Supporting the Linked Data Life Cycle Using an Integrated Tool Stack

Bert Van Nuffelen[1(✉)], Valentina Janev[2], Michael Martin[3], Vuk Mijovic[2], and Sebastian Tramp[3]

[1] TenForce, Leuven, Belgium
bert.van.nuffelen@tenforce.com
[2] Institute Mihajlo Pupin, Pupin, Serbia
{valentina.janev,vuk.mijovic}@pupin.rs
[3] University of Leipzig, Leipzig, Germany
{martin,tramp}@informatik.uni-leipzig.de

Abstract. The core of a Linked Data application is the processing of the knowledge expressed as Linked Data. Therefore the creation, management, curation and publication of Linked Data are critical aspects for an application's success. For all of these aspects the LOD2 project provides components. These components have been collected and placed under one distribution umbrella: the LOD2 stack. In this chapter we will introduce this component stack. We will show how to get access; which component covers which aspect of the Linked Data life cycle and how using the stack eases the access to Linked Data management tools. Furthermore we will elaborate how the stack can be used to support a knowledge domain. The illustrated domain is statistical data.

1 Introduction

Publishing Linked Data requires the existence of management processes that ensure the quality. The management process passes through several stages; in the Linked Data life cycle the main stages are ordered in their typical application order. The starting point is very often the extraction stage in which data from the source format is turned into RDF. The extracted RDF formatted data must be stored in an appropriate storage medium, making the data available for further processing. At this moment the data is ready to be queried and can be manually updated to correct small mistakes. Within the linking stage the data is enriched by interconnecting the data with external data sources. These data linkages create new opportunities: the data can now be classified according to the external data; information that is spread over two entities can be fused together, ... All these data manipulations can be monitored with quality metrics. When the desired data quality is reached the data can be made public and be explored by end-user applications.

Of-course the world is ever changing and hence data will reflect this. Therefore, there is support for the evolution of the data from one structure into another.

S. Auer et al. (Eds.): Linked Open Data, LNCS 8661, pp. 108–129, 2014.
DOI: 10.1007/978-3-319-09846-3_6

For all these stages research institutes and companies around the world have created tools. At the start of the LOD2 project these tools were scattered around the Linked Data community. Specialists in the area shared lists of components in various degree of completeness. The LOD2 project had the ambition to start a platform in which all Linked Data components were collected. This common distribution platform was called the LOD2 stack, and will continue to exist after the LOD2 project has finished as the Linked Data stack[1]. Components in the stack are easy to install and directly usable. Moreover, they come with pre-configured setups that make the interplay between them easier. These additional features cannot be offered by the individual component owners but requires central coordination.

In the first part of this chapter, the LOD2 stack is elaborated in more detail. The second part is dedicated to the specialization of the LOD2 stack for statistical data. Indeed the LOD2 stack is in its own right is not dedicated towards a particular use case. For particular kinds of data, such as statistical data, the components of the stack can be further specialized and pre-configured to offer a much better dedicated end user support.

2 The LOD2 Linked Data Stack

The LOD2 Linked Data stack is a distribution platform for software components which support one or more aspects of the Linked Data life cycle. Each package contains a pre-configured component that on installation results in a ready-to-use application. The pre-configuration ensures that the deployed components are able to interact with each other. The system architecture of the deployed Linked data stack components is explained in Sect. 2.1. The subsequent sections provide more details on the distribution platform and what the requirements are for software to take part of it. Finally we provide an overview of the LOD2 stack contents in Sect. 2.5.

2.1 Building a Linked Data Application

The LOD2 stack facilitates the creation of Linked Data applications according to a prototypical system architecture. The system architecture is shown below in Fig. 1. From top to bottom, one has first the application layer with which the end-user is confronted. The applications are built with components from the component layer. These components communicate between each other and interact with the data in the data layer via the common data access layer.

The data access layer is build around the data representation framework RDF. The data is exchanged in RDF and retrieved with SPARQL queries from SPARQL end-points. All data format heterogeneity is hidden for the components by this layer. This yields an uniform data view easing the configuration of the data flow between the components. The RDF representation yields important

[1] http://stack.linkeddata.org

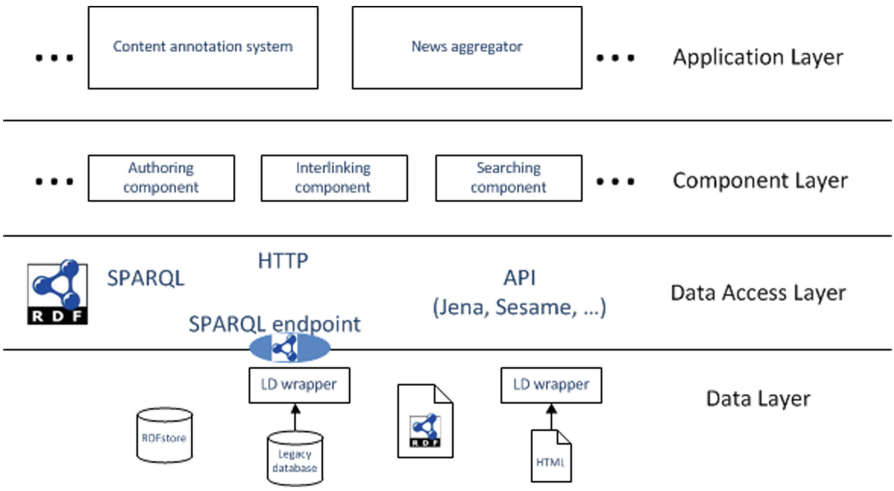

Fig. 1. Linked Data application system architecture

advantages making it suited for the role as common data representation formalism. It is a W3C standard, domain neutral, and it is web enabled: all identifiers are web addressable. And last but not least, data integration starts with just merging the data together in one store.

Matching the Linked Data life cycle presented in the introduction to the system architecture shows that the extraction and storage tools feature in the data layer and most of the other are part of the component layer.

An application end user will seldom be directly in touch with the underlying data layer. They are offered an application interface that shows the information in a domain adapted intuitive interface. Few of the LOD2 stack browsing and exploration components have been lifted and designed to this stage. Most of the stack components are designed for the (Linked Data) data manager. The components provide user interfaces that aid the data manager in its task. For instance, the SILK workbench is a user interface for creating linking specifications. This specification can then be used by the silk engine which might be embedded in a larger application. That is the task of the last targeted audience: the Linked Data application developer.

