Requirements on Linked Data Consumption Platform

Jakub Klímek
Charles University in Prague
Faculty of Mathematics and Physics
klimek@ksi.mff.cuni.cz

Petr Škoda
Charles University in Prague
Faculty of Mathematics and Physics
skoda@ksi.mff.cuni.cz

Martin Nečaský
Charles University in Prague
Faculty of Mathematics and Physics
necasky@ksi.mff.cuni.cz

ABSTRACT

The publication of data as Linked Open Data (LOD) gains traction. There are lots of different datasets published, more vocabularies are becoming W3C Recommendations and with the introduction of DCAT-AP v1.1 and the emergence of the European data portal and a multitude of national open data portals, lots of datasets are discoverable and accessible using their DCAT-AP metadata in RDF. Yet, the consumption of LOD is lacking in comfort and availability of tools that would exploit the benefits of LOD and allow users to discover, access, integrate and reuse LOD easily, as promised by the promoters of LOD and supposedly paid by the additional effort put into the 5-star data publication by the publishers. Compared to the consumption of 3-star CSV and XML files, the consumption of LOD is still quite complicated and the LOD benefits are not exploited enough nor visible enough to justify the effort for many publishers. In this paper we identify 40 requirements which a Linked Data Consumption Platform (LDCP) should satisfy in order to be able to exploit the LOD benefits in a way that would ease the LOD consumption and justify the additional effort put into LOD publication. We survey 8 relevant and currently available tools based on their coverage of the identified requirements.

Categories and Subject Descriptors

H.3.5 [**Online Information Services**]: Data sharing; H.3.5 [**Online Information Services**]: Web-based services

Keywords

Linked Data, RDF, consumption, discovery, visualization

1. INTRODUCTION

A considerable amount of data represented in a form of Linked Open Data (LOD) is now available on the Web. More and more data publishers are convinced that the additional effort put into the last 2 stars of the now widely accepted and promoted 5-star Open Data deployment scheme[1] will bring the promised benefits to the users of their data. The publishers rightfully expect that the users will appreciate the 5-star data much more than the 3-star CSV files or 2-star Excel files given that the proper publication of 5-star data is considerably harder and more costly to achieve.

[1]http://5stardata.info/

WWW2016 Workshop: Linked Data on the Web (LDOW2016).

Naturally, the expectations of users of LOD are quite high as the promoters of LOD promise that the non-trivial effort put into publishing the data as LOD will bring benefits mainly to them, the users of the data. These benefits, compared to 3-star Open Data, include better described and understandable data and its schema, safer data integration thanks to the global character of the URIs, better reuse of tools and libraries, more context thanks to the links and the ability to (re)use only parts of the data. They are supposed to come at the cost of learning RDF, dealing with potentially broken links and understanding the risks of presenting data from foreign datasets. However, in the best case scenario, what the user gets now is usually either a link to a rather large dump in Turtle or RDF/XML, a limited SPARQL endpoint or the possibility of dereferencing a URI and getting an HTML or RDF description of a data entity. While this is appreciated by Linked Data experts and enthusiasts, more regular users used to 3-star open data will not be able to enjoy the expected benefits. They expect a level of experience higher than what they get by using tools for "less open" 3-star data such as Microsoft Excel[2], Google Sheets[3] and many more for tabular data and tools such as Altova XMLSpy[4], oXygen XML editor[5] and many more for XML data. These tools provide a comfortable way of working with such data and therefore, the experience of working with 5-star Linked Open Data is expected to be even better. This is supported also by the presentation of the 5-star scheme where the fourth star is for using URIs and, preferably, RDF, and the only downside should be the need to learn RDF. However, this is not the case so far because compared to 3-star data there is a lack of quality tools for LOD processing at the consumer end.

LOD is mainly a data publishing format, therefore, the data published in this format usually need to be transformed back to a representation usable by conventional tools which need e.g. the speed of relational databases or the power of existing data mining or machine learning techniques. Linked Open Data exists for some time now and lots of the necessary techniques and vocabularies which can be used to facilitate its proper consumption are W3C Recommendations now. There is a multitude of tools which work with LOD in specific ways. What is missing is a platform, or at least a set of compatible and reusable tools, that could be recommended to the users, who are familiar with 3-star open data and used to work

[2]https://products.office.com/en/excel
[3]https://www.google.com/sheets/about/
[4]http://www.altova.com/xmlspy.html
[5]https://www.oxygenxml.com/

with it, as the platform that will enable them to immediately enjoy the promised benefits of LOD.

The contribution of this paper is a set of requirements on a Linked Data Consumption Platform (LDCP) which we identified as crucial in order to be able to demonstrate the benefits of Linked Open Data to the consumers. We consider three types of stakeholders, a journalist, who wants to create articles based on Linked Data, a data analytic, who wants to use the data published as Linked Data in a tool of his choice and a developer, who wants to build services on top of the platform. Majority of the requirements are based on existing W3C Recommendations and popular research areas. Moreover, we compare 8 existing tools, which are presented in the literature as tools for Linked Data consumption, with respect to the given requirements. Some of them provide visualizations features, which is a kind of consumption. Some of them provide other features also important for consumption (e.g., data transformation, advanced data loading from different kinds of data sources, data previews, etc.).

