PlanetData

Network of Excellence

FP7 – 257641

D5.3 PlanetData data management tools catalogue and access portal

Coordinator: Nguyen Quoc Viet Hung (EPFL)

With contributions from: INRIA, JSI, UPM, FUB, SOTON

1st Quality Reviewer: Oscar Corcho
2nd Quality Reviewer: Steffen Stadtmüller

Deliverable nature:	Report (R)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	M30
Actual delivery date:	M30
Version:	1.0
Total number of pages:	25
Keywords:	tool catalogue, linked data, sensor data, ADMS

Abstract

One of the objectives of PlanetData will be the generation of a catalogue of tools and services that can be applied in the context of Big/Open/Linked Data. This deliverable provides information about the tools which have been collected from PlanetData core and associated members. The tool catalogue will be organized and published as a part of the PlanetData portal in WP7. More precisely, we describe metadata of each tool as well as information about the tools according to this metadata. This metadata is largely based on the ADMS vocabulary (`http://www.w3.org/ns/adms`) that is being proposed in the context of the EU JoinUp initative. The reason we opt for this vocabulary is that it allows to generate not only the HTML-based version of the catalogue, but also an RDF-based version. The details of these tools will be revised and updated on a regular basis.

Executive Summary

In this deliverable, we have collected 20 tools in total and described 9 classes of metadata (according to ADMS vocabularies): general information, language, documentation, contact, status, item, distribution, license, and publisher. These tools have been collected from PlanetData core and associated members through a Google Docs form. After collecting the information, we eliminate some errors and inconsistencies to organize a consistent version of tool catalogue. The current list of tools include:

- GSN, OKKAM (EPFL)
- Datalift (INRIA)
- LDIF, D2RQ (University of Mannheim)
- Yet Another SPARQL GUI (VU University)
- ODEMapster, morph-streams, geometry2rdf, OOPS! (UPM)
- Rhizomer (Universitat de Lleida)
- IJS Newsfeed, Videk, LODMiner (JSI)
- CKANext-SILK, ckanext-extractor, ckanext-sparql, ckanext-metadata (Universidad de Deusto)
- HDT (University of Valladolid)

Note that this report only serves as a summary of up-to-date tools. We plan to put systematically full information of these tools in the disemmination platform of WP7.

Document Information

IST Project Number	FP7 – 257641	Acronym	PlanetData
Full Title	PlanetData		
Project URL	http://www.planet-data.eu/		
Document URL	provide the URI for the document		
EU Project Officer	Leonhard Maqua		

Deliverable	Number	D5.3	Title	PlanetData data management tools catalogue and access portal
Work Package	Number	WP5	Title	PlanetData Lab

Date of Delivery	Contractual	M30	Actual	M30
Status	version 1.0		final ☐	
Nature	Report (R) ☒ Prototype (P) ☐ Demonstrator (D) ☐ Other (O) ☐			
Dissemination Level	Public (PU) ☒ Restricted to group (RE) ☐ Restricted to programme (PP) ☐ Consortium (CO) ☐			

Authors (Partner)	Nguyen Quoc Viet Hung (EPFL)			
Responsible Author	Name	Nguyen Quoc Viet Hung	E-mail	quocviethung.nguyen@epfl.ch
	Partner	EPFL	Phone	+41216937573

Abstract (for dissemination)	One of the objectives of PlanetData will be the generation of a catalogue of tools and services that can be applied in the context of Big/Open/Linked Data. This deliverable provides information about the tools which have been collected from PlanetData core and associated members. The tool catalogue will be organized and published as a part of the PlanetData portal in WP7. More precisely, we describe metadata of each tool as well as information about the tools according to this metadata. This metadata is largely based on the ADMS vocabulary (`http://www.w3.org/ns/adms`) that is being proposed in the context of the EU JoinUp initative. The reason we opt for this vocabulary is that it allows to generate not only the HTML-based version of the catalogue, but also an RDF-based version. The details of these tools will be revised and updated on a regular basis.
Keywords	tool catalogue, linked data, sensor data, ADMS

Version Log			
Issue Date	Rev. No.	Author	Change
17/02/2013	0.1	Nguyen Quoc Viet Hung	First version
01/03/2013	0.2	Nguyen Quoc Viet Hung	Added metadata
28/03/2013	1.0	Nguyen Quoc Viet Hung	Added information of all tools
05/04/2013	1.1	Nguyen Quoc Viet Hung	Incorporated with reviewers' comments

Table of Contents

Abbreviation

ADMS	Asset Description Metadata Schema
DCMI	Dublin Core Metadata Initiative
GML	Geography Markup Language
WKT	Well-known text

1 Introduction

As the number of data are likely to grow at large, data management tools has become essential for the management of systems and infrastructure. In the context of PlanetData visions, we collect several tools and organize them in such a way that the partners can use and collaborate under the same environment. In order to index, search, and browse the information about collected tools, there is a need to design the metadata to construct a tool catalogue. All collected tools will be categorized according to the proposed metadata and their information will be published on the dissemination platform, which will be developed in WP7.