2.2 Becoming LOD2 Linked Data Stack Component

The system architecture defines minimal requirements for components to become part of the LOD2 stack, they are that

- Information is exchanged in RDF format
- Business information is requested through SPARQL endpoint access
- Updates are provided as SPARQL updates

A typical example is the SILK workbench. The business data is retrieved via querying SPARQL endpoints and the result (the link set) can be stored in a SPARQL endpoint that is open to updates. Of course these requirements do not hold for all communication channels of a component, for instance, extraction components such as D2R. Those obviously have to communicate with the original source of the information which is different than RDF, but the outcome is RDF.

A component is distributable via the LOD2 stack repository if it is provided as a Debian package that installs on Ubuntu 12.04 LTS. The Debian packaging system has been chosen because it is a well established software packaging system that is used by the popular Linux distribution Ubuntu. We decided for a reference OS release to ensure quality and reduce the maintenance efforts of the components. This choice does not limit the deployment of the software on other Linux distributions, whether they are Debian based or not. Using a tool called alien, debian packages can be installed on a RedHat distribution.

The above mentioned requirements ensure there is a common foundation between all components in the LOD2 stack. For application developers trained in the Linked Data domain the creation of an information flow is hence always possible. Because each data publishing editorial process is unique and the best solution depends on the data publishing organizations needs, the stack does not aim for a homogeneous LOD2 Linked Data publishing platform where all components are hidden behind a single consistent user interface. The goal is however on improving the information flow between components. In the first place, this is done via making sure that deployed components have access to each others output. Additionally, LOD2 has contributed supporting APIs and vocabularies to the component owners. If they extend their components with these, the data flow between the components will be further improved.

Using this approach, each component still has its individual identity but they are interoperable with each other.

2.3 LOD2 Stack Repository

In order to guarantee stability of the available component stack, a multi stage environment has been setup. There exist 3 stages currently:

- The developers' area: here the developers put their packages.
- The testing stage: this is a collection of LOD2 packages that are subject to integration tests. The goal is to detect with automatic testing problems in the installation of the packages.
- The stable stage: this is a collection of LOD2 packages that pass the tests.

The LOD2 stack managers are responsible for moving packages from the developers' area into the testing stage and then to the stable stage.

Orthogonally we have created 2 repositories that are OS-release dependent. These contain components that are dependent on the OS. The typical example is Virtuoso for which Ubuntu 12.04 64 bit builds are provided. Build and installation instructions are present to support more recent and other Linux distributions.

Developers have to contribute a Debian package with the correct LOD2 stack configuration. The rationale behind this choice for this approach is to distribute the knowledge on building packages to all partners, but more important to create awareness for software deployment. When the component owners are responsible for building the Debian packages they face the imperfections that make their software hard to deploy. That experience has as positive effect that deployment issues are tackled early on. All necessary information for the developers is collected in the contribution documentation[2].

The LOD2 stack repository distributes software components. However in addition to these components, there are more components or information sources valuable to the LOD2 stack available online. These come in two categories:

- software components which are only accessible online due to various reasons: special setup, license rules, etc. Examples are the Sindice search engine and PoolParty.
- information sources: for example dbpedia.org and vocabulary.wolterskluwer.de.

2.4 Installing the LOD2 Linked Data Stack

The installation of a local version of the LOD2 stack[3] is done in a few steps and are available online[4]. In short, the next steps must be executed:

1. Setup a Ubuntu 12.04 64 bit LTS system.
2. Download the repository package (of the stage of interest) and install it.
3. Update the local package cache.
4. Install the whole LOD2 stack or selected components.

If during the installation issues occur, support-stack@lod2.eu can be contacted for assistance. Background information and frequently occurring situations are documented at documentation wiki[5].

2.5 The LOD2 Linked Data Stack Release

In the following paragraphs all LOD2 stack components are summarized. First we list those that are available as Debian package, followed by those that are online available and finally we provide a table of online data sources that are of interest and have been supported by the LOD2 project.

2.5.1 Available as Debian Packages

Components *colanut, limes* (ULEI). LIMES is a link discovery framework for the Web of Data. It implements time-efficient approaches for large-scale link discovery based on the characteristics of metric spaces. The COLANUT (COmplex

[2] http://stack.linkeddata.org/how-to-contibute/
[3] http://stack.linkeddata.org
[4] http://wiki.lod2.eu/display/LOD2DOC/How+To+Start
[5] http://wiki.lod2.eu/display/LOD2DOC/Known+issues

Linking in A NUTshell) interface resides on top of LIMES and supports the user during the link specification process.
http://aksw.org/Projects/LIMES

Components *d2r, d2r-cordis* (UMA). D2R Server is a tool for publishing relational databases on the Semantic Web. It enables RDF and HTML browsers to navigate the content of the database, and allows applications to query the database using the SPARQL query language. The d2r-cordis package contains an example setup.
http://d2rq.org/d2r-server

Components *dbpedia-spotlight-gui, dbpedia-spotlight* (UMA/ULEI). DBpedia Spotlight aims to shed light on the Web of Documents. It recognizes and disambiguates DBpedia concepts/entities mentioned in natural language texts. There is an online instance that can be used for experimenting.
https://github.com/dbpedia-spotlight/dbpedia-spotlight/

Components *dl-learner-core, dl-learner-interfaces* (ULEI). The DL-Learner software learns concepts in Description Logics (DLs) from examples. Equivalently, it can be used to learn classes in OWL ontologies from selected objects. It extends Inductive Logic Programming to Descriptions Logics and the Semantic Web. The goal of DL-Learner is to provide a DL/OWL based machine learning tool to solve supervised learning tasks and support knowledge engineers in constructing knowledge and learning about the data they created.
http://dl-learner.org/

Component *libjs-rdfauthor* (ULEI). RDFauthor is an editing solution for distributed and syndicated structured content on the World Wide Web. The system is able to extract structured information from RDFa-enhanced websites and to create an edit form based on this data.
http://aksw.org/Projects/RDFauthor

