Inf Syst Front (2017) 19:321–336
DOI 10.1007/s10796-016-9687-1

Challenges and opportunities in renovating public sector information by enabling linked data and analytics

Spiros Mouzakitis[1] · **Dimitris Papaspyros**[1] · **Michael Petychakis**[1] · **Sotiris Koussouris**[1] · **Anastasios Zafeiropoulos**[2] · **Eleni Fotopoulou**[2] · **Lena Farid**[3] · **Fabrizio Orlandi**[4] · **Judie Attard**[4] · **John Psarras**[1]

Published online: 6 August 2016

Abstract Linked Data has become the current W3C recommended approach for publishing data on the World Wide Web as it is sharable, extensible, and easily re-usable. An ecosystem of linked data hubs in the Public Sector has the potential to offer significant benefits to its consumers (other public offices and ministries, as well as researchers, citizens and SMEs), such as increased accessibility and re-use value of their data through the use of web-scale identifiers and easy interlinking with datasets of other public data providers. The power and flexibility of the schema-defying Linked Data, however, is counterbalanced by inborn factors that diminish the potential for cost-effective and efficient adoption by the Public Sector. The paper analyzes these challenges in view of the current state-of-the-art in linked data technologies and proposes a technical framework that aims to hide the underlying complexity of linked data while maintaining and promoting the interlinking capabilities enabled by the Linked Data Paradigm. The paper presents the innovations behind our proposed solutions as well as their advantages, especially for the non-expert users.

Keywords Linked data · Public sector information · Open data · RDF · Analytics

1 Introduction

As public sector information data sources and initiatives have proliferated over the last years, linking and combining datasets has become one of the major topics for the data consumers (researchers, citizens, SMEs). Due to the massive growth of available data, conventional methods of data integration are bound to fail while the complexity of processes within organizations ask for more agile options to link and mash-up data

✉ Spiros Mouzakitis
smouzakitis@epu.ntua.gr

Dimitris Papaspyros
dpap@epu.ntua.gr

Michael Petychakis
mpetyx@epu.ntua.gr

Sotiris Koussouris
skous@epu.ntua.gr

Anastasios Zafeiropoulos
efotopoulou@ubitech.eu

Eleni Fotopoulou
azafeiropoulos@ubitech.eu

Lena Farid
lena.farid@fokus.fraunhofer.de

Fabrizio Orlandi
orlandi@iai.uni-bonn.de

Judie Attard
Judie.Attard@iais.fraunhofer.de

John Psarras
john@epu.ntua.gr

[1] National Technical University of Athens, Iroon Polytechneiou 9, 10682 Zografou, Athens, Greece

[2] UBITECH Ltd, Thessalias 8 & Etolias 10, 15231 Chalandri, Athens, Greece

[3] Fraunhofer FOKUS, Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany

[4] University of Bonn, Regina-Pacis-Weg 3, 53113 Bonn, Germany

in a qualified way. Availability and matching of diverse data sources become more crucial and therefor the need for standards-based tools is growing.

In most cases public sector information is not published in a machine-processable format that would allow data re-users from the public and private sector to automate combining public data with other public or proprietary data sources (Analysis Report of Public Sector Data and Knowledge Sources" 2012). In exacerbation to this situation, many public organizations provide open data as unstructured documents or reports, thus making the effort and cost of linking and utilizing this information unbearable for data consumers. The vast majority of Public Data sources do not provide the datasets in a standard format which would support true semantic enrichment and interlinking of data (such as Resource Description Framework, RDF). The very few Public data initiatives that do follow the Linked Data (W3C on Linked Data 2016) Paradigm mostly focus only on the metadata for the discovery layer of the datasets, therefore leaving the significant value of analysing and linking the actual information contained in the data itself by large unexploited.

During the last years, significant research activities have appeared that focus on industrial relevant scenarios, such as the LATC (LATC Project 2016) and the LOD2 (LOD2 Project 2016) projects that aim to contribute high-quality interlinked versions of public semantic web datasets and promoting their use in new cross-domain applications by developers across the globe (Verborgh et al. 2014). In the context of these efforts and emerging tools, while there is considerable support for linked data in other issues, such as storage (Virtuoso, Sesame), linkage (Silk Framework), discovery and publishing (SPARQL standard) and even visualization of RDF graphs (LodLive, CubeViz), there are very limited options for renovating existing data into Linked Data. Currently available solutions either support specific structured data formats, such as spreadsheets (XLWrap) and relational databases (D2R, Triplify) or provide RDF representations of data for specific sources (DBpedia). Lastly, most existing work related to exploring and visualizing RDF is limited on concrete domains and concrete datatypes and is mainly focused towards academic researchers that are familiar with the semantic web technologies.

In this context, a unified solution for transforming and renovating existing data sources, regardless of the original data format, would greatly enhance the ability of public organizations to provide usable, machine-processable linked data, while offering citizens and SMEs the opportunity to combine and link existing public sector information with privately-owned data in the most resourceful and cost-effective manner. Towards this direction, however, there is also a strong need for supporting consumers unfamiliar with the linked data paradigm through interfaces that hide the underlying complexity and allow the re-use of existing software applications. In the following chapter a further analysis will be conducted on such needs by performing an overall overview of the available linked data tools.

The objective of this paper to reveal the common barriers and challenges in view of the current state-of-the-art in linked data technologies and proposes a technical framework that aims to hide the underlying complexity of linked data while maintaining and promoting the interlinking capabilities enabled by the Linked Data Paradigm. To attain this objective we followed a structured approach: first, we looked at workflows and use cases that involve Linked Data, and settle on a selection of categories of tools to be analysed. For the selection of the tools we first gathered a comprehensive list of all currently online linked data tools. The final set of selected tools was decided based on their popularity, impact, maintenance status and last update date as well as discussion with data providers, public sector employees and developers from SMEs. For each category of tools, we discussed with our target group a number of concepts and tools that appear to be relevant for their work. As each category differs in its purpose and problem solving features, it comes with different specific qualitative and quantitative evaluation criteria. Then, for each category of tools, we made a feature comparison table, and complement it by a list of remarks for individual tools. The state-of-the-art analysis aimed to identify the shortcomings of the currently available tools and helped us derive requirements for the proposed transformation framework.

The state-of-the-art analysis on existing Linked Data tools is presented in section 2, focusing on different aspects of data renovation (transformation to linked data, semantic enrichment through named entity recognition and vocabulary services), management (linked data database engines) and exploitation (querying linked data, linked data exploration, visualization and analytics). In section 3 we focus on identifying the obstacles, limitations and overall effort in linked data technology adoption by both existing as well as new Public Sector Information (PSI) projects.

In section 4, the LinDA Workbench solution and its applicability in the context of public sector is presented, where it is argued that it can be a basis to face limitations identified in section 3.

Finally, in section 5 the role of Linked Data in the PSI and the potential role of LinDA Workbench is drawn, while future steps and necessary actions are identified and presented.

2 State-of-the-art analysis on potential linked data tools to be exploited

In this section we will made a state-of-the-art analysis on potential Linked Data tools that are currently available for exploitation. We mainly organised the tools in the following categories:

- **Transformation tools to Linked Data**: This set of tools allows users to publish their data as linked data through mapping rules. Usually users provide a connection to the database, select the data table they want and make their mappings to vocabularies.
- **Vocabulary Services**: These services provide a catalog of standardized or popular linked data vocabularies
- **Named-entity recognition engines**: Named-entity recognition engines usually take a word or text as an input and substitute words or phrases with linked data Entities (e.g. the word 'country' becomes the DBpedia entity Country)
- **Linked Data Database Engines**: This set of tools allow the storage and management of linked data sources, similar to the Database management systems.
- **Linked Data Query Tools**: Linked Data Query tools allow users to perform searches against the linked data databases.
- **Visualization and Exploration Tools**: This set of tools are usually typical visualization or data browsing tools that support the graph representation and features of linked data.
- **Analytic Tools**: Similar to the visualization tools, the analytic tools perform analytic algorithms and functions adjusted to the graph structure of linked data.

The following chapter presents these tools in detail.