In this report, we use the ADMS vocabulary[1], which is being proposed in the context of the EU JoinUp initative. We extend this vocabulary for the metadata as well as provide the actual list of tools according to this metadata. The rationale behind this choice is that it allows to generate not only the HTML-based version of the catalogue, but also an RDF-based version.

More precisely, this deliverable describes a catalogue of tools for large-scale data management. We categorize each tool according to the proposed metadata, helping PlanetData partners in the use of listed tools by explicit links to documentation and use cases. In comparison with the previous version of tool catalogue in D5.1, the current version in this deliverable has 20 tools where 17 of them is new and the other three have some updates. Other remaining tools in D5.1 have no update.

The deliverable is organized as follows. Chapter 2 summarizes the metadata used for tool catalogue according to ADMS vocabularies. Chapter 3 discusses the details of tool catalogue and actual information about collected tools. Currently, we provide three cateorizations in three important dimensions: by input data, by functionality, and by representation technique. Chapter 4 is dedicated to the summarization of the tools and concludes the deliverable.

[1] `http://www.w3.org/ns/adms`

2 Metadata

For the purpose of tool catalogue, we use ADMS vocabulary `http://www.w3.org/ns/adms` as the core metadata. This metadata helps to organize collected tools more logically and search them more efficiently. We opt for ADMS since it is a standard vocabulary proposed in European Commission Joinup [1] and allows to generate not only the HTML-based version of the catalogue, but also an RDF-based version.

2.1 General Information

Generation information of a tool consists of meta-attributes that describe a broad overview and general description about it. These meta-attributes respect the ADMS vocabularies, including:

1. Name (adms:Asset#ame)
2. Alternative name (adms:Asset#Alt_Name)
3. Date of creation (adms:Asset#date_creation)
4. Date of last modification (adms:Asset#date_last_modification)
5. Description (adms:Asset#description)
6. Keywords (adms:Asset#keyword)
7. Version (adms:Asset#version)
8. Version notes (adms:Asset#version_notes)

2.2 Language

This is the natural language of the tool, which is used for communication, documentation, and user interface. Technically, the range of this attribute (adms:Language) is dcterms:LinguisticSystem [2].

2.3 Documentation

This attribute specifies the class of documents that further describe a tool or give guidelines for its usage. According to ADMS, all documents must have a title in dcmi-terms [3]. There are different kinds of documents included in the documentation of a tool:

1. Homepage (adms:Documentation#homepage)
2. Main documentation (adms:Documentation#main)
3. Related documentation (adms:Documentation#related)
4. Related web page (adms:Documentation#other)

[1] `https://joinup.ec.europa.eu/`
[2] `http://www.w3.org/ns/adms`
[3] `http://dublincore.org/documents/2012/06/14/dcmi-terms/?v=term`

2.4 Contact

This attribute provides contact information of the person responsible for a developed tool, such as name, e-mail, and address. Technically, its domain value is specified by VCard [4], which is a widely used specification and can fully represent any contact information in general. A contact information includes, but not limited to, the following important attributes:

1. Name (vcard:name)
2. E-mail (vcard:email)
3. Full address (vcard:address)
4. Telephone (vcard:phone)
5. Web page (vcard:web)

2.5 Status

This attribute serves as an indicator of the development maturity of a tool or its distribution. Technically, ADMS uses skos:Concept class to fully represent this attribute. The domain value of this attribute belongs to one of following properties:

1. Completed (adms:Status#Completed)
2. Deprecated (adms:Status#Deprecated)
3. UnderDevelopment (adms:Status#UnderDevelopment)
4. Withdrawn (adms:Status#Withdrawn)

2.6 Item

An item is contained inside a tool. This could be, but not limited to a module or a component of this tool. Sometimes, an item can be extended separately and become a new tool. The meta-attributes of an item consists of:

1. Label (adms:Item#label)
2. Description (adms:Item#description)

2.7 Distribution

This attribute, in this context, represents a particular release of a tool, which can used as a fully-functional software for a particular purpose. A distribution is typically a downloadable computer file that implements the specific requirements of a tool. Each distribution is associated with only one tool. A distribution also contains some meta attributes of general information:

1. Date of creation (adms:Distribution#date_creation)
2. Date of last modification (adms:Distribution#date_last_modification)
3. Name (adms:Distribution#Name)