Component *LODRefine* (Zemanta). Open Refine (http://openrefine.org/) extended with RDF (http://refine.deri.ie/) and Zemanta API (http://developer.zemanta.com) extensions.
http://code.zemanta.com/sparkica/

Component *lod2demo* (TenForce). The LOD2 demonstrator is a web application which brings together all LOD2 stack components in one interface. Components are loosely coupled through the Virtuoso store via which all information is exchanged. It also serves as the top level meta package in order to install the whole stack content on a machine. This is the top level meta package which installs the whole stack on a machine.
https://lod2-stack.googlecode.com/svn/trunk/lod2demo

Component *lod2-statistical-workbench* (TenForce/IMP). A web interface that aggregates several components of the stack organized in an intuitive way to support the specific business context of the statistical office. The workbench contains several dedicated extension for the manipulation of RDF data according to the

Data Cube vocabulary: validation, merging and slicing of cubes are supported. Also the workbench has been used to explore authentication via WebID and keeping track of the data manipulations via a provenance trail.
https://lod2-stack.googlecode.com/svn/trunk/lod2statworkbench

Component *lod2webapi* (TenForce). An REST API allowing efficient graph creation and deletion as well as regex based querying. It also support a central prefix management.
https://lod2-stack.googlecode.com/svn/trunk/lod2-webapi

Component *unifiedviews* (SWCG). UnifiedViews is Linked (Open) Data Management Suite to schedule and monitor required tasks (e.g. preform reoccurring extraction, transformation and load processes) for smooth and efficient Linked (Open) Data Management to support web-based Linked Open Data portals (LOD platforms) as well as sustainable Enterprise Linked Data integrations inside of organisations.
https://grips.semantic-web.at/display/UDDOC/Introduction

Components *ontowiki, ontowiki-common, ontowiki-mysql, ontowiki-virtuoso, owcli, liberfurt-php* (ULEI). OntoWiki is a tool providing support for agile, distributed knowledge engineering scenarios. OntoWiki facilitates the visual presentation of a knowledge base as an information map, with different views on instance data. It enables intuitive authoring of semantic content. It fosters social collaboration aspects by keeping track of changes, allowing to comment and discuss every single part of a knowledge base.
http://ontowiki.net

Component *ontowiki-cubeviz* (ULEI). CubeViz is a facetted browser for statistical data utilizing the RDF Data Cube vocabulary which is the state-of-the-art in representing statistical data in RDF. Based on the vocabulary and the encoded Data Cube, CubeViz is generating a facetted browsing widget that can be used to filter interactively observations to be visualized in charts. On the basis of the selected structure, CubeViz offer beneficiary chart types and options which can be selected by users.
http://aksw.org/Projects/CubeViz

Component *ontowiki-csv-import* (ULEI). Statistical data on the web is often published as Excel sheets. Although they have the advantage of being easily readable by humans, they cannot be queried efficiently. Also it is difficult to integrate with other datasets, which may be in different formats. To address those issues this component was developed, which focusses on conversion of multi-dimensional statistical data into RDF using the RDF Data Cube vocabulary.
https://github.com/AKSW/csvimport.ontowiki

Component *ore-ui* (ULEI). The ORE (Ontology Repair and Enrichment) tool allows for knowledge engineers to improve an OWL ontology by fixing inconsistencies and making suggestions for adding further axioms to it.
http://ore-tool.net/

Component *r2r* (UMA). R2R is a transformation framework. The R2R mapping API is now included directly into the LOD2 demonstrator application, allowing users to experience the full effect of the R2R semantic mapping language through a graphical user interface.

Component *rdf-dataset-integration* (ULEI). This tool allows the creation of debian packages for RDF datasets. Installing a created package will autoload the RDF dataset in the Virtuoso on the system.

Component *sieve* (UMA). Sieve is a Linked Data Quality Assessment and Fusion tool. It performs quality assessment and resolves conflicts in a task-specific way according to user configuration.
http://sieve.wbsg.de

Component *sigmaee* (DERI). Sigma EE is an entity search engine and browser for the Web of Data. Sig.ma EE is a standalone, deployable, customisable version of Sig.ma. Sig.ma EE is deployed as a web application and will perform on the fly data integration from both local data source and remote services (including Sindice.com).
http://sig.ma

Component *silk* (UMA). The Silk Linking Framework supports data publishers in setting explicit RDF links between data items within different data sources. 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 fulfil 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.
http://wifo5-03.informatik.uni-mannheim.de/bizer/silk/

Component *silk-latc* (DERI). An improved version of SILK that has been used in the LATC project.

Component *siren* (DERI). SIREn is a Lucene/Solr extension for efficient schemaless semi-structured full-text search. SIREn is not a complete application by itself, but rather a code library and API that can easily be used to create a full-featured semi-structured search engine.
http://rdelbru.github.io/SIREn/

Component *sparqled* (DERI). SparQLed is an interactive SPARQL editor that provides context-aware recommendations, helping users in formulating complex SPARQL queries across multiple heterogeneous data sources.
http://sindicetech.com/sindice-suite/sparqled/

Component *sparqlify* (ULEI). Sparqlify is a SPARQL-SQL rewriter that enables one to define RDF views on relational databases and query them with SPARQL.
https://github.com/AKSW/Sparqlify

Component *sparqlproxy-php* (ULEI). A PHP forward proxy for remote access to SPARQL endpoints; forwards request/response headers and filters out non-SPARQL URL arguments.
https://github.com/AKSW/SparqlProxyPHP

Component *spatial-semantic-browser* (). The spatial semantic browser (recently labeled as Facete) project is comprised of a JavaScript library for faceted browsing of RDF data and an application for browsing geo-related RDF data. The application thereby offers filtering by facets, pivoting, and display of the data on a map.