This paper is structured as follows. In Section 2 we provide a motivating example of a user scenario in which a journalists wants to consume LOD and expects a platform that will enable him to enjoy the promised LOD benefits. In Section 3 we describe the identified requirements and in Section 4 we survey existing tools and their coverage of the identified requirements. In Section 5 we survey related studies of Linked Data consumption options and in Section 6 we conclude.

2. MOTIVATING EXAMPLE

To motivate our work, let us suppose a potential user of Linked Open Data, who could ideally use the Linked Data Consumption Platform (LDCP) to gain the benefits promised by the promoters of publishing data as LOD. The potential user is a data journalist used to working with non-RDF open data on the web such as HTML pages, CSV tables and Excel files and XML files. The goal of our user is to collect data about population of cities in Europe and display an informative map in an article he is working on. The intended audience of the article are statisticians who are used to CSVs, so the user wants to also publish the underlying base data as a CSV file attached to the article. The user now wants to use Linked Data because he heard that it is the highest possible level of openness according to the, now widely accepted and promoted, 5-star Open Data deployment scheme. The user expects the experience working with LOD will be somehow better than the experience working with 3-star data such as CSV or XML files once he learns RDF. Let us suppose that the user has learned RDF and understands the above mentioned side effects of the distributed character of LOD. Now the user expects to get better access to data and a better experience working with the data than if the data was available in a 3-star CSV or XML file. Naturally, the first thing which the user expects is a tool that can be used to work with such data and enjoy the promised benefits of LOD. Ideally, the tool would be an integrated platform that supports the whole process of LOD consumption from identification of data sources to the resulting visualization or processed data ready to be used by other tools while using the benefits of LOD where applicable.

Intuitively, the user needs to find data about cities in Europe, their location so that they can be placed on a map and data about their population. Here, LOD can help already since recently a number of national open data catalogs emerged and are being harvested by the European data portal[6] which utilizes the DCAT-AP v1.1[7] vocabulary for metadata about datasets and exposes a SPARQL endpoint, which can be queried for e.g. keywords (city) and formats (RDF - SPARQL, Turtle, JSON-LD, etc.) LDCP could therefore support loading and querying of dataset metadata from DCAT-AP enabled catalogs and even contain a few well-known catalogs, such as the European data portal, preset so that the user can see and search some datasets right away.

Let us suppose that the user has identified a few candidate datasets that could be of use to him and needs to choose the ones that contain information needed for his goal. Let us also suppose that the metadata loaded from a catalog contains correct access information for each candidate dataset. A helpful functionality of LDCP could be various kinds of summaries of the candidate datasets and characterizations based both on the metadata of the dataset and the actual data of the dataset. For example, for each candidate dataset, LDCP could offer the vocabularies used, numbers of instances of individual classes and their interlinks, previews tailored to the vocabularies used, datasets linked form the candidate, datasets linking to the candidate, spatial and time coverage, etc. Using this information, the user would be able to choose a set of datasets containing the required information easily.

Another feature that could help the user at this time is recommendation of related datasets. Based on information on datasets already selected such as vocabularies used, entities present or their metadata, LDCP could suggest similar datasets to the user. Given a set of datasets to be integrated, a useful feature would be the ability to analyze those datasets in order to find out whether they have a non-empty intersection, e.g. whether the population information specified in one dataset actually links to the cities and their locations present in another dataset.

Let us now suppose that the user has all the datasets needed for his goal. There could be an interoperability issue new to Linked Data caused by the existence of multiple vocabularies describing the same domain. In our example, it could be that the dataset containing the locations of the cities could have the geocoordinates described using the *schema:GeoCoordinates*[8] class from the Schema.org vocabulary while e.g. the tools to be used next accept the *geo:Point*[9] class of the WGS84 Geo Positioning vocabulary. Since both of those vocabularies are well-known and registered at Linked Open Vocabularies (LOV)[10], LDCP could contain components for data transformation between them and offer them to the user automatically. Because of the growing number of vocabularies, a desired feature would be sharing of those transformation components in the community around LDCP.

Our potential user now has all the datasets needed in an interoperable format. What is left is to choose the entities and properties from those datasets that are needed for the goal and which can be omitted. This could be done in a graphical way e.g. used by graphical SPARQL query builders such as SPARQLGraph [24], but not necessarily complicated by the full expressiveness of SPARQL.

Finally, the data should be prepared for further processing.

[6] http://www.europeandataportal.eu/

[7] https://joinup.ec.europa.eu/asset/dcat_application_profile/asset_release/dcat-ap-v11

[8] http://schema.org/GeoCoordinates

[9] http://www.w3.org/2003/01/geo/wgs84_pos#Point

[10] http://lov.okfn.org/

This could mean RDF enabled visualizations such as in the Linked Data Visualization Model (LDVM) [14] or further processing outside of LDCP, which requires assisted export to tabular data in CSV, tree data in XML, generic graph data e.g. for Gephi[11] or even republication in RDF.

