator, hasOrganization.

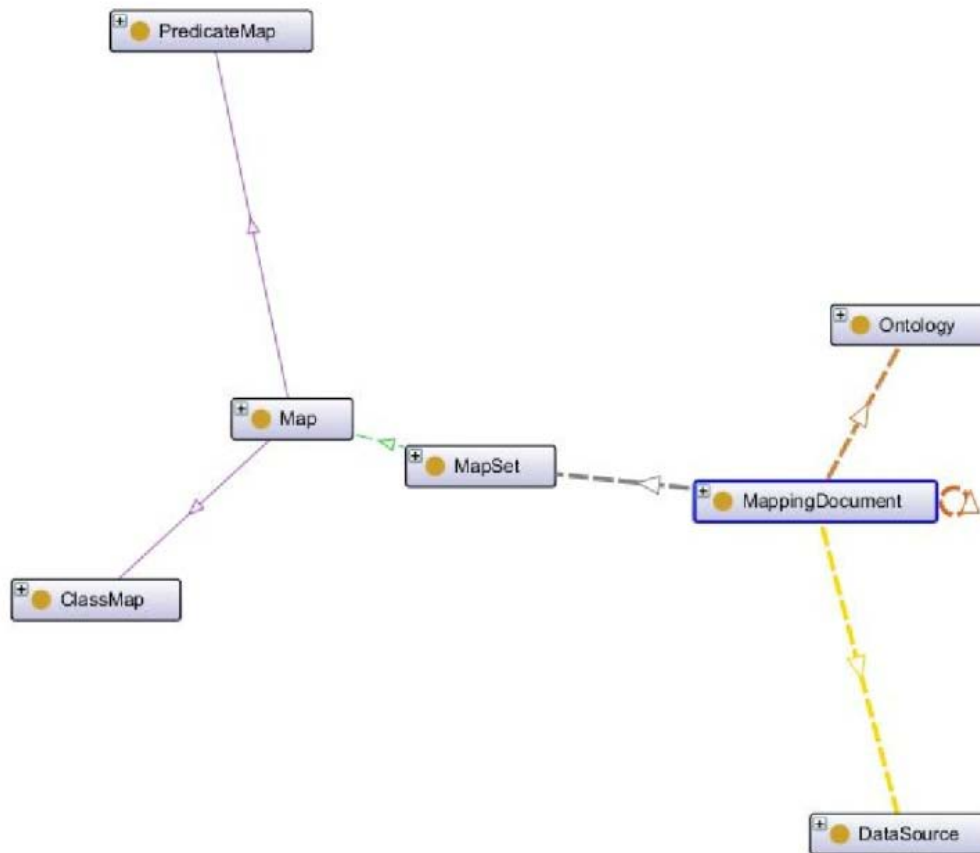


Figure 6 MappingPedia Overview

Mapping Related Elements. A MapSet is a set of Map instances, which can be seen on Figure 6. A Map acts as an superclass of Class Mapping (ClassMap) and Predicate Mapping (PredicateMap). This class contains the identification of the mapping (hasMapID) and mapping collection specific properties (hasMappingTag)

Class Mapping Related Elements. A Class Mapping (ClassMap), as shown on Figure 7, is defined as a subclass of Map and is used to generate the `rdf:type` triples of resources. The type of the subject is either a blank node or a resource, and it is specified through the property `hasSubjectType`. The expression used to generate the subjects URIs of the triples is specified through the property `hasSubjectURI`. The range of the triples, which is the class to be mapped, is specified through predicate `hasMappedClass`. The relational data that serves as the source of class mapping is represented as `ClassMapSource` and is connected through `hasClassMapSource` property. The source can be in the form of SQL string, a database table or a database view, and each of them is specified

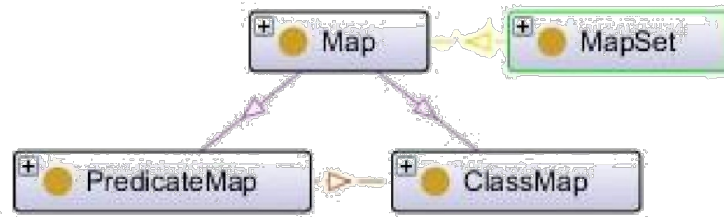


Figure 7 MappingPedia Map element

as a subclass of ClassMapSource. A class ConditionalExpression can be attached through property

hasClassMapCondition, which purpose is to specify under which condition a class mapping can be evaluated.