Component *stanbol* (External - ULEI). Apache Stanbol's intended use is to extend traditional content management systems with semantic services.
https://stanbol.apache.org/

Component *valiant* (TenForce). Valiant is an command line tool that automates the extraction of RDF data from XML documents. Intended for bulk application on a large amount of XML documents.
https://github.com/tenforce/valiant

Component *virtuoso-opensource* (OGL). Virtuoso is a knowledge store and virtualization platform that transparently integrates Data, Services, and Business Processes across the enterprise. Its product architecture enables it to deliver traditionally distinct server functionality within a single system offering along the following lines: Data Management & Integration (SQL, XML and EII), Application Integration (Web Services & SOA), Process Management & Integration (BPEL), Distributed Collaborative Applications.
http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/

Components *virtuoso-vad-bpel, virtuoso-vad-conductor, virtuoso-vad-demo, virtuoso-vad-doc, virtuoso-vad-isparql, virtuoso-vad-ods, virtuoso-vad-rdfmappers, virtuoso-vad-sparqldemo, virtuoso-vad-syncml, virtuoso-vad-tutorial* (OGL). Virtuoso Application Distributions as debian packages, These VAD packages extend the functionality of Virtuoso, i.e. there is the web system admin interface (the conductor), the interactive SPARQL interface and many more.
http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/

Component *EDCAT* (TenForce). EDCAT is a service API for a DCAT based Linked Data catalogue. It provides a json compatible view with the DCAT W3C standard. The API is extendible via a plugin architecture. Its main purpose is to serve as an integration layer for to establish dataset catalogues inside organizations.
http://edcat.tenforce.com

Component *CKAN* (OKF). CKAN is a system for storing, cataloguing and visualising data or other "knowledge" resources. CKAN aims to make it easy to find, share and re-use open content and data, especially in ways that are machine automatable.
http://ckan.org

Component *lod2stable-repository, lod2testing-repository.* These packages activate the repositories in order to get a coherent stack installation.
https://lod2-stack.googlecode.com/svn/trunk/lod2repository

2.5.2 Available as Online Component

Component *Payola* (UEP). Payola is a web application which lets you work with graph data in a new way. You can visualize Linked Data using several preinstalled plugins as graphs, tables, etc. Moreover, you can create an analysis and run it against a set of SPARQL endpoints. Analysis results are processed and visualized using the embedded visualization plugins.
http://www.payola.cz

Component *PoolParty Thesaurus Manager* (SWC). PoolParty is a thesaurus management system and a SKOS editor for the Semantic Web including text mining and linked data capabilities. The system helps to build and maintain multilingual thesauri providing an easy-to-use interface. PoolParty server provides semantic services to integrate semantic search or recommender systems into systems like CMS, DMS, CRM or Wikis.
http://poolparty.biz

Component *PoolParty Extractor* (SWC). The PoolParty Extractor (PPX) offers an API providing text mining algorithms based on semantic knowledge models. With the PoolParty Extractor you can analyse documents in an automated fashion, extracting meaningful phrases, named entities, categories or other metadata. Different data or metadata schemas can be mapped to a SKOS thesaurus that is used as a unified semantic knowledge model.
http://lod2.poolparty.biz/extractor/testextractor

Component *Sindice* (DERI, OGL). Sindice is a state of the art infrastructure to process, consolidate and query the Web of Data. Sindice collates these billions of pieces of metadata into an coherent umbrella of functionalities and services.
http://sindice.com

2.5.3 Available Online Data Sources

Table 1 provides an overview of the main data sources to which LOD2 has contributed.

2.5.4 The LOD2 Stack Components Functional Areas Coverage

When distributing the components over the Linked Data Publishing cycle functionalities the following Fig. 2 is obtained. In this figure the component is associated with its main role in the data publishing cycle. In the middle are placed the applications that are not dedicated to one area, such as the automatization platform UnifiedViews, and the applications that exploit the LOD2 stack, such as the lod2demo and LOD2 Statistical Workbench. Online components are

Table 1. Online Data Sources supported by LOD2

URL	SparqlEndpoint
Sindice (DERI, OGL)	
http://sindice.com	
CKAN Repositories (OKFN)	
http://publicdata.eu	http://publicdata.eu/sparql
Dbpedia (ULEI, SWC, OGL)	
http://dbpedia.org, http://live.dbpedia.org, http://de.dbpedia.org	http://dbpedia.org/sparql
LODcloud (OGL)	
http://lod.openlink.com	http://lod.openlink.com/sparql
WebDataCommons RDFa, Microdata and Microformat data sets (UMA)	
http://webdatacommons.org/structureddata/	http://webdatacommons.structureddata/
German Courts and Labor Law taxonomies (WKD, SWC)	
http://vocabulary.wolterskluwer.de/arbeitsrecht.html	http://vocabulary.wolterskluwer.de/PoolParty/sparql/arbeitsrecht
http://vocabulary.wolterskluwer.de/courts.html	http://vocabulary.wolterskluwer.de/PoolParty/sparql/court

marked with a globe icon. The cubed formatted components are dedicated components for the statistical domain. They are embedded in the LOD2 Statistical Workbench.

In this figure, we see that the current version of the stack has a high number of components geared towards the extraction, storage and querying parts. This can be explained by the large number of data formats that need to be transformed to RDF. Every component tackles a specific subset of these formats. Most of the other components contain a small selection of specialized tools for a specific task. In the search/browsing/exploration stage, every component has its own way of visualizing the data.

3 A Customized Linked Data Stack for Statistics

The work on the LOD2 Statistical Workbench was motivated by the need to support the process of publishing statistical data in the RDF format using

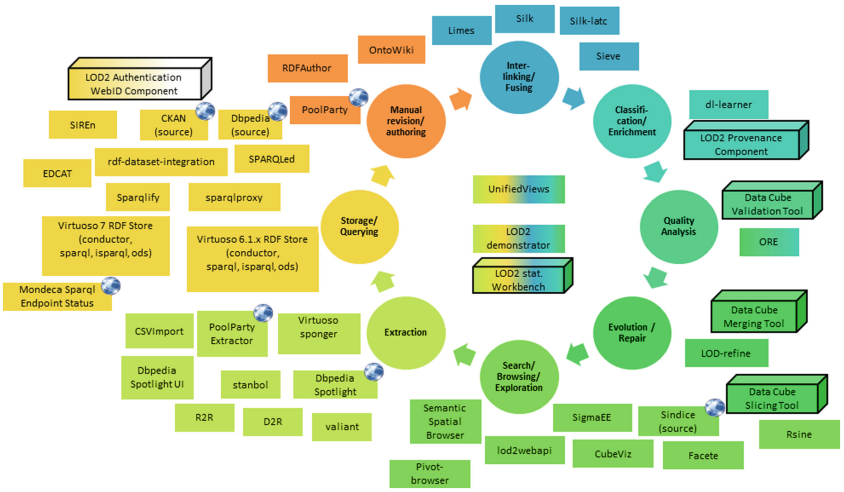

Fig. 2. Distribution of the LOD2 stack components w.r.t. Linked Data Publishing cycle

common vocabularies such as the RDF Data Cube[6]. The aim here was to provide support for performing different operations such as

- efficient transformation/conversion of traditional data stores (e.g. CSV, XML, relational databases) into linked, machine readable formats;
- building and querying triple stores containing RDF Data Cubes;
- validating RDF Data Cubes;
- interlinking and adding meaning to data;
- visualization and exploration of multi-dimensional RDF Data Cubes;
- publishing statistical data using a LOD publication strategy and respective metadata about the RDF data cube within a selected portal (i.e. a CKAN instance).