2.1 Transformation tools to linked data

Transformation tools towards Linked Data take structured or semi-structured data sources and transform them into semantically enriched data (Michel et al. 2013). Converters may be part of a one-time migration effort or part of a running system which provides a semantic web view of a given application. W3C provides a comprehensive list of current converters per various formats and types (Excel, File Systems, OAI-PHM, SQL, UML, etc.). Each of the RDF converters tries to be as specific as possible in identifying the semantics associated with the data that is being converted. ***XLWrap*** for instance, is a spreadsheet-to-RDF wrapper which is capable of transforming spreadsheets to arbitrary RDF graphs based on a mapping specification. It supports Microsoft Excel and Open Document spreadsheets such as comma- (and tab-) separated value (CSV) files and it can load local files or download remote files via HTTP. ***Triplify*** follows a similar approach to publish RDF and Linked Data from relational databases. Triplify is based on mapping HTTP-URI requests onto relational database queries expressed in SQL with some additions. Triplify transforms the resulting relations into RDF statements and publishes the data on the Web in various RDF serializations, in particular as Linked Data. ***D2R Server*** is a tool for publishing the content of relational databases on the Semantic Web, where data is modelled and represented in RDF. D2R Server uses a customizable D2RQ mapping to map database content into this format and allows the RDF data to be browsed and searched. Requests from the Web are rewritten into SQL queries via the mapping. ***Anything To Triples*** is a library, a web service and a command line tool that extracts structured data in RDF format from a variety of Web documents. Currently it supports the following input formats: RDF/XML, RDFa, Microformats (Adr, Geo, hCalendar, hCard, hListing, hResume, hReview, License, XFN and Species) and CSV). Lastly, ***Sparqlify*** - In its core functionality, it is a server program that accepts SPARQL queries, translates them to SQL queries according to a specification in Sparqlify Mapping Language (SML), places the SQL queries to a specified database, translates the reply to RDF, again directed by the SML specification, and sends back the result. – The SML script describes how the database content is to be translated to RDF. An overview of these tools can be found in the following table:

Some interesting conclusions can be drawn by the comparison in Table 1:

- The transformation process almost always requires a transformation mapping as an additional input to the actual data source. This mapping is written either in the RDF format or some other custom format, including the SPARQLification Mapping Language (SML) in the case of SPARQLify, a language with elements from N3, SPARQL and SQL. Any23 does not require a mapping file, but in this case the transformations to RDF are either well defined and automatable (RDFa/ hCard/hCalendar to RDF or transformations between different RDF formats) or, in the case of CSV, ad-hoc vocabularies are used, which lead to the construction of low quality linked data.
- None of the above tools use a Graphical User Interface (GUI) for either the whole transformation or at least the composition of the mapping document. They are primarily used either as web services and/or through Command-Line Interface (CLI).
- When combined, these two insights above mean that practically a "Linked Data Transformation" Expert with advanced technical skills would be necessary for the adoption of linked data, coding or formally defining transformation mappings on a per-case basis and invoking the actual transformation from interfaces with low user-experience quality.

2.2 Vocabulary services

Due to the fact that vocabulary re-usability is vital for the growth of Linked Data, various vocabulary repositories have been created in the past. The added value of such repositories consists of the ability to search for various vocabularies in a

single database, to get access to vocabulary metadata like author and upload information, the ontology's scope and intended usage (Prateek et al. 2010), as well as user feedback and metadata about mappings that interlinks different vocabularies. This way, between others, repository users can determine which vocabulary is suitable for different scenarios, understand the design rational behind the vocabularies and determine whether a set of vocabularies retrieved from the repository can be used together. The utilization of such repositories facilitates the transformation of existing datasets, allowing Linked Data tools to operate on originally tabular data sources.

Prefix.Cc (http://Prefix.Cc/) Prefix.cc is a W3C tool developed to simplify a common task in the work of RDF developers: remembering and looking up URI prefixes. Prefix.cc, built without ability to accept user feedback, advanced search features or interlinking between different vocabularies, is more of an index than a full-scale repository. It nevertheless contains references to a wide variety of open vocabularies.

Linked open vocabularies Linked Open Vocabularies (LOV) is described on its homepage as an entry point to the growing ecosystem of linked open vocabularies (RDFS or OWL ontologies) used in the Linked Data Cloud. LOV offers advanced features like vocabulary interlinking and vocabulary metadata, like author information and basic vocabulary descriptions. Through LODStats, Linked Open Vocabularies offers statistics about the usage of vocabularies in various linked open datasets.

2.3 Named-entity recognition engines

Named Entity Recognition (NER) is a tool in the tool chain that is typically employed in Natural Language Processing (NLP). There, it contributes to the annotations of the source text by mapping it to pre-defined categories, such as names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc.

NER plays a vital role in the realm of semantic web as it supports Linked Data principles and the growth of the Web of Data. This can be enabled by extracting parts from a text and annotating them with Unique Resource Identifier (URI) that represent Linked Data resources available in sites such as DBpedia or Wikidata. In doing so, rich semantic queries can be issued over a broad decentralized knowledge base. Popular NER tools include: ***AlchemyLanguageAPI*** - is a commercial tool that provides content owners and web developers with a rich suite of content analysis and meta-data annotation tools. AlchemyAPI uses statistical natural language processing technology and machine learning algorithms to analyze content, extracting semantic meta-data: in-formation about people, places, companies, topics, languages, and more. ***OpenER*** - the projects main goal is to provide a set of ready to use tools to perform some natural language processing tasks, free and easy to adapt for Academia, Research and Small and Medium Enterprise to integrate them in their workflow. More precisely, OpenER detects and disambiguates entity mentions and performs sentiment analysis and opinion detection on the texts. In a similar manner **Gate** aims to solve almost any text processing problem including NER. ***DBpedia Spotlight*** is a tool for automatically annotating mentions of DBpedia resources in text, providing a solution for linking unstructured information sources to the Linked Open Data cloud through DBpedia. It can also be used for building your solution for Named Entity Recognition, Keyphrase Extraction, Tagging, etc. amongst other information extraction tasks. DBpedia Lookup – is a web service that can be used to look up DBpedia URIs by related keywords. Related means that either the label of a resource matches, or an anchor text that was frequently used in Wikipedia to refer to a specific resource matches (for example the resource http://DBpedia.org/resource/United_States can be looked up by the string "USA").

2.4 Linked data database engines

Triple stores are databases used for the storage and retrieval of triples through semantic queries written in the SPARQL language. Triple stores may be purpose-built or customized

Table 1 Comparison of RDF transformation tools

Tool	Input	Mapping / Mapping Format	Interface	Literature
XLWrap	Excel, CSV	Yes / RDF	Web Service	(Langegger and Wolfram 2009)
SPARQLify	RDB	Yes / SML	SPARQL Endpoint, CLI	(Sparqlify: a SPARQL-SQL rewriter 2016)
Triplify	RDB	Yes / Custom format	HTPP	(Auer 2009)
D2R	RDB	Yes / RDF	Web Service	(Bizer and Cyganiak 2006; Bizer 2003)
Anything to RDF (Any23)	(x)HTML with RDFa, hCard/hCalendar, CSV	No	Web Service, CLI	(Introduction to Apache Any23 2016)

Table 2 Comparison of NER services

Tool/Service	License	Methodology	Literature
Alchemy API	Commercial	Content Analysis, Sentiment Analysis	(Blei et al. 2003)
OpenER	Apache License v.2	Natural Language Processing, Sentiment Analysis	(García-Pablos et al. 2013; Manteli et al. 2014)
Gate	GNU	Natural Language Processing, Text Analysis, Information Extraction	(Introduction to Linguistic Annotation and Text Analytics 2009)
DBpedia Lookup	Apache License v.2	Text Analysis	(Bizer et al. 2009)
DBpedia Spotlight	Apache License v.2	Text Analysis, Textual Context Awareness	(Bizer et al. 2009; Mendes 2011)

general purpose databases, but they must always be optimized for the storage and retrieval of triples. In addition to queries, triples can usually be imported/exported using RDF and other formats. Comparison, evaluation and classification of triple stores have been an active research field since the early days of linked data (Rohloff et al. 2007). Many modern triple stores actually operate as quad stores, saving the information context alongside with every statement (subject, predicate and object) stored in the database (Tables 2, 3, 4 and 5).