Predicate Mapping Related Elements. A Predicate Mapping (PredicateMap), which is shown on Figure 8, is defined as a subclass of Map and is used to generate triples statements with either a datatype property or object property. The property isIn points to the concept mapping in which the predicate mapping belongs to. A property mapping generates a set of triples in the form subject predicate object. The subject is the result of corresponding class mapping containing the predicate mapping. The predicate is the property to be mapped, and it is defined through property hasMappedPredicate. The source of object value is defined with property hasPredicateMapSource, which can be obtained from database column or from constant value. The type of the object can be literal (its datatype can be detected from the database column, ontology element, or manually defined as xsd datatype), or as a resource with URI. A property hasPredicateMapConditionAction specifies under which condition and what transformation value will be evaluated on the predicate mapping.

Condition Action and Expression Related Elements. A ConditionAction class, which can be seen on Figure 9, is used to represent if-then like situation. Instances of this class can be used on several usages, for example, when specifying the URI generation of a subject, when specifying a condition to be true when evaluation class or predicate mapping, and so on. The property hasCondition, whose range is ConditionalExpression class, is used to represent the condition part. The property hasAction, whose range is TransformationExpression, is used to specify the Action part of this class. Both ConditionalExpression and TransformationExpression are subclasses of Expression class, which is a representation of an expression, defined as an operator together with its operands. Common usages of ConditionAction, such as concatenation of several parameters, conjunctive only conditional expression or a translation table can be represented