The potential benefits of converting statistical data into Linked Data format were studied through several scenarios for the National Statistical Office use case (cf. Table 2) [1].

3.1 Application Architecture and Scenarios

The LOD2 Statistical Workbench[7] implements the Linked Data application architecture sketched in Sect. 2. The workbench introduces a number of new components such as the *RDF Data Cube Validation tool*, the *RDF Data Cube Slicing tool* and the *RDF Data Cube Merging tool* dedicated for the statistical

[6] http://www.w3.org/TR/vocab-data-cube
[7] http://demo.lod2.eu/lod2statworkbench

Table 2. Potential goals and benefits of converting statistical data into Linked Data.

Scenario	Benefits/expected added value
Goal: *Metadata management*	
Code lists - creating and maintaining	Standardization on the metadata level: (a) will allow harmonization of specific concepts and terminology, (b) will improve interoperability and (c) will support multilinguality in statistical information systems across Europe
Goal: *Export*	
Export to different formats	Data exchange with other semantic tools, as well as other commonly used spreadsheet tool e.g. Microsoft Excel
Goal: *RDF Data Cube - Extraction, Validation and Initial Exploration*	
	Standardization of the extraction process
(a) CSV Data Extraction	CSV2DataCube
(b) XML Data Extraction	XML2DataCube
(c) SDMX-ML 2 RDF/XML	SDMX2RDFDataCube
Goal: *RDF Data Cube Quality Assessment (validation and analysis of integrity constraints)*	
Building well-formed RDF Data Cubes, where statistical data has been assigned an unique URI, meaning and links to similar data. This approach facilitates search and enables re-use of public statistical data	The well-formed RDF Data Cubes satisfy a number of integrity constraints and contain metadata thus enabling automation of different operations (exchange, linking, exploration)
Goal: *RDF Data Cube - Transformation, Exploratory Analysis and Visualization*	
(a) Merging RDF Data Cubes	Data fusion i.e. creation of a single dataset and different graphical charts that supports the exploratory analysis (e.g. indicator comparison)
(b) Slicing RDF Data Cubes	Facilitate creation of intersections in multidimensional data
(c) Visualization of RDF Data Cubes	Efficient analysis and search for trends in statistical data
Goal: *Interlinking*	
(a) Code lists - Interlinking	Assigning meaning, improved interoperability of data with similar governmental agencies
(b) CSV Data Extraction and Reconciliation with DBpedia	Assigning meaning
Goal: *Publishing*	
Publishing to CKAN	Increased transparency, improved accessibility of statistical data

domain. The workbench has also been augmented with extensions to explore other aspects: the *LOD2 authentication component*, the *LOD2 provenance component* and the *CKAN Publisher*.

In order to support the end-user, a new graphical user interface as been created wherein the LOD2 components are more intuitively organized for the statistical domain. There are grouped in the five topics: Manage Graph, Find more Data Online, Edit & Transform, Enrich Datacube, and Present & Publish.

Import features. The LOD2 Statistical Workbench is a framework for managing Linked Data stored in the RDF Data Cube format. Because statistical data is often provided in tabular format, it supports importing data from CSV. The CSV2RDF component allows the end users to transform tabular data from a CSV file into a multidimensional RDF Data Cube. Alternatively, LODRefine can be used. LODRefine is capable to import all kinds of structured formats including CSV, ODS and XSL(X) and transform them to RDF graphs based on arbitrary vocabularies.

Also the import from XML files is supported. The main international standard for exchanging statistical data is SDMX[8]. The users have the possibility to pass XML data as input to the XSLT processor and transform into RDF. The workbench provides ready to use XSLT scripts to deal with SDMX formatted data.

Additionally, using the *Find more Data Online* submenu, the user is able to find and import more data into the local RDF store using the respective tool of Statistical Workbench.

Semantic integration and storage. Linked Data applications are based on server platforms that enable RDF triple storage, semantic data integration and management, semantic interoperability based on W3C standards (XML, RDF, OWL, SOA, WSDL, etc). The Virtuoso Universal Server is used for this purpose in the LOD2 Statistical Workbench.

RDF Data Cube transformation features. Specialized components have been developed to support the most common operations for manipulating statistical data such as merging datasets, creating slices and data subsetting (Edit & Transform submenu). As each dataset defines components (e.g. dimensions used to describe the observations), the merging algorithm checks the adequacy of the input datasets for merging and compiles a new RDF Data Cube to be used for further exploration and analysis. Additionally, the slicing component can be used to group subsets of observations where one or more dimensions are fixed. This way, slices are given an identity (URI) so that they can be annotated or externally referenced, verbosity of the data set can be reduced because fixed dimensions need only be stated once, and consuming applications can be guided in how to present the data.

[8] http://sdmx.org

RDF Data Cube validation. The RDF Data Cube Validation tool [8] supports the identification of possibly not well-formed parts of an RDF Data Cube. The therein integrated analysis process consists mostly of integrity constraints rules represented as SPARQL queries as are defined in RDF Data Cube standard. The validation operation is applicable at several steps in the Linked Data publishing process e.g. on import/extraction/transformation from different sources or after fusion and creation of new RDF Data Cubes.