[4] http://www.w3.org/TR/vcard-rdf/

4. Description (adms:Distribution#Description)
5. Representation technique (adms:Distribution#Representation): the range of this attribute is fallen into one of following values: Archimate/ BPMN/ CommonLogic/ DTD/ Datalog/ Diagram/ Genericode/ Human-Language/ IDEF/ KIF/ OWL/ Prolog/ RDFSchema/ RIF/ RelaxNG/ RuleML/ SBVR/ SKOS/ SPARQL/ SPIN/ SWRL/ Schematron/ TopicMaps/ UML/ WSDL/ WSMO/ XMLSchema/ others.
6. Format (adms:Distribution#Format): indicates the technical computer format in which the distribution is available. Technically, ADMS uses dcterms:FileFormat class to fully represents this attribute.

2.8 License

This meta-attribute describes the license type of the tool. There are various types of license such as GPL, LGPL, Apache, and MIT. A license has following properties:

1. Name (adms:License#Name)
2. Description (adms:License#Description)
3. Type (adms:License#Type)

2.9 Publisher

This attribute links to the organization (or person), who makes the metadata for the tool available. The publisher is not necessary (but usually) the author of the tool. Technically, the range of this attribute (adms:metadataPublisher) is dcterms:Agent. A publisher has basic elements:

1. Name (adms:Publisher#Name)
2. Type (adms:Publisher#Type)

3 Tools

In this section, we will provide the summary of collected tools and their catalogues. First, we offer the actual list of descriptions of these tools in Section 3.1. Then, Section 3.2 proposes the tool catalogues in three important dimensions: by input data, by functionality, by representation technique. In each dimension, we consider relevant categories to reflect the properties of each tool. While these categories are sufficient for the current catalogue, we might reconsider to extend or adapt new catagories, if the number of tools grows larger.

3.1 Tool Description

In this section, we will provide the summary of metadata for each tool. For the sake of simplicity, the listings only contain the following key aspects:

- Description (description of the tool)
- Date of creation (the date of creation of the tool source code)
- Date of last modification (the date of last changes to the tool source code)
- Keywords
- Language (the natural language used in the tool)
- Status (the development status of the tool)
- Representation (the techniques or technologies used to develop the tool)
- License (the type of license needed to use it)
- Publisher (the owner of the tool or the organization who is responsible for the publishing of the tool)
- Homepage (link to the documentation pages)

The current list of tools includes:

MonetDB

Description A relational database management system for high-performance data warehouses for business intelligence and eScience. Since a few years column store technology as pioneered in MonetDB has found its way into the product offerings of all major commercial database vendors. The market for applications empowered by these techniques provide ample space for further innovation, e.g. as demonstrated by our ongoing projects. At the same time, the landscape for major innovations remain wide open. A peek preview is given in the award winning paper titled: The Researcher's Guide to the Data Deluge: Querying a Scientific Database in Just a Few Seconds.

MonetDB is actively used in our research and real life applications. Nightly builds and regression testing ensure its quality, bug tracking helps to collect experiences and feature requests. Browsing the source code repository is supported by the Mercurial web frontend. Contributions ranging from bug reports, cross-platform issues, patches and features are highly appreciated.

Keywords column-stored, XML, data management

Language English

Status UnderDevelopment

Representation Java, XML, XMLSchema

License MonetDB Public License

Publisher CWI

Homepage `http://www.monetdb.org/`

Global Sensor Networks - GSN

Description 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 built the required processing path. For example, one can imagine an anemometer that would sent 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 splited 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.

Date of creation - last modification 11/11/2004 - 10/02/2013

Keywords data stream, sensor network, distributed system

Language English

Status UnderDevelopment

Representation Java, XML, XMLSchema

License GPL

Publisher EPFL

Homepage `http://sourceforge.net/apps/trac/gsn/`

LDIF - Linked Data Integration Framework

Description 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 data sets such as DBpedia or Freebase. Linked Data applications that want to consume data from this global data space face the challenges that:

Data sources use a wide range of different RDF vocabularies to represent data about the same type of entity; The same real-world entity, for instance a person or a place, is identified with different URIs within different data sources; Data about the same real-world entity coming from different sources may contain conflicting value. For example the single value attribute population for a specific country can have multiple, different values after merging data from different sources. 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.

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 normalize URI while keeping track of data provenance.