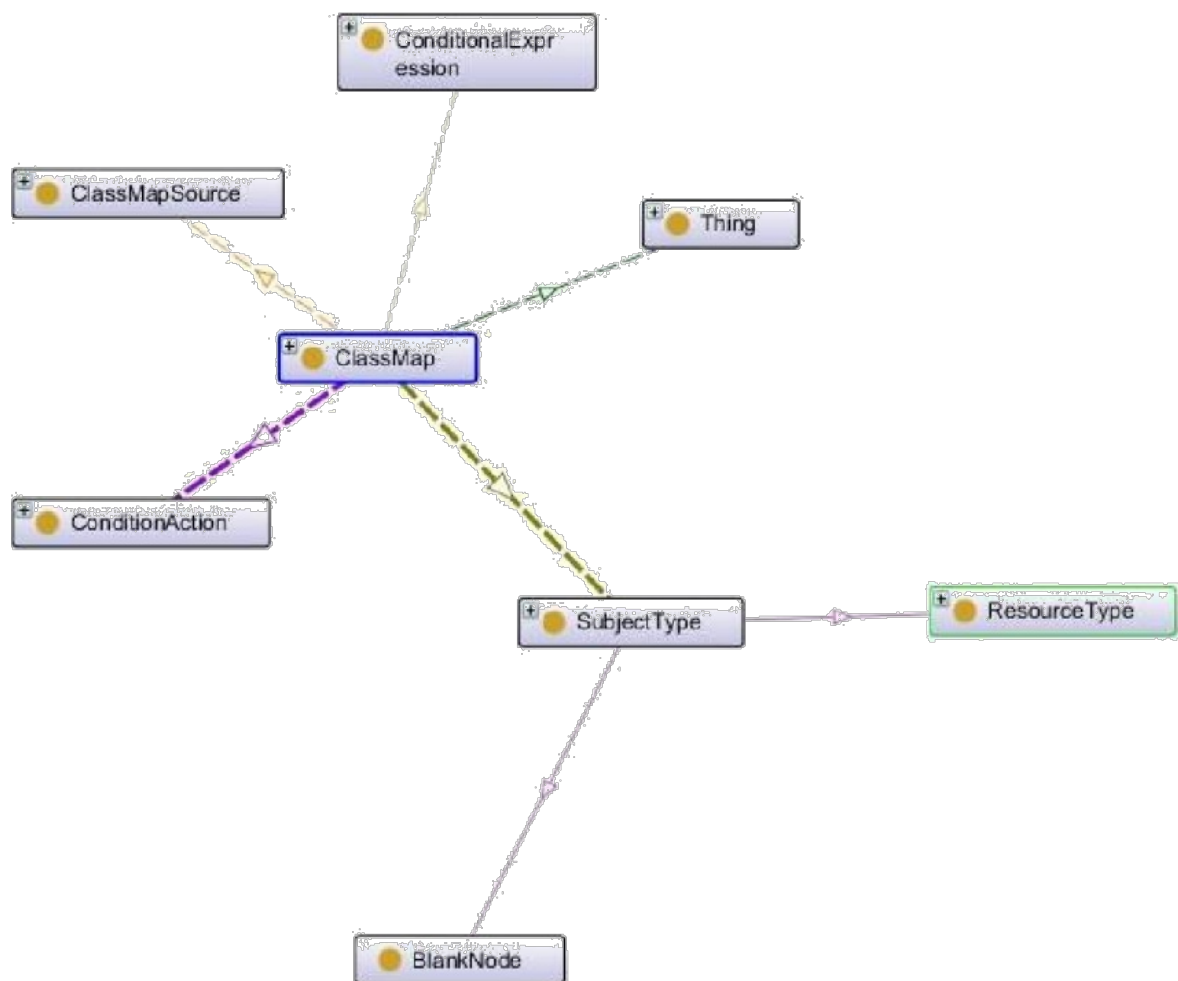


Figure 8 MappingPedia ClassMap element

as a rule pattern(RulePattern). Some predefined patterns have been defined, such as patterns involving only conditional expression without modifying values are represented as ConditionalPattern, patterns involving value transformation only without specifying the condition are represented as TransformationPattern, patterns that act as lookup tables, such as month numbers to month names, are represented as TranslationTable.

Current status and future work

During this period, we have implemented an ingestion tool for R2O mapping documents, which produces an instantiation of the ontology above for each R2O mapping document provided to it in XML format. A triple store has been set up for hosting our R2O mapping documents collection, which has currently 24 mappings, mostly related to geographical domain. These mappings were collected through a public request on Semantic Web mailing lists. The corresponding SPARQL endpoint can be accessed at this URL:

<http://mappingpedia.linkeddata.es/sparql>.

The following additional work will be done in the following periods:

More ingestion tools. The ingestion tool should be extended to accept D2R and R2RML documents, at least, although we will also analyze the