AllegroGraph AllegroGraph is a commercial RDF graph database and application framework for the storage and querying of RDF data. It can be deployed as a standalone database server and offers interfaces (entry points) for remote access where the communication between the AllegroGraph Server and client processes is realized through HTTP. AllegroGraph's storage capabilities are not limited to storing RDF-data; instead, due to its generic storage model, it can store any form of graphs whose elements can be represented as tuples consisting of node elements s and o, an edge p, and additional data. AllegroGraph provides a number of additional features such as geospatial or temporal reasoning, social network analysis, or federation. **GraphDB -** GraphDB (formerly BigData) is a clustered RDF store for ordered data (B + Trees) and designed to run as a server on commodity hardware. Additional scale can be achieved by simply plugging in more data services dynamically at runtime, which will self-register with the centralized service manager and start managing data automatically. Scale-out is achieved via dynamic key-range partitioning of the B + Tree indices. **Jena -** Jena TDB is a component of the Jena Semantic Web framework and available as open-source software released under the BSD license. It can be deployed as a server on Java systems and accessed through the Java-based Jena API library as well as through a set of command line utilities. Jena TDB also supports the SPARQL query language for RDF data. **Oracle Spatial Module 11 g** - The Spatial module of Oracle's Database Enterprise Edition 11 g also supports RDF data management. Oracle 11 g is a commercial product, which is free of charge for non-commercial purposes. It is deployed as a database server and provides programming interfaces (connectors) for SQL (SQLplus, PL/SQL) and Java-based applications (via Jena/Sesame and JDBC). For each model (graph) Oracle generates a separate triple table containing references to subject, predicate, and object values, which are stored in an external dictionary table. Hence, it follows a generic triple table layout. **Sesame** - Sesame is an open source Java framework for processing RDF data. This includes parsing, storing, inferencing and querying of/over such data. It offers an easy-to-use API that can be connected to all leading RDF storage solutions. It allows you to connect with SPARQL endpoints and create applications that leverage the power of Linked Data and Semantic Web. Sesame offers two out-of-the-box RDF databases (the in-memory store and the native store), and in addition many third party storage solutions are available. The framework offers a large set of tools to developers to leverage the power of RDF and related standards. Sesame fully supports the SPARQL 1.1 query and update languages for expressive querying and offers transparent access to remote RDF repositories using the exact same API as for local access. ***Stardog*** is a semantic graph database, implemented in Java. It provides support for RDF and all OWL 2 profiles providing extensive reasoning capabilities and uses SPARQL as a query language. APIs are available for the Jena and Sesame as well as SNARL, the Stardog Native API for the RDF Language. **Virtuoso** - OpenLink Virtuoso is a SQL-ORDBMS and Web Application Server hybrid (aka Universal Sever) that provides SQL, XML, and RDF data management in a single multithreaded server process. Virtuoso is also an OWL Reasoner. Virtuoso delivers RDB2RDF (formerly known as SQL2RDF) directly; via the Sponger and its cartridges, it can also deliver RDF from GRDDL, RDFa, microformats, and many more inputs. Virtuoso includes a Live SPARQL Query Service Endpoint in all installations, although the SPARQL implementation differs from the SPARQL 1.1 specification.

In our discussion we have identified the various approaches that each triplestore decides to follow for their own reasons. Some of those are scalability over reasoning, other provide a

Table 3 An overview of existing linked data database engines

Triplestore	License	Interface	Reasoning	Literature
AllegroGraph	Commercial	(Partial) SPARQL 1.1, Sesame API	RDFS++, Geospatial, Social network analysis	(Aasman 2006)
GraphDB	Commercial	Java/JavaScript API	OWL (partial), custom reasoning rulesets	(Güting and Hartmut. 1994)
Jena	Apache License v.2	Java API/ SPARQL (via Fuseki plugin)	RDFS, OWL (partial)	(Apache 2012; Apache 2013)
Oracle Spatial Module 11 g	Commercial	SQL, JDBC	Provided by plugins (PelletDB, TrOWL)	(Oracle Spatial Database System 2016)
Sesame	BSD License	SPARQL, Sesame API, Sesame Workbench	Provided by plugins	(Broekstra et al. 2002)
Stardog	Commercial	SPARQL, SNARL API, Sesame API	OWL (partial)	(Stardog Enterprise Data Unification with Smart Graphs 2016)
Virtuoso	GNU GPL	SPARQL, SIMILE API, ODBC, JDBC, Virtuoso Conductor	OWL (partial)	(Erling and Mikhailov 2009)

greater focus on the last, others like Virtuoso give an emphasis on plugins that could cover a greater area of applications and this discussion can continue for long. For this specific reason, W3C initiated a discussion amongst all interested and involved stakeholders in the form of a working group[1] that would decide what are the basic principles for any platform, database, etc. to be considered part of the Linked Data ecosystem. After two years of interesting conversations, there is now a first official recommendation[2] that summarises the key ideas and gives specific guidelines for the future of this area.

2.5 Linked data query tools

Querying Linked Data Sources is not considered a straightforward process for the non-savvy users, given the interconnections and complexity of Linked Data structure. This category comprises applications or tools that enable a user to generate a SPARQL (Prud'Hommeaux and Seaborne 2008) query without requiring knowledge about the underlying complexity. This means that users are not expected to know the SPARQL query language, Semantic Web concepts, or the schema of the dataset in question. These tools could include one or more of the following features:

- The tool can enable the user to query any dataset available through an endpoint.
- The user is able to explore a dataset in order to better identify the specific data required for the use case at hand.
- Various methods can be used to generate a query, such as autocomplete or drag and drop, however it is important that the user does not need to know the SPARQL query language or the structure of the dataset to be queried.

This category includes tools such as:

a). **Flint SPARQL Editor** and **YASQE** that provide features such as auto-completion that is sensitive to the syntactic context at the cursor position, highlighting and error detection and auto-completion for properties and classes, which are populated by what is actually in the dataset thus significantly saving time to create a query. The user still has direct access to the full SPARQL syntax,

b). **Facete2**, **Lodlive, gFacet, OntoWiki Browser**. These categories of SPARQL querying engines provide a visual, intuitive way for users to explore SPARQL endpoints. More specifically they allow exploration of RDF data by combining graph-based visualization with faceted filtering techniques. Therefore the user doesn't have to be familiar with the SPARQL syntax in order to explore a specific endpoint and at the same time has a visual representation of its query and the underlying structure of the data,

c). **Quepy**: These querying applications promote the semantic power of linked data in comparison with other traditional data structures. These applications are able to transform natural language questions (Zaihrayeu et al. 2007) to queries in a database query language. They can be customized to different kinds of questions in natural language and database queries. Therefore the user doesn't need to know SPARQL syntax and is able to perform complicated queries that take into advantage semantic reasoning in a natural language.

For the sake of completeness, we mention here another W3C effort which is the RDF Data Shapes (http://www.w3.org/2014/data-shapes) which aims to provide a standard for describing structural constraints for RDF and validating against them. For this reason they

[1] http://www.w3.org/2012/ldp/wiki/Main_Page#Status
[2] http://www.w3.org/TR/ldp/

Table 4 Linked data querying tools

Name	License	Interface	Domain	Literature
Flint Editor	MIT License	Text editor + Basic auto-completion	-	(Flint SPARQL Editor 2016)
YASQE	MIT License	Text editor + Advanced auto-completion	-	(Rietveld and Hoekstra. 2013)
Facete2	GNU GPL v.2	Graphical (map-based)	Geospatial information	-
LodLive	MIT License	Graphical (form-based)	Limited number of datasets	(Camarda et al. 2012)
gFacet	GNU GPL	Graphical (tabular), data exploration oriented	-	(Heim et al. 2008)
OntoWiki	GNU GPL v.2	Graphical (wiki-style), data exploration oriented	-	(Auer et al. 2006)
Quepy	BSD	Python API, Web service	DBpedia, Freebase	(Quepy, A framework to convert natural language to database queries. 2016)
AllegroGraph Gruff	Commercial	Graphical (graph-based)	Only AllegroGraph endpoints	(Gruff: Grapher-Based Triple-Store Browser for AllegroGraph 2016)

currently explore extensions of SPARQL that would better facilitate their ideas and vastly enriching the Linked Data toolkit (Prud'hommeaux et al. 2014).

Some interesting insights can be drawn from the information in the above table:

- Many of the querying tools are focused on specific datasets or types of endpoints, with an interface matching the needs of the specific data. PSI applications would have to adhere to the scheme of these datasets to utilize these tools, which may or may not be appropriate in different usage scenarios.
- Exploration oriented Linked Data querying tools are generally simpler than other querying tools, as they focus on content views based on the existing categorization in the dataset itself, often with some basic filtering capabilities, but do not have the ability to produce advanced, general purpose queries.
- Graph-based GUI querying tools are the most powerful, as they closely match the structure of linked data. However they can easily become overwhelming for users without any previous experience with Linked Data who have been generally working with tabular formats (CSV, Excel or RDBMSs).