3. REQUIREMENTS

In this section, we identify and group the technical user requirements that we see as crucial for a Linked Data Consumption Platform (LDCP). We focus on requirements which can be satisfied by exploiting the benefits of consuming Linked Data compared to 3-star data and which, when implemented, can be used to demonstrate those benefits.

3.1 Dataset discovery

In order to start working with Linked Data, the user has to first identify relevant datasets, which is what we call dataset discovery. There are multiple ways in which LDCP could assist the user with this task.

The most obvious way is the ability to load dataset metadata from a data catalog. There is the DCAT[12] vocabulary and DCAT-AP v1.1, the DCAT application profile for European data catalogs, providing the vocabulary support for dataset metadata. There are existing data catalogs utilizing these vocabularies and providing access to the machine readable, standardized metadata. Examples of such data catalogs are the European Data Portal[13], integrating dataset metadata from various national data catalogs and the European Union Open Data Portal[14] providing information about the datasets from the institutions and other bodies of the European Union. LDCP should therefore support loading this machine readable information and provide means of searching the metadata to identify datasets needed by the user. LDCP should also have some well-known data catalogs preloaded, so that the user can start choosing datasets right away. A part of this discovery scenario is the ability of the user to add a dataset to LDCP through its URI, provided the URI is dereferencable. As a partial coverage of this requirement we may consider the non-RDF, but widely used CKAN API[15].

Requirement 1 (Catalog support) *Support for loading dataset metadata from dereferencable dataset URIs and data catalog APIs, data dumps and SPARQL endpoints which provide the metadata (preferably DCAT and DCAT-AP compatible metadata).*

The previous requirement assumes that publishers describe their datasets with manually created metadata. However, a more advanced way of dataset discovery is through implementation or support of a custom crawling and indexing service not necessarily relying on metadata found in data catalogs. Such a service can build its own index of datasets comprising metadata computed automatically on the base of the content of the datasets. The index may encompass used classes and predicates, labels of resources or their other properties present in the dataset [30]. Examples of such services are Sindice [21] or LODStats [7].

[11] https://gephi.org/
[12] https://www.w3.org/TR/vocab-dcat/
[13] http://www.europeandataportal.eu
[14] https://open-data.europa.eu
[15] http://docs.ckan.org/en/latest/api/index.html

Requirement 2 (Advanced discovery) *Support for loading dataset metadata from a third-party or own indexing service which computes the metadata automatically on the base of the dataset content.*

Discovered datasets marked as relevant by the user can than be used as a context for other kind of dataset discovery which searches for additional semantically related datasets. By the term "semantically related" we mean various kinds of semantic relationships among datasets. E.g. these can be datasets which provide statements about the same resources, use the same vocabularies or contain links to the datasets in the context.

Requirement 3 (Context-aware discovery) *Support for recommendation of additional datasets semantically related to datasets already discovered and selected by the user.*

3.2 Data input

A common problem of Linked Data tools out there is their limited support for standards of representation of Linked Data. LDCP should support all the standard ways of accessing Linked Data so that there are no unnecessary limitations of the available data. These include IRI dereferencing, SPARQL endpoint querying and the ability to load RDF dumps in all standard RDF 1.1 serializations[16], i.e. RDF/XML, Turtle, TriG, N-Triples, N-Quads, RDFa and JSON-LD. This functionality can be achieved relatively easily, e.g. by integrating Eclipse RDF4J[17] (formerly OpenRDF Sesame) or Apache Jena[18]. Inability to load all standardized formats from the dump (only some) will be classified as partial coverage of the requirement.

Requirement 4 (IRI dereferencing) *Ability to load RDF data by dereferencing IRIs using HTTP content negotiation and accepting all RDF 1.1 serializations.*

Requirement 5 (RDF dump load) *Ability to load data from an RDF dump in all standard RDF serializations, both from an URL and locally.*

Requirement 6 (SPARQL querying) *Ability to load data from a SPARQL endpoint.*

Besides these traditional ways of getting Linked Data, LDCP should support the recently published Linked Data Platform[19] specification for getting resources.

Requirement 7 (Linked Data Platform input) *Ability to load data from the Linked Data Platform compliant servers.*

In addition to RDF data sources, LDCP should support at least in some basic form the input of non-RDF data sources. For tabular data in CSV, this can be done automatically in a standardized way using Generating RDF from Tabular Data on the Web[20] W3C Recommendation. For tree-like data in XML, a support for XSLT transformation can be included and for JSON data, support for adding a JSON-LD[21] context can be provided.

[16] https://www.w3.org/TR/rdf11-new/#section-serializations
[17] http://rdf4j.org/
[18] https://jena.apache.org/
[19] https://www.w3.org/TR/ldp/
[20] https://www.w3.org/TR/csv2rdf/
[21] https://www.w3.org/TR/json-ld/

Requirement 8 (Non-RDF data input) *Ability to load non-RDF data sources using standardized methods.*

In addition to be able to load data in all standardized ways, LDCP should also provide a way of saying that the data source should be periodically monitored for changes, e.g. like in the Dynamic Linked Data Observatory [11]. This includes a specification of the periodicity of the checks and the specification of actions to be taken when a change is detected. The actions could vary from a simple user notification to triggering a whole pipeline of automated data transformations. This way, the user can see what is new in the data used for his goals and whether the data is simply updated or needs his attention due to more complex changes.