Date of creation - last modification 6/29/2011 - 02/21/2013

Keywords linked data, data integration, schema mapping, identity resolution, data quality assessment, data fusion

Language English

Status Completed

Representation Java

License Apache

Publisher University of Mannheim and MES Semantics

Homepage `http://ldif.wbsg.de/`

Morph-Streams

Description SPARQL-Stream is a language that extends SPARQL for continuous query processing over streaming data. The Morph-streams module for SPARQL-Stream is a a java library that enables the execution of SPARQL-Stream queries, using different underlying DSMS or CEP (e.g. Esper, GSN, Cosm, SNEE, etc.). This tool allows posing SPARQL-Stream queries to an existing datasource using R2RML mappings. The mappings provide a descriptive way of relating ontological concepts (e.g. classes and properties) to elements of the DSMS or CERP schema (streams, tables). Morph uses a query rewriting approach to transform the SPARQL-Stream queries to native queries understandable and executable by the DSMS or CEP, using the R2RML mappings. Then, When morph executes the queries in the original datasources, it is capable of translating the responses to variable bindings or triples, depending on the type of query.

Date of creation - last modification 14/07/2011 - 05/03/2013

Keywords data stream, sensor network, query rewriting, sparql, query processing, rdf stream

Language English

Status UnderDevelopment

Representation Java, Scala

License GPL

Publisher UPM

Homepage `https://github.com/jpcik/morph-streams`

Videk

Description Videk is a mash-up based on several sources of data for environmental intelligence, including data coming from Smart Objects. Videk currently uses four sources of sensor and linked data and relies on StreamSense engine for storage and processing. On the server side Videk uses StreamSense, a sensor stream processing system based on tightly integrated and scalable custom software modules. StreamSense provides interfaces and means of information collection from a set of Smart Objects and generic APIs for data feeds on one hand; and interfaces to application developers on the other. Videk provides functionality such as, finding illuminance measurements around a given location or, showing all the locations in some region that measure illuminance.

Date of creation - last modification 2011 - now

Keywords Mash-up, sensors, web of things, real-time, data mining, semantic web.

Language english

Status UnderDevelopment

Representation XML

License Unknown

Publisher Unknown

Homepage `http://sensors.ijs.si/`

Datalift Platform - Datalift

Description The Datalift web platform is a tool suite for converting, structured data sources and publishing them as linked data on the web. Datalift brings raw structured data coming from various formats (relational databases, CSV, XML, ...) to semantic data interlinked on the Web of Data. Datalift is an experimental research project funded by the French national research agency. Its goal is to develop a platform to publish and interlink datasets on the Web of data. Datalift will both publish datasets coming from a network of partners and data providers and propose a set of tools for easing the datasets publication process.

Date of creation - last modification 01/10/2010 - 13/03/2013

Keywords linked-data, structured data, interlinking, LOV, vocabulary mapping, ontologies, sql, shape, statistics, sdmx, datacube, CSV, SPARQL, JSON, Java, javascript, XML, RDF

Language English, French

Status UnderDevelopment

Representation Java, XML, OWL, RDF, RDFSchema, SPARQL, SKOS, SPIN

License Apache

Publisher INRIA

Homepage `http://datalift.org`

D2RQ Plattform - Accessing Relational Databases as Virtual RDF Graphs

Description The D2RQ Platform is a system for accessing relational databases as virtual, read-only RDF graphs. It offers RDF-based access to the content of relational databases without having to replicate it into an RDF store. Using D2RQ you can:

1. query a non-RDF database using SPARQL
2. access the content of the database as Linked Data over the Web
3. create custom dumps of the database in RDF formats for loading into an RDF store
4. access information in a non-RDF database using the Apache Jena API

D2RQ is Open Source software and published under the Apache license. The source code is available on GitHub.

Date of creation - last modification 08/12/2004 - 22/06/2012

Keywords Database-to-RDF Mapping, Linked Data Publication, SPARQL-to-SQL Rewriting

Language English

Status Completed

Representation Java, RDF, SQL

License Apache license

Publisher University of Mannheim and DERi Galway

Homepage `http://d2rq.org/`

ODEMapster

Description A tool to transform database records into RDF instances using standard (R2RML) mapping language. This tool also allows to query the database using graph query language (SPARQL). ODEMapster is also developed as a plugin of the NeOn toolkit. ODEMapster plugin provides users a Graphical User Interface that allows to create, execute, or query mappings between ontologies and databases. The mappings are expressed in R2O language, which is a mapping language between ontologies and databases featuring fully declarative, DBMS independent, and extensible set of primitive operations. This plugin works with OWL/RDF(S) ontologies and with MySQL or Oracle databases. Multiple R2O mappings per ontology can be created. ODEMapster is the processor in charge of carrying out the exploitation of the mappings defined using R2O, performing both massive and query driven data upgrade.