2.6 Visualization and exploration tools

This category includes visualization tools for tabular data and RDF data following the Linked Data principles. **Vizboard** is a workbench which provides lay-users with a set of tools to visualize arbitrary Semantic Web data. Powered by a facets browser and the VISO ontology the tool enables placing of several charts on a workspace aiming at giving comprehensive description of a dataset. Implementing a pipeline workflow model Vizboard enriches visualizations with semantics and offers a wide range of customization tools. **CODE Vis Wizard** is a visualization component of the CODE project. The idea behind the software is to parse input data to RDF Data Cube Vocabulary and recommend a suitable visualization with the help of the developed visualization ontology. The Vis Wizard aims at simple but reasonable dataset visualization relevant for users unexperienced in the Semantic Web field. **Rdf:SynopsViz** – constructs hierarchical representation of RDF data and computes statistical parameters of a dataset. The authors of the tool outline certain features such on-the-fly hierarchy construction, faceted browsing and an attempt to measure data quality via dataset metadata, allowing five types of charts, a timeline and a treemap. Rdf:SynopsViz, however, has an intricate interface that mainly targets domain experts.

Table 5 A synopsis of linked data visualization tools

Tool	Domain	Characteristics	Queries / Analytics	References
Vizboard	Generic use	User interaction built in the visualization itself	–/–	(Voigt et al. 2013b)
CODE Vis Wizard	Generic use	10–15 built in datasets	Data aggregation/–	(Mutlu et al. 2013). (Mutlu and Höfler 2014)
LOD Visualization Tool	Class hierarchies	LDVM support	–/Payola analytics support	(LOD visualization tool 2016)
OntoWiki CubeViz	Generic use	Only RDF Data Cube support	–/–	(Martin et al. 2015)

Linked Open Data (LOD) Visualization Tool is a service based on the Linked Data Visualization Model (LDVM) model. The service is able to visualize a hierarchy of classes and properties, and connection points between arbitrary concepts and view instances with the highest in−/out-degrees. Additionally, the dataset might be fed to another tool, Payola, in order to produce an analysis of the dataset. **OntoWiki CubeViz** is a faceted browser tool based on OntoWiki, which works with the data presented in RDF Data Cube Vocabulary. CubeViz can visualize statistical data in various formats, e.g. a map, column charts, a pie chart. **Payola** is another tool that follows the LDVM concept and implements analytical functions to provide lay-users with a suite of visualizations. Payola encourages collaboration between users, e.g., experts can edit visualizations and SPARQL queries and lay-users can consume a result. Having a wide range of charts and diagrams it is, however, unclear what visualization is the most suitable for a chosen dataset. A synopsis of the properties of these tools can be found in the table below:

2.7 Analytic tools

Several open source data mining tools are available today. The most commonly used include R, Weka, Knime and RapidMiner. It should be noted that no analytics framework exists that supports consumption and production of Linked Data (e.g. there is no support for RDF as an input/output format). Focus is given on the integration of existing open-source tools –namely R and Weka into a holistic platform that facilitates the consumption and production of Linked Data, taking into account and fulfilling the aforementioned requirements for supporting future interdisciplinary research (Michel et al. 2013).

3 Issues and challenges for renovating public sector information to linked data

Despite the envisioned benefits of Linked Data, a number of significant challenges and issues prevent the massive adoption of the Linked Data tools presented in the previous section by public data providers. These issues are analysed in the following sections:

3.1 Adoption effort and complexity of existing linked data tools

In spite of its power and flexibility, current Linked Data tools suffer from inborn factors that diminish the potential for an intuitive and efficient -in terms of performance- linked data experience. We analyze the challenges and issues per tool type:

3.1.1 Transformation tools

Although there has been extensive work conducted around the transformation towards Linked Data and although most open source tools seem to have the potential to be adopted by Public bodies, ease of Linked Data technology integration is hindered by the lack of support and lack of complete features (maturity) along with decreasing development efforts (Michel et al. 2013). The transformation process requires a considerable amount of effort and time from skilled semantic and domain experts. Moreover, existing tools do not come with a GUI and target Linked Data experts instead of non-technical users. Furthermore, it has been noted that few tools cover heterogeneous source formats in one instance (there is either support for RDB alone or tabular data) except for Tarql, SparqlMap and Datalift (Scharffe et al. 2012). If any, ontology finding is limited to a select number of ontologies. Some tools provide ontology inference from the database schema (Utlrawrap, Morph-RDB) although this deviates from the concept of reusing existing ontologies. Therefore for different data structures more than one transformation tool may be required, thus increasing significantly the complexity and effort of the transformation process.

3.1.2 Querying linked data

The development of querying applications and wizards that facilitate access and linking between billions of ad-hoc triples is still significantly more challenging than querying predefined schemas of relational databases and tabular data. Linked Data, typically represented using RDF as a data format, in most cases relies on its own, RDF-specific query language, the SPARQL (Prud'Hommeaux and Seaborne 2008). SPARQL queries are based on triple patterns and RDF can be seen as a set of relationships among resources; SPARQL queries provide one or more patterns against such relationships. Moreover, SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. SPARQL also supports aggregation, sub-queries, negation, creating values by expressions, extensible value testing, and constraining queries by source RDF graph (SPARQL 1.1 Query Language 2013). Despite its expressiveness and power, users outside the group of semantic web experts are not acquainted with the SPARQL query language. Although SPARQL syntax resembles SQL, it still has a steep learning curve due to vast difference between matching triple patterns queries and querying predefined, static schemas of relational databases and tabular data. Common users (including enterprise users, data analysts, data engineers and others) are used to query and process tabular data with clearly defined entities and data boundaries, whereas linked data carry the complexity of logic programming languages. Another significant issue with current SPARQL engines is the

unattainability of available options and classes at the time of query. Even advanced users who understand complex SPARQL syntax have to manually look for the available URIs, classes, properties, resources and values of an existing Linked Data repository, usually by referencing the documentation or the actual data at a separate interface or web page, before they can actually construct the query. This significant barrier is a deal-breaker for most non-expert users and gets even worse if the user wants to actually interlink more than one linked data repository (usually addressed as SPARQL endpoints). In contrast to the actual term "Linked Data" that was coined back in 2006 (Berners-Lee 2006), querying more than one SPARQL endpoints was only made available through the SPARQL 1.1 Federated Query (SPARQL 1.1 Query Language 2013) in 2013. Over the last years, lightweight linked data-like languages and formats have been introduced such as JSON-LD (Berners-Lee 2006; Sporny et al. 2014) that facilitates linked data access and processing through the usage of the popular JSON format. These developer-friendly data formats however still carry these inherent challenges of linked data mentioned above. Lastly, the distributed, graph-based architecture of Linked Data poses a number of significant challenges in terms of performance since the matching patterns run across millions or even billions of triples. Verborgh, Ruben, et al. (Prateek et al. 2010) further outlines the availability and scalability issues of querying public SPARQL endpoints.

3.1.3 Visualization tools

The majority of mature visualizations tools do not provide native support to RDF graphs, in order to support Linked Data and explore new analysis' prospects and interlinking possibilities, in comparison to isolated, legacy business data structures. The limited examples of RDF-enabled visualization engines (e.g. CubeViz) focus only on specifically structured RDF files (e.g. Cube format) thus requiring different visualization engines for different types and structures of linked data. In addition, most of the tools lack of automatic features that support the user in creating visualizations. This makes most of the tools especially suited to Linked Data experts and not to more common lay-users. This is what makes current linked data visualization tools especially difficult to adopt for public bodies and public data providers.

3.2 Trust and provenance of SPARQL endpoints

Trust and provenance is one of the most important issues when publishing PSI data as linked data. According to Karvounarakis et al. (Karvounarakis et al. 2010) understanding how an RDF triple was created or where it was copied from, is crucial to assess the data quality and strengthen data accountability. This functionality essentially calls for representing and reasoning on the provenance of replicated and incomplete sets of RDF triples manipulated by SPARQL queries worldwide. This is a significant issue especially for public bodies and authorities whose services demand the highest level of accountability and trust. Unfortunately, a great number of data providers during the last years have abandoned support and maintenance of public SPARQL endpoints thus harming the trust between consumers and open Linked Data providers.

3.3 Mentality shifting in public sector and training overhead

Public sector offices face many challenges as they attempt to integrate linked data technologies into their daily business of operations. It is important that training consultants for public servants are aware of those challenges, understand the implications of those challenges in the public domain, and are able to generate answers to the challenges. Due to bureaucracy as well as complex legislation that characterizes public services, public bodies are prone to adopt new technologies such as Linked Data at a much slower pace than private organizations and enterprises. Equally troubling, there are usually no resources or motivation at a strategic level for training on such advanced technical issues.