Requirement 9 (Monitoring of input changes) *Ability to periodically monitor a data source and trigger actions when the data source changes.*

3.3 Dataset preview

The discovery mechanism provides the user with a list of candidate datasets he can potentially find useful. Therefore, the user has to finally select the required datasets from these candidates. The dataset preview should help the user with this selection by presenting a short summary of each discovered dataset. This can be done in a number of different ways and again we focus on the benefits which arise from the Linked Data principles. Some of the Linked Data vocabularies recently became W3C Recommendations[22], which are de facto standards, and as such should be supported in LDCP. These include the Simple Knowledge Organization System (SKOS)[23], The Organization Ontology (ORG)[24], the Data Catalog Vocabulary (DCAT) and The RDF Data Cube Vocabulary (DCV)[25]. Having the knowledge of these vocabularies, LDCP can generate vocabulary specific summarization of each of the candidate datasets more effectively, giving the user more information for his decision making. For example, if LDCP detects the usage of DCV, it can summarize the dataset by giving an overview of the number of data cubes present, concepts used in component properties etc. while in the case of ORG it can summarize it by displaying the number of organizations with the numbers of their departments etc. Specifically for each of these vocabularies, the time and spatial coverage can be computed for each dataset where appropriate.

Requirement 10 (Preview – W3C vocabularies) *Support for dataset preview based on used vocabularies that are W3C Recommendations.*

While so far there are only few vocabularies that have reached the W3C Recommendation status, there is plenty of vocabularies that are also well-known and widely reused. Some of them are on the W3C track as Group notes, e.g. VoID[26] used to describe RDF datasets and vCard Ontology[27] for contact information. Others are registered in Linked Open Vocabularies [27] and are also popular, e.g. Dublin Core[28], FOAF[29], GoodRelations[30] and Schema.org[31]. Therefore, LDCP should also support previews of datasets which use these vocabularies as well.

Requirement 11 (Preview – LOV vocabularies) *Support for dataset preview based on used well-known vocabulary registered at Linked Open Vocabularies.*

A proper Linked Data dataset is described by its metadata using standardized vocabularies such as DCAT and VoID. These vocabularies provide support for basic characteristics like indication of vocabularies used, identification of main classes and properties and numbers of classes, properties and their instances. These are all relevant metadata that should be accessible to the user to support his decision making.

Requirement 12 (Preview metadata) *Provide dataset description and statistics metadata recorded in DCAT, DCAT-AP and VoID.*

Similarly to the discovery requirements, the user may require a preview compiled not on the base of descriptive and statistical metadata provided by the publisher about the dataset but on the base of metadata automatically computed on the base of the dataset content. As can be seen in the LOD Cloud diagram[32] and also in LODStats, the majority of datasets are not so big that they could not be effectively queried to get these basic statistics (e.g., classes and properties used, number of their instances, etc.) Therefore, it is a reasonable requirement that LDCP should be able to generate these statistics independently of what is contained in the metadata records.

Requirement 13 (Preview data) *Provide dataset description and statistics based on automatic querying the actual data.*

Part of the statistics based on the actual data could be the schema extracted from the data in the form of classes and properties used and their interlinks based on the presence of links among their instances. Such a schema can be very informative, however, its generation can take some time even for moderately sized datasets.

Requirement 14 (Preview schema) *Provide a schema extracted from the given dataset in a form of classes and properties among them.*

Besides metadata, statistics and used vocabularies there are many other criteria regarding data quality that should also be presented to the user to support his decision making. There is a recent survey of such techniques [29].

Requirement 15 (Quality indicators) *Provide quality measurements based on Linked Data quality indicators.*

3.4 Analysis of semantic relationships

The requirements discussed in the previous sections support the user in discovering individual datasets isolated from one another. However, the user usually needs to work with

[22] `https://www.w3.org/standards/techs/rdfvocabs#w3c_all`
[23] `https://www.w3.org/TR/skos-reference/`
[24] `https://www.w3.org/TR/vocab-org/`
[25] `https://www.w3.org/TR/vocab-data-cube/`
[26] `https://www.w3.org/TR/void/`
[27] `https://www.w3.org/TR/vcard-rdf/`
[28] `http://dublincore.org/documents/dcmi-terms/`
[29] `http://xmlns.com/foaf/spec/`
[30] `http://www.heppnetz.de/projects/goodrelations/`
[31] `http://schema.org`
[32] `http://lod-cloud.net/`

them as with one integrated graph of RDF data. Therefore, he expects that the datasets are semantically related with each other and that he can use these semantic relationships for the integration. If a dataset is not in a required relationship with other discovered datasets the user needs to know this so that he can omit the dataset from the further processing. We have briefly discussed possible kinds of semantic relationships in Requirement 3. However, while Requirement 3 only required LDCP to show datasets which are somehow semantically related to the selected one, now the user expects a deeper analysis of the relationships.