Date of creation - last modification 01/01/2007 - 03/15/2013

Keywords rdb2rdf, r2rml, rdf, sql, rdb

Language English

Status UnderDevelopment

Representation Java, RDF, SPARQL

License N/A

Publisher Universidad Politécnica de Madrid

Homepage `https://github.com/fpriyatna/odemapster`

Rhizomer

Description Rhizomer is a faceted browser that also provides a pivoting operation that allows no-experts to build complex semantic queries. It can be deployed on top of existing stores (Virtuoso, Jena, Sesame/OWLIM) and builds a user interface that provides Information Architecture components that facilitate fulfilling typical data exploration:

- Overview: global and local navigation menus are generated based on the classes instantiated by the dataset being published and also the SKOS concepts being subjects of the dataset resources.

- Zoom and Filter: when loading the dataset for the first time, it is analysed so it is possible to generate faceted views for all classes. Facets allow filtering using common values, searching for specific facet values and pivoting. The pivot operation allows switching from a particular faceted view, e.g. directors born in New Zealand, to faceted views of the sets related to the current resource set through one of the current facets, e.g. the faceted view of the films directed by the directors born in New Zaeland.

- Details on Demand: the RDF descriptions for the selected resources are rendered using an RDF2HTML+RDFa transformation. Moreover, it is also possible to use specialised visualisations like maps or timelines.

Date of creation - last modification Unknown - 18/03/2013

Keywords user interface, exploration, browser, Linked Data, Semantic Web, visualization

Language Unknown

Status UnderDevelopment

Representation Java, XML, OWL, RDF, RDFSchema, SPARQL, SKOS

License GPL

Publisher Universitat de Lleida

Homepage `http://code.google.com/p/rhizomer/`

OKKAM - Enabling the Web of Entities

Description 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.

OKKAM will contribute 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 identifierss for naming the entities which already have a public identifier (the OKKAM's razor). Therefore, OKKAM will make 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).

Date of creation - last modification 01/01/2008 - 30/06/2010

Keywords web of entities, entity name system, identifiers

Language English

Status Completed

Representation Java

License GPL

Publisher EPFL

Homepage `http://okkam.org/`

CKANext-SILK

Description An extension for interlinking datasets uploaded on CKAN using SILK Link Discovery Framework. 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.

Date of creation - last modification 21/11/2012 - 21/02/2012

Keywords linked data, interlinking, CKAN, semantic web

Language English

Status UnderDevelopment

Representation Java

License Apache2.0

Publisher DeustoTech - Internet

Homepage `https://github.com/memaldi/ckanext-silk`

Yet Another SPARQL GUI

Description YASGUI is a web-based SPARQL client that can be used to query both remote and local endpoints. It integrates linked data services and web APIs to offer features such as autocompletion and endpoint lookup. It supports query retention - query texts persist across sessions - and query permalinks, as well as syntax checking and highlighting. Specially, YASGUI fits all requirements:

- Work on all endpoints (not just the CORS-enabled ones)
- Multi-platform (i.e. a web application)
- Easy-to-work user interface (i.e. prefix fetching, syntax highlighting/checking, storing queries)

Date of creation - last modification 01/07/2012 - 10/03/13

Keywords SPARQL, Semantic Web, Endpoints

Language English

Status Completed

Representation N/A

License MIT

Publisher

Homepage `http://laurensrietveld.nl/yasgui/`

HDT

Description HDT (Header, Dictionary, Triples) is a compact data structure and binary serialization format for RDF that keeps big datasets compressed to save space while maintaining search and browse operations without prior decompression. This makes it an ideal format for storing and sharing RDF datasets on the Web. Some facts about HDT:

- The size of the files is smaller than other RDF serialization formats. This means less bandwidth costs for the provider, but also less waiting time for the consumers to download. - The HDT file is already indexed. The users of RDF dumps want to do something useful with the data. By using HDT, they download the file and start browsing/querying in minutes, instead of wasting time using parsing and indexing tools that are difficult to setup and tune. - High performance querying. Usually the bottleneck of databases is slow disk access. The internal compression techniques of HDT allow that most part of the data (or even the whole dataset) can be kept in main memory, which is several orders of magnitude faster than disks. - Highly concurrent. HDT is read-only, so it can dispatch many queries per second using multiple threads. - The format is open and is acknowledged as W3C HDT Member Submission. This ensures that anyone on the Web can generate and consume files, or even write their own implementation. - The libraries are open source (LGPL). You can adapt the libraries to your needs, and the community can spot and fix issues.

Date of creation - last modification 30/01/2013 - 14/03/2013

Keywords Binary RDF, compression, in-memory SPARQL, fast exchange.