4 The LinDA framework and approach

Although the availability of openly accessible public data has dramatically increased and the technology and infrastructure are being developed at a fast pace, the issue of positioning Public Sector Organizations in the conditions of adequately leverage LOD to improve their interoperability and Quality of Service is still far from being solved. Affordability of costs and efforts for converting and renovating PSI is still a challenge, and therefore the main consumers of PSI data – SMEs that are the backbone of the European economy, are lagging behind the exploitation of the full potential of LOD. To address this significant gap, we propose the LinDA framework (LinDA Project 2016), which is a collection of independent, yet integrated tools that aim to support Public Sector organizations as well as enterprises to effectively adopt LOD by providing a complete set of tools for publication, consumption and visualization of linked data that can be used with limited technical resources.

From a user perspective the main LinDA workflow can be summarized in three simple steps as illustrated in Fig. 1. More specifically the three workflow steps are:

Step 1: Turn data into RDF: Using LinDA Transformation Engine, Public Sector providers can publish their data as linked data in a few, simple steps. They can simply connect to their database(s), select the data

table they want and make their mappings to popular and standardized vocabularies. LinDA assists even more by providing automatic suggestions to the mapping through its Vocabulary API.

Step 2: Query/link your data: With the LinDA Query Builder, Public Servants as well as data consumers can perform simple or complex queries through an intuitive graphical environment that eliminates the need for SPARQL syntax.

Step 3: Visualize/analyze your data: LinDA Visualization and Analytic engines can help data consumers gain insight from the data that the Public Sector generates. The added-value of LinDA visualizations and analytics in comparison with traditional tools is that it takes advantage of the enriched metadata contained within the Linked Data format to produce more meaningful visualizations. On top of that, users can gracefully link their data with any other private or public data therefore realizing an ecosystem of data extractions and visualizations, which can be bound together in a dynamic and unforeseen way.

According to this workflow the user can utilize either external public data or internal, private sources.

If the initial data source is in RDF format (N3, RDF/ XML, Turtle, Trix and TriG) the user can directly insert the data source to the available data sources of the LinDA Workbench. If the initial data source is in another format (relational database, csv, tsv, etc.) the LinDA Workbench guides the user to the Transformation Engine in order to transform the data into the RDF format, with the utilization of popular Linked Data vocabularies. Once in RDF, the user can then visit the list of data sources and activate one of the available LinDA Services. More specifically, the user has the option to

a). query it through the Query Designer,
b). run linked data analytics,
c). create linked data visualizations,
d). manage and update existing datasets.

By querying internal or external RDF data sources, the user can extract specific information and save the query for future use. Similar to the list of data sources, the user is able to perform specific operations to the list of the saved queries including visualizing, analysing and editing a query item through the respective tools. Visualization configurations and Analytic processes are also saved in a dedicated list of respective items. Lastly, the LinDA Workbench provides a ubiquitous full-index search for all entities of the system (Vocabularies, Data Sources, Visualization configurations, saved Analytic processes and Queries). Overall, the workflow made possible via the LinDA Workbench allows Public sector organizations and ministries to seamlessly transition from traditional data formats into the world of Linked Open Data.

The following sections present the components of our proposed framework, each with its major features. These separate tools are integrated into a common web application for easy installation in various deployment scenarios. Having this in mind, the overall high-level architecture for the LinDA Workbench is now presented.

4.1 LinDA transformation engine

The Transformation Engine (TE) is the pivotal part of the provisioning framework. The purpose of the LinDA TE is renovating (semi-)structured private and Open Data by transforming them into semantically described and enriched data. Two strains of transformation are currently supported: the transformation of relational data from SQL databases, and the transformation of tabular data from CSV files. In the user interface, the user is guided through the process of the transformation by a wizard-style "Mapping Designer" that includes explanatory text to aid with the renovation process. The Mapping Designer generates a mapping file as a result. This mapping file captures the necessary relations from input data elements (i.e. table, column, row, and literal) to RDF elements (i.e. < rdfs:Class>, <rdfs:Property>). The mapping file is expressed in the SML, which provides a SPARQLlike syntax to address elements that are foreign to the Resource Description Framework, i.e. tables and columns.

5 RDB-to-RDF transformation

When connected to a SQL database as input, the TE presents the schema of the database and lets the user specify the mapping according to the SML. The TE leverages the functionality of the Sparqlify project to execute the transformation and

Fig. 1 Simplified workflow from a user perspective

providing the RDF output data. The resulting RDF data can therefore either be dumped to a triple store, or be served by the SPARQL-SQLrewriter engine of Sparqlify as a SPARQL endpoint. The rewriter expects a SPARQL query on the RDF data basis, and rewrites it to a SQL query which can be optimized and executed efficiently within the RDBMS.

6 CSV-to-RDF transformation of tabular data from CSV files

When providing a CSV or TSV file as data input, the TE presents the data accordingly, treating it as relational data with just one table. In this case, Sparqlify can be leveraged as well. However, an ad-hoc query compilation is not reasonable, as the resulting extraction operation would not perform well on plain text files. The resulting data can be dumped to disk or to a triple store instead.

When dealing with relational or tabular data, a straight-forward way to interpret the structure of the data is to regard tables or tabular files as data about a specific subject; database or CSV columns describing a certain property about the subject; and where each row in the table constitutes a statement about one individual. Therefore, tables should be annotated with one or more RDF classes, whereas columns should be annotated with at least one RDF property.

The end user is assisted in the production of Linked Data by using the Mapping Designer in a guided and user-friendly manner. As the quality of the produced Linked Data is paramount, a user-supervised, semi-automated process is adopted as opposed to a fully automated process, where the domain expert has several means to adjust the transformation and give preference from already existing "higher" vocabularies. The mapping designer provides an interface to search for relevant RDF elements among a selection of frequently used and well-described vocabularies using the external Linked Open Vocabularies web service16. On the other hand, the Mapping Designer is based on simple assumptions concerning the structure of the output data (as outlined above). This limits the expressiveness of the RDF graph yet positively simplifies the complexity of the user interface, thus lowering the complexity of usage for the domain expert.

6.1 Vocabulary and transformation metadata

LinDA Vocabulary and Metadata repository fulfils the need for a repository of Linked Data vocabularies, accessible by users and other tools in the LinDA ecosystem. Linked Data vocabularies are collections of relationships between real world data, known as properties, and object categories, known as classes, typically expressed in RDF. Based on Linked Data vocabulary repositories that already exist and are open to the public, the LinDA vocabulary repository presents many advantages for enterprises and nontechnical users. The presented architecture alongside with its implementation facilitate both enterprise users as well as other LinDA applications and Linked Data applications in general in the processes of creating and publishing semantically rich linked datasets in a more robust and expandable way. The repository can be installed within the LinDA workbench. Each vocabulary repository installation retains an independent metadata database that is synchronized to a central, master vocabulary repository. Every vocabulary repository instance periodically communicates to the master LinDA vocabulary repository to search for new vocabularies and metadata or updates to existing vocabularies. Knowledge inside the master repository is gathered from different catalogues around the web, with local repositories also having the ability to add repositories locally or remove unwanted ones. Thus the vocabulary repository of a public organization or ministry can be enriched with private vocabularies that are stored in a private database, from where they are exposed to all other LinDA tools and modules.

The user is encouraged to select classes and properties calls to an exposed RESTful Web based API allow applications to specify search terms and categories. The API shall then operate as an oracle, suggesting a list of entities that may be suitable in the specified context, ranked according to factors like community rating and other vocabulary statistics provided by the master vocabulary repositories, like statistics from the LODStats framework (Prateek et al. 2010). The repository also offers a web interface where users can browse the various vocabularies, along with all the entities defined by them, including classes and properties. Interconnections between different entities are described, and new information like vocabulary usage examples has been added, in order to allow users better understand the intended usage of each vocabulary. It should be noted that all information is automatically extracted by the vocabulary definitions, using SPARQL queries. Features like auto-detection of search term language and immediate translation also facilitate non-English speakers' needs, and can also improve the suggestions produced for databases and CSV files with terms in other languages. Vocabulary definition source documents are also accessible in all major RDF formats, and a visualization of the RDF graph allows users to better perceive each vocabulary's structure. The repository uses various libraries in its web application's implementation. Django is a lightweight framework written in python that emphasizes the Don't Repeat Yourself (DRY) principle. The jQuery library is used to facilitate AJAX requests and to make the application more dynamic all around. Finally, with ElasticSearch it becomes possible to support multiple search requests to the web application and the Vocabulary API at the same time.