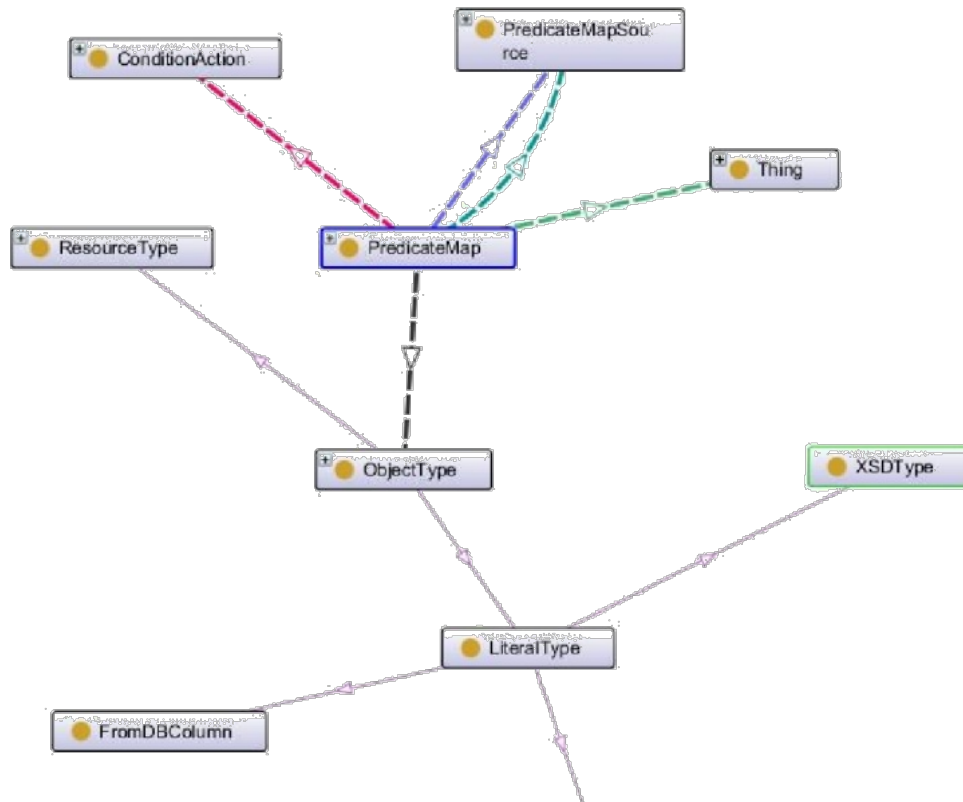


Figure 9 MappingPedia PredicateMap element

possibility of adding NOR2O configuration documents and other similar languages.

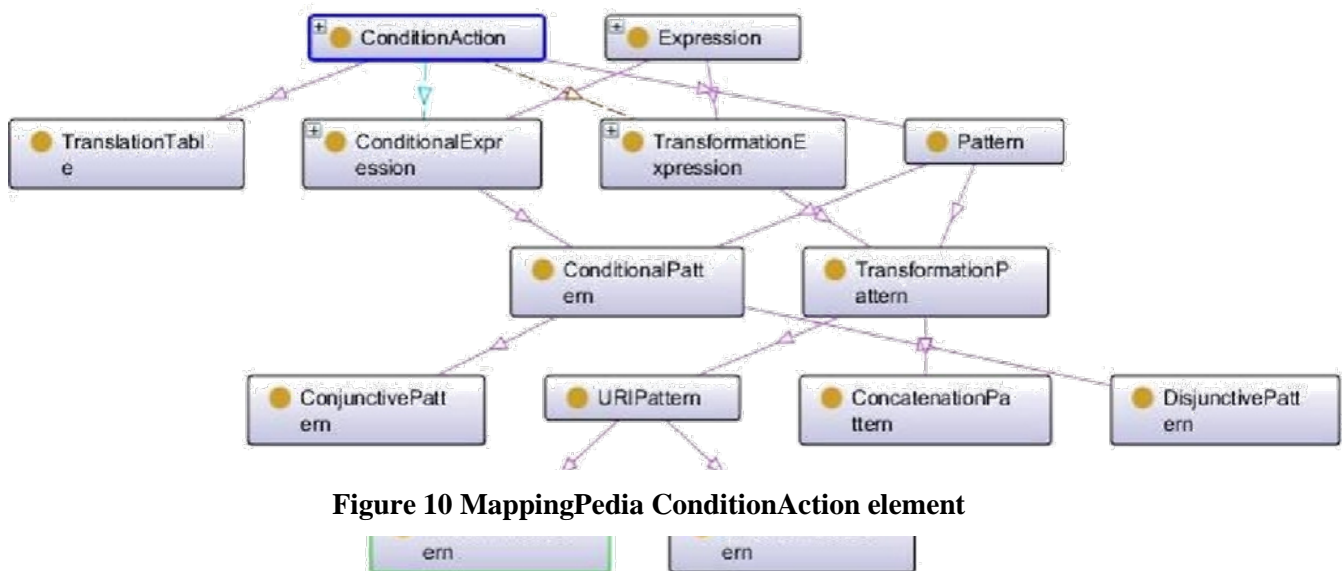
Web-based system and service. The current system works as a Java command-line system and this is not user friendly. A website that allows users to upload their mappings and transform them and load them to the mapping repository in a easy manner is a must.

Additional and sustained mapping collection. UPM has made a public request for mappings in the Linked Data and Semantic Web community through maillist list, although the result was limited. We will continue with this effort.

Refine the mapping vocabulary. The vocabulary can be improved, especially on the mapping part. One strong candidate here is to reuse the R2RML vocabulary for the mapping part.