Authoring, querying and visualization. The OntoWiki authoring tool facilitates the authoring of rich semantic knowledge bases, by leveraging Semantic Wiki technology, the WYSIWYM paradigm (What You See Is What You Mean [3]) and distributed social, semantic collaboration and networking techniques. CubeViz, an extension of OntoWiki, is a facetted browser and visualization tool for statistical RDF data. It facilitates the discovery and exploration of RDF Data Cubes while hiding its complexity from users. In addition to using the browsing and authoring functionality of OntoWiki, advanced users are able to query the data directly (SPARQL) using one of the following offered SPARQL editors: OntoWiki query editor, Sindices SparQLed component and the OpenLink Virtuoso SPARQL editor.

Enrichment and interlinking. Linked Data publishing isn't just about putting data on the web, but also about creating links, so that a person or machine can explore the web of data. Therefore, the enrichment and interlinking features are very important as a pre-processing step in the integration and analysis of statistical data from multiple sources. LOD2 tools such as SILK and Limes facilitate mapping between knowledge bases, while LOD Open Refine can be used to enrich the data with descriptions from DBpedia or to reconcile with other information in the LOD cloud. PoolParty allows users to create their own high quality code lists and link the concepts therein to external sources as well. Once the code lists have been established, they can be reused as dimension values in Data Cubes or linked to Cubes that have been created separately.

Export and Linked Data publishing. The LOD2 Statistical Workbench export features are reachable via the Manage Graph and Present & Publish submenus. The Manage Graph option allows exporting of a graph with all its content in RDF/XML, RDF/JSON, Turtle, Notation 3. CubeViz supports subsetting of the data and extraction of a portion that is interesting for further analysis in CSV and RDF/XML format. The *CKAN Publisher* component aims at automating the upload and registration of new data with existing CKAN instances.

The use of the LOD2 Statistical Workbench for different data management operations is illustrated with online tutorials[9] for the scenarios summarized in Table 2.

[9] http://wiki.lod2.eu/display/LOD2DOC/LOD2+Statistical+Workbench

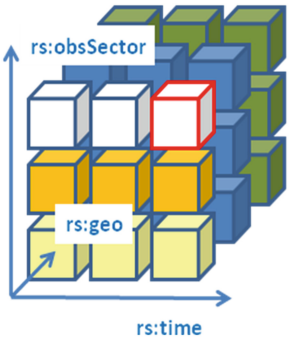

Fig. 3. RDF Data Cube - graphical representation

3.2 LOD2 Statistical Workbench in Use

This section provides some basic concepts of the Data Cube Vocabulary and how these were adapted in the Statistical Workbench, followed by some examples of using the workbench.

3.2.1 The RDF Data Cube Vocabulary

A statistical data set comprises a collection of observations (see Fig. 3) made at some points across some logical space. Using the RDF Data Cube vocabulary, a resource representing the entire data set is created and typed as qb:DataSet[10] and linked to the corresponding data structure definition via the qb:structure property.

The collection must be characterized by a set of dimensions (qb: DimensionProperty) that define what the observation applies to (e.g. time rs: time, observed sector rs:obsSector, country rs:geo)[11] along with metadata describing what has been measured (e.g. economic activity, prices) through measurements. Optionally, additional information can be provided on how the observation or cube was measured and how the observations are expressed through the use of attribute (qb:AttributeProperty) elements (e.g. units, multipliers, status).

The qb:dataSet property (see excerpt below) indicates that a specific qb:Observation instance is a part of a dataset. In this example, the primary measure, i.e. observation value (represented here via sdmx-measure:obsValue), is a plain decimal value. To define the units the observation in question is measured in, the sdmx-attribute:unitMeasure property which corresponds to the SDMX-COG concept of UNIT_MEASURE was used. In the example, the code MIO_NAT_RSD corresponds to millions of national currency (Serbian dinars). The values in the time and location dimensions (rs:geo and rs:time), indicate that the observation took place in the Republic of Serbia (geographic region code RS), and in 2003 (time code Y2003), respectively.

[10] qb is the prefix http://purl.org/linked-data/cube#.
[11] rs is the prefix http://elpo.stat.gov.rs/lod2/RS-DIC/rs/.

```
@prefix qb: <http://purl.org/linked-data/cube#> .
@prefix rs: <http://elpo.stat.gov.rs/lod2/RS-DIC/rs/> .
@prefix accounts: <http://elpo.stat.gov.rs/lod2/RS-DATA/NA/dsd/> .
@prefix time: <http://elpo.stat.gov.rs/lod2/RS-DIC/time/> .
@prefix geo: <http://elpo.stat.gov.rs/lod2/RS-DIC/geo/> .
@prefix measure: <http://elpo.stat.gov.rs/lod2/RS-DIC/esa95/> .

<http://elpo.stat.gov.rs/lod2/RS-DATA/NA/GDP_usage_Exports/data> a qb:DataSet ;
 rdfs:label "GDP␣usage␣-␣Exports"^^xsd:string ;
 rdfs:comment "Source:␣RZS␣(http://www.stat.gov.rs/)" ;
 qb:structure accounts:GDP_usage_Exports ;
 dcterms:subject <http://purl.org/linked-data/sdmx/2009/subject/2.2>;
 dc:publisher "Stat.␣Office␣of␣the␣Republic␣of␣Serbia"^^xsd:string .

<http://elpo.stat.gov.rs/lod2/RS-DATA/NA/GDP_usage/data/obs46> a qb:Observation;
 qb:dataSet <http://elpo.stat.gov.rs/lod2/RS-DATA/NA/GDP_usage/data> ;
 sdmx-attribute:unitMeasure measure:MIO_NAT_RSD ;
 sdmx-measure:obsValue "124309.7" ;
 rs:obsSector <http://elpo.stat.gov.rs/lod2/RS-DIC/esa95/P31_S13>;
 rs:geo geo:RS ;
 rs:time time:Y2003.
```

Listing 1.1: RDF representation of an observation

Each data set has a set of structural metadata (see Table 3). These descriptions are referred to in SDMX and the RDF Data Cube Vocabulary as Data Structure Definitions (DSD). Such DSDs include information about how concepts are associated with the measures, dimensions, and attributes of a data cube along with information about the representation of data and related metadata, both identifying and descriptive (structural) in nature. DSDs also specify which code lists provide possible values for the dimensions, as well as the possible values for the attributes, either as code lists or as free text fields. A DSD can be used to describe time series data, cross-sectional and multidimensional table data. Because the specification of a DSD is independent of the actual data that the data cube is about, it is often possible to reuse a DSD over multiple data cubes.