6.2 Publication and consumption tools

The Linked Data Publication and Consumption Framework aims to assist public data publishers and consumers in analysing and interlinking public sector information with external, consumer's data. In Fig. 2 the basic workflow of the processes within the framework is shown.

Our process starts with dataset exploration. Here a user can explore existing open datasets, both with regards to their content and also with regards to the underlying schema. The user then has two options, either directly executing a SPARQL query in the SPARQL Query Tool or use the Query Builder Tool to generate the query. Finally, the user can proceed to convert the result set in a number of different formats and export results. This conversion allows users to import data from linked open datasets into their native system. The results can also be exploited further through the LinDA Visualisation and Exploration Ecosystem, and the LinDA Analytics and Data Mining Services.

The main tools provided within this framework, which provide the functionality described above, are the following:

RDF2Any API This API provides the functionality for both the dataset exploration and converting of RDF data into a number of target formats, including JSON, RDB, CSV, and a generic format. The latter conversion allows a user to upload a template and execute a conversion specific to his/her needs. The RDF2Any API can also be easily extended with other converters.

SPARQL query tool Intended for experienced users, this tool allows users to directly input SPARQL queries which are executed on defined external endpoints. The users can then preview, convert the results in a number of formats, and export the set of results. The format conversion functionality is obtained through the consumption of the RDF2Any API.

Query designer tool As opposed to the SPARQL Query Tool, this tool is aimed for users who are not familiar with the SPARQL query language. It provides drag and drop and auto-complete features that allow a user to easily build the desired query and access the required data. This tool also caters for users who either don't know the specific dataset(s) that contains the data that suits their needs, or otherwise would like to explore existing open datasets. Through this tool, the user can navigate through classes, subclasses, instances, and properties. Similarly, to the SPARQL Query Tool, the user can then preview, convert, and export the resultset. The Query Builder tool also operates through the consumption of the RDF2Any API which provides the required functionality.

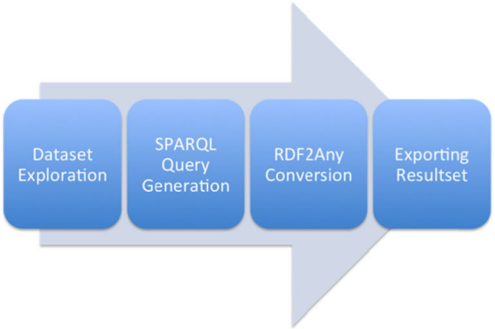

Fig. 2 Workflow within the linked data publication and consumption framework

6.3 Visualization tool

The role of LinDA Visualization is to provide a largely automatic visualization workflow that enables SMEs to visualize data in different formats and modalities. In order to achieve this, a generic web application is being developed based on state-of-the-art Linked Data approaches (Brunetti et al. 2013; Dadzie and Rowe 2011; Voigt et al. 2013a) to allow for visualizing different categories of data, e.g. statistical, geographical, temporal, arbitrary data, and a largely automatic visualization workflow for matching and binding data to visualizations. The Visualization tool consists of two main components:

- The Explore and Select Data component;
- The Visualization component.

The Explore and Select Data component allows users to pre-select the data to be processed and provides a concise preview of the data through a tabular representation. The input data formats supported are RDF (any serialization and vocabulary) and CSV. The selection performed with this component is used as an input of the Visualization component.

The Visualization component is responsible for the creation of various plots, charts and maps and offers opportunities to customize visualization options or save and share the graphical results. The visualization is powered by a recommendation algorithm which aims at suggesting the most suitable visualization type according to different features of the input data. The algorithm consists of the following fundamental steps:

1. Defining the dimensions and scales of measurement of the selected data;

2. Building possible allocations, or combinations, of the dimensions;
3. Comparing the constructed allocations with premade patterns, which describe the required parameters of a visualization component (e.g. the formats for the X and Y axes of a line chart);
4. Ranking of the most relevant allocations and building the list of preferred visualizations.

An essential part of the recommendation engine is the visualization ontology: a high-level model of the visualization workflow, its components and its particular instances (e.g. line chart, map, etc.). The implemented tool adopts Ember.JS as a main operational platform combining a comprehensive user interface with a relatively lightweight backend. By adopting existing open source libraries the following visualizations to an extensible modular infrastructure have been integrated: Line Chart, Bar Chart, Column Chart, Pie Chart, Scatter Chart, Area Chart, Bubble Chart, and Map.

6.4 Analytics and data mining

The LinDA Analytics and Data Mining component supports the realisation of analysis based on the consumption and production of Linked Data. A library of basic and robust data analytic functionality is provided through the support of a set of algorithms, enabling SMEs to utilise and share analytic methods on Linked Data for the discovery and communication of meaningful new patterns that were unattainable or hidden in the previous isolated data structures. High priority is given to the user friendliness of the provided interfaces based on the design of specialized workflows per algorithm category (e.g. workflows for supervised learning techniques such as classification and regression/forecasting algorithms and unsupervised learning techniques such as clustering and pattern discovery (association) algorithms). The LinDA Analytics and Data Mining component supports RDF as an input and output format, while the provided output from the analysis is interlinked with the input RDF source or sources. In addition to the interlinking, information regarding the type of the analytic process executed – including configuration and description issues – is saved and made available to the end users for further use. RDF input is supported through the selection of existing queries that have been prepared through the LinDA Query Designer, their execution and the production of the input RDF data source for the analytic process. The next step regards the selection of the appropriate algorithm to be executed and the configuration of the analytic process to be followed. As already mentioned, a set of algorithms are integrated and custom workflows are prepared based on default parameters. Specifically, integration of the Weka open-source tool and the R open source project for statistical computing is realized. A default configuration is proposed at each case, while there is flexibility for executing the algorithm with customized configuration. Upon the finalisation of the configuration phase, the algorithm is executed in a transparent way for the end user and the output results are produced. Different types of output formats are supported based on the peculiarities of each algorithm. Thus, the analytics output can be an RDF file, plain text or even specialized graphs (Fig. 3).

As already mentioned, in case of RDF output, interconnection of the output dataset to the input dataset and the executed analytics process is provided. This interconnection is based on the design of a specific ontology that describes the overall analytic process followed and the relationship among the input and output data sources. Thus, the end users are able to have access to the analytics processes executed in the past along with the input and output files used or produced. This functionality is considered critical in cases that re-evaluation of algorithms/processes have to be realised in the examined business scenarios where input data are updated. As stated earlier, the design and deployment of the analytics and data mining component is realised with main objective its user-friendliness towards the public sector employees. However, even if specific workflows are designed per algorithm category that hide part of the complexity, detailed knowledge of the algorithm's functionality and configuration parameters is required. Since, in most cases, a set of analysis has to be realised by interplaying also with the parameters, the end users are responsible for properly configuring the considered parameters per algorithm according to their needs.

6.5 Architectural properties of the LinDA workbench

From a technical perspective, the LinDA tools presented earlier in this chapter are modular and can be deployed and used separately in other projects, as independent tools. With a view to achieving a seamless workflow between the tools, however, the LinDA Workbench has been developed. The LinDA Workbench orchestrates the tools and handles the main communication with the selected triple store. Moreover, it acts as an integrated environment with the ability to host new tools through an extensible python-based system. The LinDA Workbench provides an authentication and authorization framework above all integrated tools, handling user accounts, groups, permissions and cookie-based user sessions. Data Source Management is the core submodule of the LinDA Workbench, where users can create, edit or delete graph-based RDF models or provide a link to a public SPARQL endpoint. Moreover, LinDA workbench provides a context-menu with all available services (explore, visualize, etc.) for each data source and global configuration page for all hosted