Conclusion

In this part of the deliverable we have focused on two main activities. On the one hand, we have made a brief analysis of the two types of data provisioning



tools that can be identified in the state of the art (direct mapping and mapping-based tools), providing references to survey papers that describe and compare them, and providing some examples of those tools that will be made available in the PlanetData Lab. On the other hand, we have described our proposal for the collection of mappings used in the mapping-based approaches, and which has started with the generation of a metadata vocabulary to describe and store them, and with the creation of a tool for the ingestion of one type of mapping documents (R2O). This repository will be enriched with new mappings and new mapping formats and we hope to make it available as an interesting resource for the community that is interested in analyzing these mappings and using them for their data integration or data ingestion processes.

5 Conclusion

PlanetData Lab provides a catalog of tools that support various tasks related to large-scale data management, with particular attention to linked and streaming data. The tools are developed or maintained by the partners of PlanetData and they also offer help (through documentation or direct support) to other PlanetData partners and to the European research community in general. The PlanetData Lab is available at

<http://www.planet-data.eu/results/data-and-toolsets>

The set of tools will continually be updated.

References

- [1] I. Astrova. Reverse engineering of relational databases to ontologies. *The Semantic Web: Research and Applications*, pages 327-341, 2004.
- [2] I. Astrova, N. Korda, and A. Kalja. Rule-based transformation of sql relational databases to owl ontologies. In *Proceedings of the 2nd International Conference on Metadata & Semantics Research*. Citeseer, 2007.
- [3] J. Barrasa, O. Corcho, and A. Gomez-Perez. R2o, an extensible and semantically based database-to-ontology mapping language. In *SWDB*, volume 3372. Citeseer, 2004.
- [4] C. Bizer and R. Cyganiak. D2r server-publishing relational databases on the semantic web. In *5th International Semantic Web Conference*, page 26, 2006.
- [5] A. Buccella, M.R. Penabad, F.R. Rodriguez, A. Farina, and A. Cechich. From relational databases to owl ontologies. In *Proceedings of 6th Russian Conference on Digital Libraries*, 2004.
- [6] Periklis Stavrou Dimitrios-Emmanuel Spanos and Nikolas Mitrou. Bringing relational databases into the semantic web: A survey. Submitted to *Semantic Web Interoperability, Usability, Applicability*, 2010.
- [7] L. Han, T. Finin, C. Parr, J. Sachs, and A. Joshi. Rdf123: from spreadsheets to rdf. *The Semantic Web-ISWC 2008*, pages 451-466, 2008.
- [8] A. Langegger and W. W"o . Xlwrap{querying and integrating arbitrary spreadsheets with sparql. *The Semantic Web-ISWC 2009*, pages 359-374, 2009.
- [9] M. Li, X.Y. Du, and S. Wang. Learning ontology from relational database. In
- [10] *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, volume 6, pages 3410-3415. IEEE, 2005.
- [11] S.S. Sahoo, W. Halb, S. Hellmann, K. Idehen, T. Thibodeau Jr, S. Auer, J. Sequeda, and A. Ezzat. A survey of current approaches for mapping of relational databases to rdf. *W3C RDB2RDF XG Incubator Report*, 2009.
- [12] J.F. Sequeda, R. Depena, and D.P. Miranker. Ultrawrap: Using sql views for rdb2rdf. *Proc. of ISWC2009*.
- [13] J.F. SEQUEDA, S.H. TIRMIZI, O. CORCHO, and D.P. MIRANKER. Survey of directly mapping sql databases to the semantic web.
- [14] G. Shen, Z. Huang, X. Zhu, and X. Zhao. Research on the rules of mapping from relational model to owl. In *Proceedings of the OWLED*, volume 6, pages 21-29, 2006.
- [15] L. Stojanovic, N. Stojanovic, and R. Volz. Migrating data-intensive web sites into the semantic web. In *Proceedings of the 2002 ACM symposium on Applied computing*, pages 1100-1107. ACM, 2002.
- [16] B. Villazon-Terrazas, A. Gomez-Perez, and J.P. Calbimonte. Nor2o: a library for transforming non-ontological resources to ontologies. 2010.