Language English

Status UnderDevelopment

Representation Java, RDF, SPARQL, C++

License GNU LESSER GENERAL PUBLIC LICENSE

Publisher DataWeb Research (University of Valladolid)

Homepage `http://www.rdfhdt.org/`

geometry2rdf

Description A tool that generates RDF triples from geometrical information, which can be available in GML or WKT. This will increase the possible reuse of that information. It converts the information into a format, RDF, that it's easier to consult and more reusable that the information available in the DDBB. geometry2rdf is a library for generating RDF files for geometrical information (which could be available in GML or WKT). The GML and WKT is manipulated with GeoTools. The current version of the library works with Oracle geospatial databases and relies on Jena.

Date of creation - last modification 01/02/2011 - now

Keywords geometry rdf transformation

Language English

Status Completed

Representation Java, RDF

License N/A

Publisher OEG

Homepage `http://www.oeg-upm.net/index.php/technologies/151-geometry2rdf`

SPARQL Extension for CKAN - ckanext-sparql

Description This CKAN plugin offers two main functionalities: - It allows the configuration of a SPARQL endpoint for the whole CKAN platform in which metadata about every dataset in the platform can be queried. - It allows dataset editors to configure a RDF store and manage the RDF data of the dataset directly from CKAN, enabling at the same time a SPARQL endpoint for querying this data.

Date of creation - last modification 04/03/2013 - 21/03/2013

Keywords ckan, sparql, rdf store

Language English

Status UnderDevelopment

Representation Java

License AGPL

Publisher MORElab

Homepage `https://github.com/morelab/ckanext-sparql`

SPARQL endpoint analyzer and metadata generator for CKAN - ckanext-metadata

Description SPARQL endpoint analyzer and metadata generator for CKAN. It provides automatic extraction of remote SPARQL endpoints and calculates different properties such: number of subjects, number of predicates, properties, etc.

Date of creation - last modification 12/12/2012 - 29/01/2013

Keywords metadata, ckan, extension

Language English

Status UnderDevelopment

Representation Java

License AGPL

Publisher Universidad de Deusto

Homepage `www.morelab.deusto.es`

LODMiner

Description LOD Miner is a system for recommending missing properties for a given object. The input to the system is a set of objects or entities, each described with a set of properties. The system then tries to find the missing properties for a specified object based on similarity to other objects. Examples are RDF graphs from LOD.

Date of creation - last modification 2012 - now

Keywords linked open data, graph, missing properties, prediction.

Language english

Status UnderDevelopment

Representation Java

License N/A

Publisher JSI

Homepage `http://lodminer.net/`

OOPS! - OntOlogy Pitfall Scanner!

Description OOPS! is a web-based tool, independent of any ontology development environment, for detecting potential pitfalls that could lead to modelling errors. This tool is intended to help ontology developers during the ontology validation activity, which can be divided into diagnosis and repair. Currently, OOPS! provides mechanisms to automatically detect a number of pitfalls, thus helps developers in the diagnosis activity.

Date of creation - last modification 14/11/2011 - 21/03/2013

Keywords ontology, ontology development, pitfall detection, ontology evaluation

Language English

Status UnderDevelopment

Representation Java

License GPLv3

Publisher Ontology Engineering Group

Homepage `http://www.oeg-upm.net/oops`

CKAN Extractor - ckanext-extractor

Description CKAN plugin for the automatic extraction of data sources. It enables administrator to upload transformation plugins written in Python which extract data from non-structured sources. The extension provides a common framework for transformation development and periodic execution of tasks using celery.

Date of creation - last modification 27/02/2013 - 15/03/2013

Keywords data extraction

Language English

Status UnderDevelopment

Representation Java

License AGPL

Publisher Universidad de Deusto

Homepage `https://github.com/morelab/ckanext-extractor`

IJS Newsfeed

Description Newsfeed provides a clean, continuous, real-time aggregated stream of semantically enriched news articles from RSS-enabled sites across the world.

The pipeline performs the following main steps:

1) Periodically crawl a list of RSS feeds and a subset of Google News and obtain links to news articles

2) Download the articles, taking care not to overload any of the hosting servers

3) Parse each article to obtain

3a) Potential new RSS sources mentioned in the HTML, to be used in step (1)

3b) Cleartext version of the article body

4) Process articles with Enrycher (English and Slovene only)

5) Expose two streams of news articles (cleartext and Enrycher-processed) to end users.