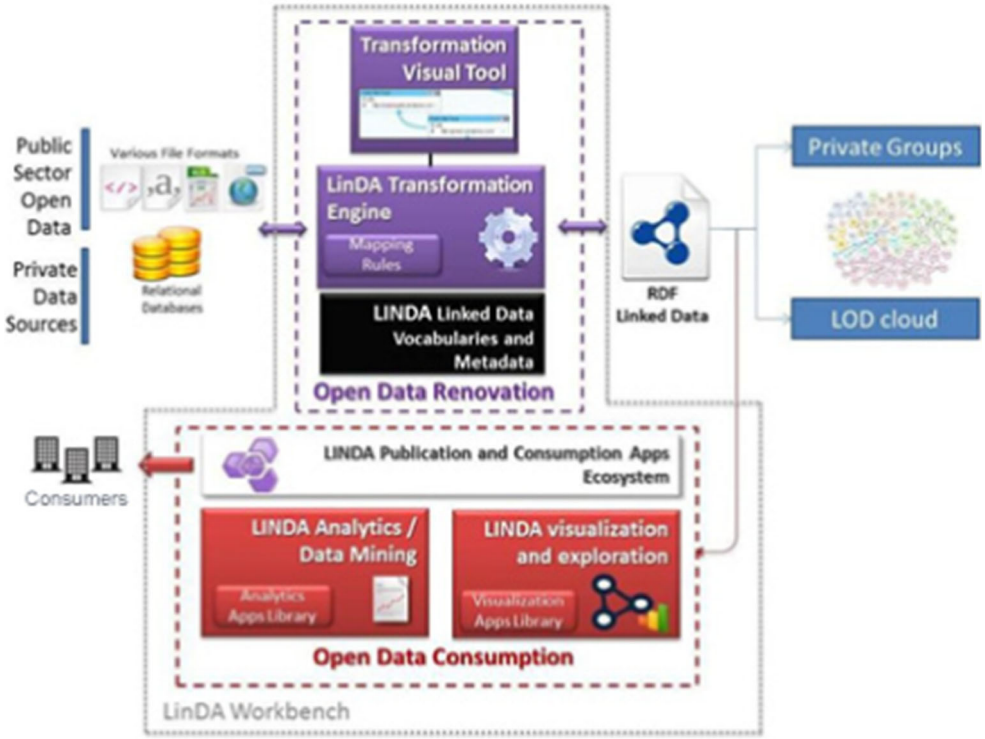

Fig. 3 The LinDA workbench high-level architecture

tools. Lastly LinDA Workbench provides an administration interface for the management of all entities of the system (Data Sources, Queries, Analytics, Users, Groups, etc.) As such the main workflow of the tools is illustrated in Fig. 2.

7 Conclusions

This paper investigated the challenges of linked data tools adoption from the Public Sector organizations in view of the current state-of-the-art of linked data technologies. The paper then presented a linked data technical framework (LinDA) that aim to hides the underlying complexity of the SPARQL syntax while preserving or even strengthening the power of the linked data paradigm. The LinDA framework can be used in order to minimize the required adoption effort of linked data technology and offer more options and flexibility to the end-users of PSI data. It could be argued that the power provided through the combination of linked data technologies and analytics can facilitate the Public Sector to realise advanced analysis and gain insights in the context of policy formulation.

Taking into account the experience gained through this research, it could be argued that the main benefits from adopting the suggested LinDA framework are:

- the simplification of the data collection, representation and interlinking part supporting the renovation of public sector information along with the exploitation of data available in private data sources;
- the design of set of queries able to provide in an easy and flexible way periodical views with information that is considered helpful for the daily operation of public office, agency or organization;
- the realization of analysis (including the production of analytics and visualisations) –not exclusively by data scientists- that can take advantage of the power of linked data and lead to insights in a much easier and less time consuming manner, compared to the relevant processes in the past;
- the production and maintenance of interoperable data by reconciliating against hundreds of popular / standardized vocabularies that are being updated or the usage of owned domain-specific vocabularies;
- the capacity for using it without the need for expertise on data management, analytics and software engineering domains, taking into account the low complexity and short learning curve of the functionalities provided by the suggested framework.

Furthermore, a set of suggestions can be provided to new adopters of linked data tools:

- start from the clear description of the objectives of the analysis and identify the questions that have to be answered;
- identify data sources that can be proven useful for the analysis and prepare a mapping of the potential interlinkings among different data sources that can reduce the data management burden;

- examine the quality of the provided data prior to the analysis in order to identify shortcomings, missing values, outliers etc. that may impact the analysis results; based on the collected information try to get based on the most qualitative and reliable data sources;
- identify vocabularies that can be used for the appropriate transformation of the available data to RDF; if you are not aware of such vocabularies, you may be based on the automated suggestions provided by our framework LinDA.;
- be based on reliable SPARQL endpoints in order to be able to get the required information when needed and without lot of delays and connectivity problems;
- prepare carefully your queries since they may possibly accompany an analysis for a large timing period and, thus, it may reduce a lot the effort required for ad-hoc modifications and
- collaborate with a data scientist –if needed- towards the design of the exact algorithms to be used in the analysis phase, the configuration of the algorithms, as well as the interpretation of the analysis results.

Future work includes the construction of an efficiency assessment framework for a large-scale evaluation of the LinDA framework in public sector organizations. Work is going forward on using the suggested consumption framework to evaluate and further improve the tools based on the feedback by ministries, public organizations and data consumers in Europe.

References

Aasman, J. (2006). *Allegro graph: RDF triple database*. Cidade: Oakland Franz Incorporated.

Analysis Report of Public Sector Data and Knowledge Sources", (2012) ENGAGE Project, available at http://www.engage- project.eu

Apache Jena (2012) "The apache software foundation."

Apache Jena (2013)"Reasoners and rule engines: Jena inference support." *The Apache Software Foundation*.

Auer, Sören, et al. (2009) "Triplify: light-weight linked data publication from relational databases." *Proceedings of the 18th international conference on World wide web*. ACM.

Auer, Sören, Sebastian Dietzold, and Thomas Riechert. (2006) "OntoWiki–a tool for social, semantic collaboration." *The Semantic Web-ISWC 2006*. Springer Berlin Heidelberg, 736–749.

Berners-Lee, Tim (2006–07-27). "Linked Data—Design Issues". W3C, available at: https://www.w3.org/DesignIssues/LinkedData.html

Bizer, Christian. (2003) "D2r map-a database to rdf mapping language.", available at: http://wifo5-03.informatik.uni-mannheim.de/bizer/d2rmap/www2003-D2R-Map.pdf

Bizer, Christian, and Richard Cyganiak. (2006) "D2r server-publishing relational databases on the semantic web." *Poster at the 5th International Semantic Web Conference*.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., & Hellmann, S. (2009). DBpedia-a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web, 7*(3), 154–165.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003) Latent Dirichlet allocation. Journal of Machine Learning Research (March 2003). 3:993–1022.

Broekstra, Jeen, Arjohn Kampman, and Frank Van Harmelen. (2002) "Sesame: A generic architecture for storing and querying rdf and rdf schema." *The Semantic Web—ISWC 2002*. Springer Berlin Heidelberg, 54–68.

Brunetti, J. M., Auer, S., García, R., Klímek, J., & Nec˘aský, M. (2013). *Formal Linked Data Visualisation Model, in proceedings of international conference on information integration and web- based applications & services*. Vienna: Austria.

Camarda, Diego Valerio, Silvia Mazzini, and Alessandro Antonuccio. (2012) "LodLive, exploring the web of data." *Proceedings of the 8th International Conference on Semantic Systems*. ACM.

Dadzie, A. & Rowe, M., (2011) *Approaches to Visualising Linked Data: A Survey,* semantic web, Nr. 2, p. 89–124.

Erling, Orri, and Ivan Mikhailov. (2009) "RDF Support in the Virtuoso DBMS."*Networked Knowledge-Networked Media*. Springer Berlin Heidelberg, 7–24.

Flint SPARQL Editor, (2016) available at: http://openuplabs.tso.co.uk/demos/sparqleditor

García-Pablos, A., Cuadros, M., & Rigau, G. (2013). *OpeNER demo: Open polarity enhanced named entity recognition, 6th language technology conference*. Poznán, Poland: LTC.

Gruff: Grapher-Based Triple-Store Browser for AllegroGraph, (2016) available at: http://franz.com/agraph/gruff/

Güting, Ralf Hartmut. (1994) "GraphDB: Modeling and querying graphs in databases."*VLDB*. Vol. 94.

Heim, Philipp, Jürgen Ziegler, and Steffen Lohmann. (2008) "gFacet: A Browser for the Web of Data." *Proceedings of the International Workshop on Interacting with Multimedia Content in the Social Semantic Web (IMC-SSW'08)*. Vol. 417.

Introduction to Apache Any23, (2016) available at: https://any23.apache.org/

Introduction to Linguistic Annotation and Text Analytics (2009) Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers, 2009;4:1, 1–165.

Karvounarakis, Grigoris, Zachary G. Ives, and Val Tannen. (2010) "Querying data provenance." *Proceedings of the 2010* ACM SIGMOD international conference on management of data. ACM.