Each existing semantic relationships among discovered datasets should be presented to the user with a description of the relationship, i.e. the kind of the relationship and its deeper characteristics. The descriptions will help the user to understand the relationships and decide whether they are relevant for the integration of the datasets or not.

When the datasets share resources the deeper characteristics may involve the information about the classes of the resources, the ratio of the number of the shared resources to the total number of the resources belonging to these classes, compliance of the datasets in the statements provided about the shared resources, etc. When the datasets are linked together the deeper characteristics may be similar, e.g., information about the classes of the linked resources from both datasets complemented with the linking predicate, the ratio of linked resources to all resources which belong to the classes, etc. This can be extended by considering not only direct links expressed in a form of a single RDF statement but indirect links formed by a path of RDF statements. When the datasets neither share the resources nor there are links between them, the semantic relationship may be given by the fact that they contain resources of the same classes. The provided deeper characteristics for this kind of relationship may involve information about the shared classes, numbers of resources which can be unified and compliance of the datasets in the statements about those resources.

Requirement 16 (Semantic relationship analysis) *Provide characteristics of existing semantic relationships between datasets which are important for the user to be able to decide about their possible integration.*

As a supplemental requirement to the previous one is to support methods for automated or semi-automated deduction of semantic relationships between datasets. These methods may be useful when semantic relationships cannot be directly discovered on the base of statements present in the datasets but new statements must be deduced from the existing ones. There are tools like SILK [28], SLINT [20] or SERIMI [1] which support so called *link discovery* which means deducing new statements linking resources between two datasets. Another family of tools supports so called *ontology matching* [22] which is a process of deducing mappings between two compared vocabularies. These methods may help LDCP to provide the user with more semantic relationships.

Requirement 17 (Semantic relationship deduction) *Provide support for automated or semi-automated deduction of semantic relationships between datasets.*

3.5 Data manipulation

The user now needs to process the discovered datasets. As we discuss later in Section 3.7, this means either visualizing the datasets or exporting them for the purposes of processing in an external tool. In general, it means to transform the datasets from their original form to another one, which is necessary for such processing. We can consider transformations, which transform RDF representation of the datasets to another RDF representation or to other data models (relational, tree, etc.). In this section, we discuss the former transformations. The others are covered in Section 3.7. A transformation involves various data manipulation steps which should be supported by LDCP.

First, LDCP must deal with the fact that there exist different vocabularies which model the same part of reality. We say that such datasets are *semantically overlapping*. Let us note that semantic overlapping is a kind of semantic relationships discussed in Requirement 16. Therefore, it often happens that the vocabulary used in the dataset is different from the one required for the further processing. In such situation it is necessary to transform the dataset so that it corresponds to the required vocabulary. The transformation can be based on an ontological mapping between both vocabularies or it can be based on a transformation script specifically created for these two vocabularies. This requirement is realistic when we consider well-known vocabularies (e.g., FOAF. Schema.org, GoodRelations, WGS84_pos, etc.). There are not so many well-known vocabularies so LDCP implementors may create, provide or even share libraries of such mappings or transformation scripts.

Requirement 18 (Vocabulary-based transformations) *Provide transformations between well-known semantically overlapping vocabularies.*

It is also frequent that datasets do not re-use well-known vocabularies but use specific vocabularies created ad-hoc by their publishers. Even though transformation support for these vocabularies in terms of the previous requirement is not realistic (simply because there are too many ad-hoc vocabularies), the user can still expect that LDCP will provide some kind of support when such vocabularies appear among the discovered datasets. In particular, the users can expect that LDCP will at least discover that there exist possible semantic overlaps between the vocabularies. Discovering these semantic overlaps may be achieved by ontology matching techniques which are already involved in Requirement 17.

Having the possible semantic overlaps, the user then needs to specify the particular transformation on his own. Or he can expect that LDCP will be able to (semi-)automatically discover the transformation itself. Such discovery can be achieved by exploiting various techniques of ontology alignment [10][5].

Requirement 19 (Vocabulary alignment) *Provide (semi-)automated discovery of transformations between semantically overlapping vocabularies.*

Another often required kind of data manipulation is inference. Inference can be used when additional knowledge about the concepts defined by the vocabulary is provided which can be used to infer new statements. Such knowledge is expressed in a form of so called *inference rules*. Semantic mappings between vocabularies mentioned above are also a kind of inference rules but here we consider them in their broader sense.

Requirement 20 (Inference) *Support inference on the base of inference rules encoded in vocabularies (or ontologies) of discovered datasets.*

A specific kind of inference rules allows one to specify that two resources are the same real-world entity (i.e. `owl:sameAs` predicate). Having such inference rule means that statements about both resources need to be fused together. Fusion does not mean simply putting the statements together but also identification of conflicting statements and their resolution [19][18].

Requirement 21 (Resource fusion) *Support fusion of statements about the same resources from different datasets.*

Besides transformations, the user will typically need the standard operations for data selection and projection. LDCP can assist the user with specification of these operations by providing graphical representation of the required data subset, which may correspond to a graphical representation of a SPARQL query like in SPARQLGraph [24] and similar approaches.