Date of creation - last modification 2012 - Now

Keywords text, news, data stream, enrichment

Language english

Status UnderDevelopment

Representation XML

License LGPL

Publisher JSI

Homepage `http://newsfeed.ijs.si/`

3.2 Tool Catalogue

The goal of the tool catalogue is to offer users guidance for tools for a particular application. When users have a new requirement, they can use the tool catalogue to find relevant tools in a short time. To implement this goal, we categorize collected tools according to three dimentions:

1. By Input Data: includes three main categories—Stream Data, Linked Data, Non-Structured Data.
2. By Functionality: consists of five categories—Produce, Publish, Consume, Provisioning, Data Management.

3. By Representation Technique: the techniques or technologies used to develop the tool. This categorization contains 8 categories—Java, XML, RDF, SQL, SPARQL, C++, OWL, Scala.

In fact, these categories are not disjoint; a tool might belongs to different categories. While these categories is sufficient for the current catalogue, we might reconsider to extend or adapt new catagories if the number of tools grows larger.

3.2.1 By Input Data

In this section, we categorize the tools due to their target data formats. Since data is the main concern in PlanetData project, it is important to know which tools should be used for specific datasets. Currently we use the following categories: Stream Data, Linked Data, Non-Structured Data that refers to the type of data which one can apply the given tool.

Table 3.1: Tool Categories By Input Data

Name	Stream Data	Linked Data	Non-Structured Data
MonetDB	x	x	
GSN	x		
Datalift		x	
LDIF		x	
D2RQ		x	
Yet Another SPARQL GUI		x	
ODEMapster		x	
Rhizomer		x	
IJS Newsfeed			x
Videk	x	x	
LODMiner		x	
OKKAM		x	
morph-streams	x	x	
CKANext-SILK		x	
geometry2rdf		x	
ckanext-sparql		x	
ckanext-metadata		x	
OOPS!	x		
ckanext-extractor			x
HDT		x	

Table 3.1 depicts the tool catalogue with respect to the input data of each tool. In general, most of collected tools support linked data. This is not surprising since PlanetData project aims to provide an open platform for online communities. Besides, only four tools support stream data (e.g. GSN supports sensor data). Overall, this categorization is a practical guideline for specific applications. We should focus more on stream data, especially if more datasets are available and more applications are developed for this type of data.

3.2.2 By Functionality

In this section, we categorize collected tools by their functionality. Moreover, this categorization will also identify missing functionality in currently available tools and services, which provide a gap analysis to develop or extend new functionality for specific usecases. There are 5 categories:

- Produce: tools fallen in this category have ability to generate new data.
- Publish: tools fallen in this category have ability to publish data on web platforms.
- Consume: tools fallen in this category have ability to process the data for specific applications; e.g. data mining, data compression.

- Provisioning: tools fallen in this category have ability to manipulate data such as data conversion and data extraction.
- Data Management: tools fallen in this category have ability to manage data in large-scale scenarios; e.g. storage, indexing, and querying.

Table 3.2: Tool Categories By Functionality

Name	Produce	Publish	Consume	Provisioning	Data Management
MonetDB			x		x
GSN	x	x	x		x
Datalift		x		x	
LDIF	x		x		
D2RQ			x	x	
Yet Another SPARQL GUI			x	x	
ODEMapster	x			x	
Rhizomer			x	x	
IJS Newsfeed		x			
Videk		x		x	
LODMiner			x		
OKKAM	x	x	x		
morph-streams				x	
CKANext-SILK			x	x	
geometry2rdf	x			x	
ckanext-sparql			x	x	
ckanext-metadata			x		
OOPS!			x		
ckanext-extractor			x		
HDT			x		x

Table 3.2 illustrates the tool catalogue in this dimension. Most of the tools are developed for consuming and provisioning data. Although there are few tools for the data management purpose, they have large impacts in data communities. For example, MonetDB is used worldwide for education, research, and businesses. It has more than 30 years of development and experiences about practial scenarios.

3.2.3 By Representation Technique

In this section, we categorize the tools by their representation technique. It is important to know which technologies are used to develop a given tool. This helps users to know the compability between new and existing tools when they are integrated in the same platform. More precisely, we consider eight techniques: Java, XML, RDF, SQL, SPARQL, C++, OWL, Scala.

Table 3.3 presents the tool in terms of reprentation technique. Most of the tools use Java and RDF. While Java is a cross-platform programming language, RDF is a general language for conceptual description of information in web resources. They allow high interoperability between different platforms, which is a good sign for future experiments of PlatnetData project.