3.2.2 Example 1: Quality Assessment of RDF Data Cubes

Prior to publishing the resulting RDF data on an existing data portal and thus enabling other users to download and exploit the data for various purposes, every dataset should be validated to ensure it conforms to the RDF Data Cube model.

Table 3. Example values for a Data Cube structure representing the Serbian economic statistics

Component property	Concept description	Identifier	Code list
Dimension	Geographical region	rs:geo	cl:geo
Dimension	Time	rs:time	cl:time
Dimension	Economic activity	rs:activityNACEr2	cl:nace_rev2
Attribute	Unit of measurement	sdmx-attribute:unitMeasure	cl:esa95-unit
Measure	Observed value	sdmx-measure:obsValue	

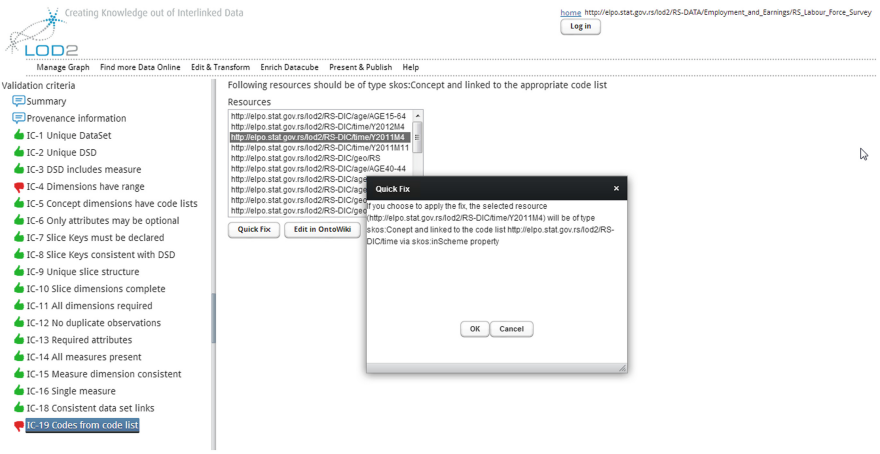

Fig. 4. RDF Data Cube - quality assessment

The data validation step is covered by the LOD2 stack, i.e. through the following software tools:

- The RDF Data Cube Validation Tool[12];
- CubeViz, a tool for visualization of RDF Data Cubes[13].

The RDF Data Cube Validation Tool aims at speeding-up the processing and publishing of Linked Data in RDF Data Cube format. Its main use is validating the integrity constraints defined in the RDF Data Cube specification. It works with the Virtuoso Universal Server as a backend and can be run from the LOD2 Statistical Workbench environment.

The main benefits of using this component are improved understanding of the RDF Data Cube vocabulary and automatic repair of identified errors. Figure 4 shows the component in action: the integrity constraints and their status are shown on the left side, while the results of analysis are shown on the right. A list of resources that violate the constraint, an explanation about the problem, and if possible, a quick solution to the problem is offered to the user. Once an RDF Data Cube satisfies the Data Cube integrity constraints, it can be visualized with CubeViz. More details can be found in the LOD2 Stack Documentation[14].

3.2.3 Example 2: Filtering, Visualization and Export of RDF Data Cubes

The facetted browser and visualization tool CubeViz can be used to filter observations to be visualized in charts interactively. Figure 5 shows an exploration session that comprises of the following steps:

[12] http://wiki.lod2.eu/display/LOD2DOC/RDF+Data+Cube+Quality+Assessment
[13] http://wiki.lod2.eu/display/LOD2DOC/Data+Subsetting+and+Export+Scenario
[14] http://wiki.lod2.eu/display/LOD2DOC/LOD2+Statistical+Workbench

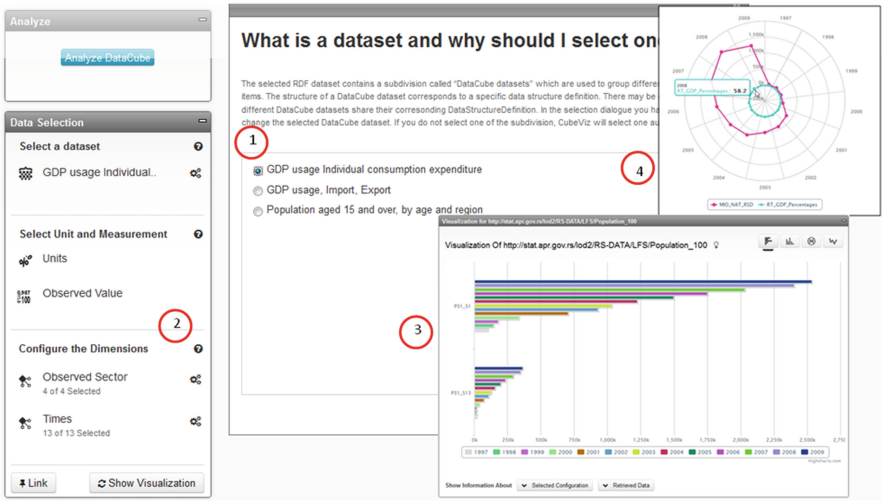

Fig. 5. RDF Data Cube - exploration and analysis

1. Select one out of the available datasets in the RDF graph;
2. Choose observations of interest by using a subset of the available dimensions;
3. Visualize the statistics by using slices, or
4. Visualize the statistics in two different measure values (millions of national currency and percentages).

3.2.4 Example 3: Merging RDF Data Cubes

Merging[15] is an operation of creating a new dataset (RDF Data Cube) that compiles observations from the original datasets (two or more), and additional resources (e.g. data structure definition, component specifications). In order to obtain meaningful charts the observed phenomena (i.e. serial data) have to be described on the same granularity level (e.g. year, country) and expressed in same units of measurement (e.g. euro, %). Therefore alignment of the code lists used in the input data is necessary before the merging operation is performed.