Requirement 22 (Assisted selection and projection) *Support assisted graphical selection and projection of data.*

Besides the above described specific data manipulation requirements, the user may need to define own specific transformations expressed in SPARQL.

Requirement 23 (Custom transformations) *Support custom transformations expressed in SPARQL.*

Having the support for all kinds of data manipulations described above it is necessary to be able to combine them together so that the discovered datasets are transformed to the required form. The user may be required to create such data manipulation pipeline manually. However, we might also expect that LDCP complies such pipeline automatically and the user only checks the result and adjusts it when necessary.

Requirement 24 (Automated data manipulation) *Support automated compilation of data manipulation pipelines and enable their validation and manual adjustment by users.*

3.6 Provenance and license management

For the output data to be credible and reusable, detailed provenance information should be available capturing every step of the data processing task, starting from the origin of the data to the final transformation step and data export or visualization. There is The PROV Ontology[33], a W3C Recommendation that can be used to capture provenance information in RDF.

Requirement 25 (Provenance) *Provide provenance information throughout the data processing pipeline using a standardized vocabulary.*

Part of metadata of a dataset should be, at least in case of open data, the information about the license under which the data is published. A common problem when integrating data from various data sources is license management, i.e. determining whether the licenses of two data sources are compatible and what license to apply to the resulting data. This problem gets even more complicated when dealing with data published in different countries. A nice illustration of the problem is given as Table 3 in ODI's Licence Compatibility[34] page.

[33] https://www.w3.org/TR/prov-o/

[34] https://github.com/theodi/open-data-licensing/blob/master/guides/licence-compatibility.md

Requirement 26 (License management) *Support license management by tracking licenses of original data sources, checking their compatibility when data is integrated and helping with determining the resulting license of republished data.*

3.7 Data output and visualization

One of the possible goals of consuming Linked Data can be generation of visualizations of discovered data sets. This can be either automated, based on correct usage of vocabularies and a library of visualization tools as in the Linked Data Visualization Model (LDVM) [14], which offers them for DCV, SKOS hierarchies and Schema.org GeoCoordinates on Google Maps, or it can be a manually specified mapping of the RDF data as in LinkDaViz [25].

Requirement 27 (Manual visualization) *Offer visualizations of Linked Data based on manual mapping from the data to the visualizations.*

Requirement 28 (Vocabulary-based visualization) *Offer automated visualizations of Linked Data based on well-known vocabularies.*

The other possible goal of our user is to output raw data for further processing. The output data can be in various formats, the most popular may include RDF for Linked Data, which can be exported in various standardized ways. The first means of RDF data output is to a dump file using standardized serializations. The inability to create a dump in all standardized formats (only some) will be classified as partial coverage of the requirement.

Requirement 29 (RDF dump output) *Ability to export RDF data to a dump file in any standardized RDF serialization.*

Another way of outputting RDF data is directly into a triplestore, using one of two standardized ways, either the SPARQL Update query[35] or using the SPARQL Graph Store HTTP Protocol[36].

Requirement 30 (SPARQL Update output) *Ability to load RDF data to a SPARQL endpoint using SPARQL Update.*

Requirement 31 (SPARQL Graph Store HTTP Protocol output) *Ability to load RDF data to a SPARQL endpoint using the SPARQL Graph Store HTTP Protocol.*

Lastly, LDCP should be able to write RDF data to Linked Data Platform compliant servers.

Requirement 32 (Linked Data Platform output) *Ability to write data to Linked Data Platform compliant servers.*

Another popular choice for data export is into CSV for tabular data. The usual way of getting CSV files out of RDF data is by using the SPARQL SELECT queries. Such CSV files can now be supplied by additional metadata in JSON-LD according to the Model for Tabular Data and Metadata on the Web[37] W3C Recommendation. The inability to produce such metadata will be classified as partial coverage of the requirement.

[35] https://www.w3.org/TR/sparql11-update/

[36] https://www.w3.org/TR/sparql11-http-rdf-update/

[37] https://www.w3.org/TR/tabular-data-model/

Requirement 33 (Tabular data output) *Ability to export CSV data and its standardized metadata.*

Besides tabular data, tree-like data are also popular among 3-star data formats. These include XML and JSON, both of which the user can get, when a standardized RDF serialization in one of those formats, i.e. RDF/XML and JSON-LD, is enough. This is already covered by Requirement 29. However, for such data to be usable by other tools, it will probably have to be transformed to the JSON or XML format accepted by the tools and this is something that LDCP should also support. The ability to export data in only one of those formats will be classified as a partial coverage of the requirement.

Requirement 34 (Tree-like data output) *Ability to export RDF data in custom XML and custom JSON.*

For advanced graph visualizations it may be useful to export the data as graph data e.g. for Gephi [3].