Table 3.3: Tool Categories By Representation Technique

Name	Java	XML	RDF	SQL	SPARQL	C++	OWL	Scala
MonetDB	x	x	x	x	x			
GSN	x	x	x			x		
Datalift	x	x	x		x	x	x	
LDIF	x							
D2RQ	x		x	x				
Yet Another SPARQL GUI	x							
ODEMapster	x		x		x			
Rhizomer	x	x	x	x	x	x	x	
IJS Newsfeed		x						
Videk		x						
LODMiner	x							
OKKAM	x	x	x					
morph-streams	x							x
CKANext-SILK	x		x					
geometry2rdf	x		x					
ckanext-sparql	x		x		x			
ckanext-metadata	x		x					
OOPS!	x		x				x	
ckanext-extractor	x		x					
HDT	x		x		x	x		

4 Conclusions

In this deliverable, we have collected 20 tools in total and described 9 classes of metadata (according to ADMS vocabularies): general information, language, documentation, contact, status, item, distribution, license, and publisher. Moreover, they are classified into three main catalogue according to three important dimensions: by input data, by functionality, and by representation technique). Table 4.1 summarizes collected tools in 7 main meta-attributes: name, keywords, language, status, representation technique, license, and publisher.

In summary, all the tools support various technologies and functionalities related to large-scale data management, with particular focuses on stream data, linked data, and non-structured data. The collected tools are developed and maintained by many partners of PlanetData project. Overall, this tool catalogue will provide fundamental information in the vision of building PlanetData dissemination platform in WP7. Not only PlanetData partners receive benefit from published documentation and direct support, but also European research community might be supported for its future collaboration.

Note that this report only serves as a summary of up-to-date tools. We plan to put systematically full information of these tools in the disemmination platform of WP7. As the tool catalogue will be publicly available, we expect that the categorizations presented in this report will be refined and improved by the research community, in particular when more data become available, more experiments are performed, and more tools are integrated into the PlanetData project in the future.

Table 4.1: Summary of Tool Catalogue

Name	Keywords	Language	Status	Representation technique	License	Publisher
GSN	data stream, sensor network, distributed system	English	UnderDevelopment	Java, XML, XMLSchema	GPL	EPFL
Datalift	linked-data, structured data, interlinking, LOV, vocabulary mapping, ontologies, sql, shape, statistics, sdmx, datacube, CSV, SPARQL, JSON, Java, javascript, XML, RDF	English, French	UnderDevelopment	Java, XML, OWL, RDF, RDF-Schema, SPARQL, SKOS, SPIN	Apache	INRIA
LDIF	linked data, data integration, schema mapping, identity resolution, data quality assessment, data fusion	English	Completed	Java	Apache	University of Mannheim and MES Semantics
D2RQ	Database-to-RDF Mapping, Linked Data Publication, SPARQL-to-SQL Rewriting	English	Completed	Java, RDF, SQL	Apache license	University of Mannheim and DERi Galway
Yet Another SPARQL GUI	SPARQL, Semantic Web, Endpoints	English	Completed		MIT	
Sciencea	european projects, r&d, analytics, science, consortium, search partners, compare territories, state of the art, EU programs, open calls.	English	Completed	Java, DEX graph database		Cordis, DBLP
ODEMapster	rdb2rdf, r2rml, rdf, sql, rdb	English	UnderDevelopment	Java, RDF, SPARQL		Universidad Politécnica de Madrid
Rhizomer	user interface, exploration, browser, Linked Data, Semantic Web, visualization		UnderDevelopment	Java, XML, OWL, RDF, RDF-Schema, SPARQL, SKOS	GPL	Universitat de Lleida
IJS Newsfeed	text, news, data stream, enrichment	english		XML		JSI
Videk	"Mash-up, sensors, web of things, real-time, data mining, semantic web."	english		XML		
LODMiner	linked open data, graph, missing properties, prediction.	english	UnderDevelopment			JSI
OKKAM	web of entities, entity name system, identifiers	English	Completed			
morph-streams	data stream, sensor network, query rewriting, sparql, query processing, rdf stream	English	UnderDevelopment	Java, Scala	GPL	UPM
CKANext-SILK	linked data, interlinking, CKAN, semantic web	English	UnderDevelopment		Apache2.0	DeustoTech - Internet
geometry2rdf	geometry rdf transformation	English	Completed	Java, RDF		OEG
ckanext-sparql	ckan, sparql, rdf store	English	UnderDevelopment		AGPL	MORElab
ckanext-metadata	metadata, ckan, extension		UnderDevelopment	Java	AGPL	
OOPS!	ontology, ontology development, pitfall detection, ontology evaluation	English	UnderDevelopment		GPLv3	Ontology Engineering Group
ckanext-extractor			UnderDevelopment		AGPL	
HDT	Binary RDF, compression, in-memory SPARQL, fast exchange.	English	UnderDevelopment	Java, RDF, SPARQL, C++	GNU LESSER GENERAL PUBLIC LICENSE	DataWeb Research (University of Valladolid)