3.3 Towards a Broader Adoption

Linked Data principles have been introduced into a wide variety of application domains, e.g. publishing statistical data and interpretation of statistics [5], improving tourism experience [6], pharmaceutical R&D data sharing [7], crowdsourcing in emergency management [4], etc. A few years ago, our analysis of the adoption of Semantic Web technologies by enterprises [2] has shown that companies benefit from features that improve data sharing and re-use (57 %), improve searching (57 %), allow incremental modelling (26 %), explicit content

[15] http://wiki.lod2.eu/display/LOD2DOC/Eurostat+Merge+and+Enhance+Scenario

relation (24 %), identifying new relationships (17 %), dynamic content generation (14 %), personalization (10 %), open modeling (12 %), rapid response to change (10 %), reducing time to market (5 %), and automation (5 %). Of the features the LOD2 Statistical Workbench provides functionality improving the following areas: data share and re-use, improved search, explicit content relation, identifying new relationships, open model and automation. The LOD2 Statistical Workbench supports both publishers and consumers of Linked Data such as national statistical offices (institutes), national banks, publication offices, etc. Some of those collaborated with us as an early adopter of the approach.

3.3.1 Use Case 1: Digital Agenda Scoreboard

In the course of the LOD2 PUBLINK 2010 activities, the digital agenda scoreboard[16] has been created as the first web portal exploiting the RDF Data Cube. The Digital Agenda Scoreboard provides insight on how 'digital' Europa is. By using an early version of CubeViz the trends are visualized embedded in human readable scenario. Behind the scenes the data is provided and aggregated in a Virtuoso store according to the Data Cube vocabulary. This data is made available to the public in different formats including the RDF representation.

3.3.2 Use Case 2: Statistical Office of the Republic of Serbia (SORS)

In the course of the LOD2 PUBLINK 2011 activities, the SORS public data was integrated into the LOD cloud via the Serbian CKAN [1]. The Serbian CKAN is a metadata repository to be used for dissemination purposes by Serbian national institutions. Maintenance activities include identifying changes in the dissemination data (new public data, changes on metadata level) and fixing the mapping process (from XML to RDF) accordingly. The SORS is in the process of adopting the LOD2 Statistical Workbench[17] that will allow the users to automatically publish data (in the existing and new packages) to the Serbian CKAN.

3.3.3 Use Case 3: Business Registers Agency

In the course of the LOD2 PUBLINK 2012 activities, example data from the Regional Development Measures and Incentives Register was triplified using the LOD2 Statistical Workbench and registered with the Serbian CKAN. The data is reachable via the Serbian CKAN[18] and can be explored through a prototype application[19].

[16] http://digital-agenda-data.eu/
[17] http://lod2.stat.gov.rs/lod2statworkbench
[18] http://rs.ckan.net/dataset/apr-register-of-regional-development-measures-and-incentives
[19] http://rs.ckan.net/esta-ld/

3.3.4 Challenges Faced by Early Adopters

For each early adopter the publishing of the statistical data as Linked Data has influenced their data publishing process. The Linked Data vision impacts the publication process typically more deeply as it sees data from a more universal perspective and not as an isolated piece of information. When the statistical observations becomes real Linked Data, it means that also the dimensions have to become Linked Data, and this typically means that other organizations that maintain the dimensions have to be consulted.

Therefore in addition to our technological support in the identifying the set of applicable vocabularies and specifying the transformation flow to be setup, there has been an important activity in supporting the early adopters with their relationship with their data suppliers.

Over time the technological support has been improved. Whereas for the first case, the Digital Agenda Scoreboard, many of the transformation steps and data cleaning steps had to be done manually, they are for the more recent applications semi-automated.

Our approach to customize the LOD2 stack not only holds for the statistical domain, but can be applied other domains as well. For instance in the GeoKnow[20] project the GeoKnow Generator is being created for the support of geo-spatial Linked Data.

4 Conclusion

The LOD2 stack has successfully established a dissemination platform for Linked Data software. After the incubation period inside the LOD2 project the Linked Data community continues to be supported via http://stack.linkeddata.org. The success of the Linked Data stack is in the first place due to the quality and the progress of the software components it distributes. The inclusion of new and updated software is the oxygen that keeps the Linked Data stack alive. This oxygen will be provided by the core contributors as they keep on improving their components and are devoted to provide regular improvement releases to the stack.

The Statistical Workbench shows that starting from the LOD2 stack, the foundations are present to create applications tuned for a particular information domain. With the current state of the LOD2 stack, data managers can prototype the required data information streams. Although there is no uniform homogenous end-user interface, exactly this prototyping ability is crucial in bootstrapping the design of the desired end-user interfaces.

[20] http://geoknow.eu/

References

1. Janev, V., Milosevic, U., Spasic, M., Vranes, S., Milojkovic, J., Jirecek, B.: Integrating serbian public data into the LOD cloud. In: Ivanovic, M., Budimac, Z., Radovanovic, M. (eds.) BCI, pp. 94–99. ACM (2012)
2. Janev, V., Vranes, S.: Applicability assessment of semantic web technologies. Inf. Process. Manage. **47**(4), 507–517 (2011)
3. Khalili, A., Auer, S.: WYSIWYM authoring of structured content based on Schema.org. In: Lin, X., Manolopoulos, Y., Srivastava, D., Huang, G. (eds.) WISE 2013, Part II. LNCS, vol. 8181, pp. 425–438. Springer, Heidelberg (2013)
4. Ortmann, J., Limbu, M., Wang, D., Kauppinen, T.: Crowdsourcing linked open data for disaster management. In: Terra Cognita Workshop 2011 at ISWC2011. CEUR WS Proceedings (2011)
5. Paulheim, H.: Generating possible interpretations for statistics from linked open data. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 560–574. Springer, Heidelberg (2012)
6. Sabou, M., Brasoveanu, A.M.P., Arsal, I.: Supporting tourism decision making with linked data. In: Presutti, V., Pinto, H.S. (eds.) I-SEMANTICS, pp. 201–204. ACM (2012)
7. Samwald, M., Jentzsch, A., Bouton, Ch., Kallesoe, C., Willighagen, E.L., Hajagos, J., Scott Marshall, M., Prud'hommeaux, E., Hassanzadeh, O., Pichler, E., Stephens, S.: Linked open drug data for pharmaceutical research and development. J. Cheminform. **3**, 19 (2011)
8. Mijovic, V., Janev, V., Vrane, S.: LOD2 tool for validating RDF data cube models. In: Conference Web Proceedings of the ICT Innovation Conference 2013 (2013)