Langegger, A., & Wolfram, W. (2009). *XLWrap–querying and integrating arbitrary spreadsheets with SPARQL*. Berlin Heidelberg: Springer.

LATC Project, (2016) available at http://latc-project.eu/

LinDA Project (2016), available at http://www.linda-proiect.eu.

LOD visualization tool, (2016) available at: http://lodvisualization.appspot.com/

LOD2 Project, (2016) available at http://lod2.eu/

Manteli, C., Hoof B. van, H. van Vliet and W. van Duinkerken, (2014) Overcoming challenges in global software development: The role of brokers, IEEE eighth international conference on research challenges in, pp 3542–3545, Morocco.

Martin, M., Abicht, K., Stadler, C., Auer, Sö., Ngomo, A.-C. N. & Soru, T. (2015). CubeViz – Exploration and Visualization of Statistical Linked Data. *Proceedings of the 24th International Conference on World Wide Web, WWW 2015*.

Mendes, Pablo N., et al. (2011) "DBpedia spotlight: shedding light on the web of documents." Proceedings of the 7th International Conference on Semantic Systems. ACM.

Franck Michel, Johan Montagnat, Catherine Faron-Zucker: (2013) A survey of RDB to RDF translation approaches and tools. Rapport de Recherche. ISRN I3S/RR 2013–04-FR. November 2013.

Mutlu, B. and Höfler ,P.. (2014) Suggesting visualisations for published data.

Mutlu, B., Höfler, P., Sabol, V., Tschinkel, G., & Granitzer, M. (2013). Automated visualization support for linked research data. In S. Lohmann (Ed.), *I-SEMANTICS (Posters & Demos), volume 1026 of CEUR workshop proceedings, pages 40–44* .CEUR-WS.org

Oracle Spatial Database System, (2016) available at: http://www.oracle.com/technetwork/database/enterprise-edition/overview/spatial11gr2-datasheet-134190.pdf

Prateek Jain, Pascal Hitzler, Amit Sheth, Kunal Verma, Peter Z (2010) Ontology Alignment for Linked Open Data. Yeh. In proceedings of the 9th International Semantic Web Conference, ISWC, Shanghai, China, 6496:402–417

Prud'Hommeaux, Eric, and Andy Seaborne. (2008) "SPARQL query language for RDF." W3C recommendation 15.

Prud'hommeaux, Eric, Jose Emilio Labra Gayo, and Harold Solbrig. (2014) "Shape expressions: an RDF validation and transformation language." *Proceedings of the 10th International Conference on Semantic Systems*. ACM.

Quepy, A framework to convert natural language to database queries. (2016) Available at: https://pypi.python.org/pypi/quepy/

Rietveld, Laurens, and Rinke Hoekstra. (2013) "Yasgui: Not just another sparql client."*The Semantic Web: ESWC 2013* Satellite events. Springer Berlin Heidelberg, 78–86.

Rohloff, Kurt, et al. (2007) "An evaluation of triple-store technologies for large data stores." *On the Move to Meaningful Internet Systems 2007:* OTM 2007 workshops. Springer Berlin Heidelberg.

Scharffe, F., Atemezing, G., Troncy, R., Gandon, F., Villata, S., Bucher, B., Hamdi, F., Bihanic, L., Kepeklian, G., Cotton, F., Euzenat, J., Fan, Z., Vandenbussche P.-Y., Vatant, B.: (2012) Enabling linked-data publication with the datalift platform. In (AAAI 2012) Workshop on semantic cities, Toronto.

SPARQL 1.1 Query Language, (2013) W3C Recommendation 21 March 2013, available at https://www.w3.org/TR/sparql11-query/

Sparqlify: a SPARQL-SQL rewriter, (2016) available at: http://aksw.org/Projects/Sparqlify.html

Sporny, Manu, et al. (2014) "JSON-LD 1.0." W3C Recommendation, January 16, 2014, available at: https://www.w3.org/TR/json-ld/

Stardog Enterprise Data Unification with Smart Graphs., (2016) available at: http://docs.stardog.com/

Verborgh, Ruben, et al. (2014) "Web-scale querying through linked data fragments." Proceedings of the 7th Workshop on Linked Data on the Web (LDOW2014.

Voigt, M., Pietschmann, S., & Meißner, K., (2013a) A Semantics- based, End-user-centered Information Visualisation Process for Semantic Web Data, in Semantic Models for Adaptive In- teractive Systems, Springer, p. 83–107.

Voigt, Martin, Stefan Pietschmann, and Klaus Meißner. (2013b) "A semantics-based, end-user-centered information visualization process for semantic web data."*Semantic Models for Adaptive Interactive Systems*. Springer London, 83–107.

W3C on Linked Data, (2016) available at http://www.w3.org/standards/semanticweb/data

Zaihrayeu, I., et al. (2007). *From web directories to ontologies: Natural language processing challenges*. Berlin Heidelberg: Springer.

Spiros Mouzakitis is a research analyst for National Technical University of Athens (NTUA). He holds a Ph.D from the National Technical University of Athens, Electrical and Computer Engineering. His current research is focused on decision analysis in the field of eBusiness and eGovernment, as well as optimization systems and algorithms, decision support systems and data and enterprise interoperability.

Dimitris Papaspyros is a Ph.D candidate and works as a developer/researcher for National Technical University of Athens (NTUA). He holds a Bachelor's Degree in Computer Science from National Kapodistrian University of Athens and is currently attending an MSc in Computational and Internet Technologies and Applications in the Harokopio University.

Michael Petychakis is a Ph.D candidate and works as a developer for the National Technical University of Athens (NTUA). He holds a diploma in Electrical and Computer Engineering from the National Technical University of Athens. His current research is focused on APIs, Linked and Open Data and Data Science.

Sotiris Koussouris holds a PhD Degree in Information Systems and Business Process Management, a Dipl. Eng. in Computer and Electrical Engineering and a MBA in Techno-Economic systems. He has over twelve years experience in information systems and telecommunication technologies with special skills in areas like Data Technologies, Semantics, eGovernment technologies and applications, Interoperability Science, Social media, Business Process Re-engineering and Business Process Modelling. Over the last years, he has worked on numerous EC and domestic funded projects including the LinDA Project and the ENGAGE Data Infrastructure acting as a project and/or technical manager in various other projects as well. He has published more than 15 articles in scientific journals and more than 80 papers in scientific conferences, while he has co-authored 1 book on Enterprise Interoperability.

Anastasios Zafeiropoulos received the Dipl.-Ing. and Ph.D. degrees from the School of Electrical and Computer Engineering, National Technical University of Athens, and the M.Sc. degree in public policy and management from the Athens University of Economics and Business. He is currently a Senior R&D Architect with Ubitech Ltd. He has acquired great experience in the area of next generation networking, green IT, software defined functionalities, and at the design of context-aware, data management, and linked-data oriented solutions. He has co-authored over 40 publications in high level international journals and conferences.

Eleni Fotopoulou received the Diploma degree in electrical and computer engineering from the National Technical University of Athens, and the Diploma degree in therapeutic pedagogy from the University of Thessaly. She is currently a Software Engineer with the R&D Department, Ubitech Ltd. She has been involved in various international, EU-funded, and national funded projects in the domain of e-Government, data management, and next generation networking solutions. She has great expertise in data analytics, linked/open data and big data technologies, e-Government and security solutions, and cloud applications orchestration frameworks.

Lena Farid is a Researcher and Project Manager at Fraunhofer FOKUS. Her current research focuses on eGovernment and Open and Linked data, Data Driven analysis and impact evaluation.

Fabrizio Orlandi is a postdoctoral researcher at the University of Bonn and the Fraunhofer IAIS institute in Bonn, Germany. He received the PhD degrees from the National University of Ireland in Galway (NUIG) in 2014. His main research interests include Linked Open Data, Data Value Chains, Open Government Data, Smart Data Ecosystems, Social Semantic Web, User Modelling and Personalisation.

Judie Attard is a Ph.D. student at the University of Bonn. She is currently also working as a researcher within the EIS group at Fraunhofer. Her main research interests include semantic web technologies, open data, and open government initiatives. She has authored or co-authored a number of publications in the fields of (linked) open data, data value chains, and context-aware systems.

John Psarras is Professor in the School of Electrical and Computer Engineering of National Technical University of Athens (NTUA) and the director of EPU NTUA. He has been a project coordinator and / or senior researcher in a number of EC projects such as LINDA, ENGAGE, ENSEMBLE, GIC, MOMENTUM, acquiring over fifteen years of experience in the areas of decision support and monitoring systems and requirements analysis and knowledge management.