Requirement 35 (Graph data output) *Ability to export RDF data as graph data.*

3.8 Developer and community support

For most of our requirements there already is a tool that satisfies them. The problem why there is no platform using these tools is caused by their incompatibility and the large effort needed to do so. LDCP does not have to be a monolithic platform and can consist of multitude of integrated tools. However, the tools need to support easy integration, which motivates the next requirements. Each part of LDCP and LDCP itself should use the same API and configuration which it exposes for others to use. The API should again be standardized, i.e. REST or SOAP based, and the configuration should be consistent with the rest of the data processing, which means in RDF with a defined vocabulary.

Requirement 36 (API) *Offer an easy to use API (REST or SOAP based) for all important operations.*

Requirement 37 (RDF configuration) *Offer RDF configuration where applicable.*

Since one of the ideas of Linked Data is distribution of effort, the data processes defined in LDCP should themselves be shareable and described by an RDF vocabulary as it is with, e.g. the Linked Data Visualization Model (LDVM) pipelines [13]. For this, a community repository for sharing of LDCP plugins like visualization components from Requirement 28, vocabulary-based transformers from Requirement 18, specialized input procedures from Requirement 8 or even whole data processing projects should be available. The ability to share only some of those will be classified as partial coverage of the requirement.

Requirement 38 (Repositories for sharing) *Offer support for plugin and project sharing repositories.*

In addition to the ability of using a project that someone else has created, an important feature is the ability of LDCP to reuse this project as a data source inside a new project. This facilitates the creation of a library of useful projects and project parts, each maintained by its author and reused by others in the same manner as other Linked Data.

Requirement 39 (Project reuse) *Offer support for reuse of shared projects as data sources.*

When the whole process of data gathering, processing and output is described in LDCP, it may be useful to expose the resulting transformation process as a web service, which can be directly consumed by e.g. web applications. This could facilitate live data views or customizable caching and update strategies.

Requirement 40 (Deployment of services) *Offer deployment of the data transformation process as a web service.*

4. REQUIREMENTS COVERAGE BY EXISTING TOOLS

In this section we survey existing approaches to Linked Data discovery, data processing and visualization and evaluate, how they cover the requirements identified in Section 3. We include approaches related to Linked Data consumption and covering multitude of our requirements, showing that they are a good starting point on a way to a full-fledged integrated solution. The single-purpose approaches which cover only one or few closely related requirements besides data input and output serve more as a motivation of our requirements, but we omit them here as they are not meant to be used as integrated platforms. An example of such a tool is the well-known linking tool SILK [28]. It loads RDF data from a dump (Requirement 5) and from a SPARQL endpoint (Requirement 6), exports the data to a dump (Requirement 29) and to a SPARQL endpoint (Requirement 30) and then covers the semantic relationships deduction Requirement 17.

Information Workbench as a Self-Service Platform for Linked Data Applications (IWB) [8] aims for support of the full life-cycle of Linked Data application development. It provides rich ways of data extraction even for non-RDF sources (Requirement 8) like relational databases, tabular data and usage of Google Refine[38]. After the extraction, the data can be previewed as a graph (Requirement 12) (Requirement 13) (Requirement 14) or in a tabular form. The workbench provides several application templates that can be easily deployed and set up. The user interface of the application can be customized to suit the needs of the user. It consists of a set of components, which are used for interaction with the underlying data. An SDK is provided, which can be used to develop new components.

The most complex representative of an integrated solution is the platform resulting from the recently finished FP7 project **Linked Data Analytics (LinDA)** [26], even though it is still far from being satisfactory as a Linked Data Consumption Platform. The platform addresses the challenge of utilization of Linked data by small and medium enterprises (SMEs). Its workflow consists of three main steps: turn data into RDF, query/link the data, analyze and visualize. LinDA integrates multiple tools to ease each of the steps. The Transformation module can be used to create RDF data from tabular data (CSV, Excel, Relational databases). The Query Builder and Query designer modules can be used to query and link data in an assisted way. The Visualisation package contains out of the box visualizations to which the user can manually map his data. The Analytics package

[38] `http://openrefine.org/`

	[8] IWB	[23] LinDA	[23] LDIF	[2] LDVizWiz	LP-ETL	[14] LP-VIZ	[12] OpenCube
1 Catalog support	(✔)	(✔)					(✔)
2 Advanced discovery			✔				
3 Context-aware discovery							
4 IRI dereferencing						✔	
5 RDF dump load	(✔)	(✔)	(✔)	(✔)	✔	(✔)	(✔)
6 SPARQL querying	✔	✔	✔	✔	✔	✔	✔
7 LDP input							
8 Non-RDF data input	✔	✔	✔		✔		✔
9 Monitoring of changes	✔						✔
10 Preview - W3C							
11 Preview - LOV							
12 Preview metadata	✔						✔
13 Preview data	✔	✔					✔
14 Preview schema	✔						✔
15 Quality indicators			✔				
16 Sem. rel. analysis							
17 Sem. rel. deduction							
18 Vocab-based transform.							
19 Auto vocab alignment	✔						✔
20 Inference							
21 Resource fusion			✔	✔			
22 Assist. sel. & proj.		✔					
23 Custom transformations		✔			✔		
24 Auto data manip.	✔						✔
25 Provenance			✔				
26 License management							
27 Manual visualization	✔	✔					✔
28 Vocab-based visualization	✔	✔		✔		✔	✔
29 RDF dump output	(✔)	(✔)	✔		✔		(✔)
30 SPARQL Update output			✔		✔		
31 SPARQL Graph Store					✔		
32 LDP output							
33 Tabular data output	(✔)	(✔)			(✔)		(✔)
34 Tree-like data output	(✔)	(✔)					(✔)
35 Graph data output							
36 API	✔				✔	✔	✔
37 RDF configuration					✔	✔	
38 Repositories for sharing	✔	✔			(✔)		✔
39 Project reuse							
40 Deployment of services					✔	✔	

Table 1: Requirement coverage by surveyed tools, red means partial coverage

contains traditional methods of statistical analysis, which are not really related to Linked Data. To support interoperability with other tools, LinDA contains the RDF2Any package to export data to conventional data formats. While LinDA offers an intuitive user interface, the supported features are still rather basic and do not exploit the benefits that Linked Data can offer satisfactorily.

Linked Data Integration Framework (LDIF) [23] aims to solve the issue when the same or related real word entities are represented as different resources using different vocabularies and sometimes even having conflicting properties. It integrates Sieve [18] for data quality (Requirement 15) and fusion (Requirement 21), R2R [4] for transformation of data among vocabularies (Requirement 18) and tracking provenance (Requirement 25), SILK [28] (Requirement 17) for link discovery and LDSpider [9] (Requirement 2) for discovery of additional data. The LDIF approach is through integration pipelines, which consist of several steps: collection of data (SPARQL, dump download, crawling), mapping to schema (transformation into a target vocabulary), resolving identities, quality assessment and data fusion and output (dump or quad store). For provenance tracking, RDF named graphs are utilized. In addition, LDIF exposes a REST based status monitor and the configuration of LDIF is in XML, which partially satisfies Requirement 37 as its offers possibility for integration with other tools.

Linked Data Visualization Wizard (LDVizWiz) [2] utilizes SPARQL queries to detect predefined categories of a dataset. The categories are detected on the base of a predefined set of vocabularies. The used SPARQL queries support the owl:sameAs links (Requirement 21) to get the whole representation of an entity identified by multiple URIs. Based on the detected categories, LDVizWiz offers predefined visualizations (Requirement 28).

LinkedPipes ETL (LP-ETL) [39] is an ETL tool mainly for Linked Data publication. However, with its library of data processing units (DPUs), it can also support the Linked Data consumption as, besides RDF input and output through SPARQL (Requirement 6) (Requirement 30) (Requirement 31) and dumps (Requirement 5) (Requirement 29) and non-RDF input and output (Requirement 8) (Requirement 33). There is a repository for sharing of DPUs on GitHub through which the functionality can be extended. The user interface is, however, not meant for users who would like to consume Linked Data and is purely ETL oriented. LP-ETL is a successor to UnifiedViews [15] and has a better developer support in a form of APIs (Requirement 36), RDF configuration (Requirement 37) and the ease of deployment of ETL processes as services (Requirement 40).

LinkedPipes Visualization (LP-VIZ) [40] is a an implementation of the Linked Data Visualization Model [14]. It aims for automated visualizations of Linked Data based on usage of well-known vocabularies. It analyzes the input datasets from dumps or SPARQL endpoints and if a supported vocabulary is detected (Requirement 28), a visualization pipeline is dynamically constructed (Requirement 24) possibly transforming from one vocabulary to another, if an appropriate transformer component is present in the instance (Requirement 18), and offered to the user. The components of the pipeline are described by an RDF vocabulary [13] as well as the pipeline itself (Requirement 37), the pipeline discovery can be triggered using an API (Requirement 36) and the resulting visualization is assigned a permanent URI that can be used for embedding in web pages and applications (Requirement 40).

OpenCube [12] is a set of integrated components which aims for the support of both publication and consumption of RDF data cubes. The main part of the OpenCube project is build on top of the Information Workbench (IWB) described above. IWB is used as an architecture backbone and it is extended with several components. The components are divided into three groups: create, expand and exploit. The *create* group focuses on data publishing and provides components to employ following technologies: TARQL[41], D2RQ (non-standard conversion from relational databases) and a transformation from JSON-stat format. The *expand* group focuses on discovery of compatible data cubes based on their dimensions and measures (Requirement 17), data cube expansion with a compatible data cube (Requirement 21)

[39] `http://etl.linkedpipes.com`
[40] `http://visualization.linkedpipes.com`
[41] `http://tarql.github.io/`

and data cube aggregation. The *exploit* group focuses on consumption and offers components to browse, visualize and analyze the stored data cubes using R[42].

There is a mention of **Assisted Linked Data Consumption Engine (ALOE)**[43] at the website of the AKSW group, however, no results of this project were found. The project was supposed to provide a platform for easier consumption of Linked Data through tackling schema mismatches.

5. RELATED WORK

In this section we go over related work in the sense of similar studies of Linked Data consumption possibilities.

The Survey on Linked Data Exploration Systems [16] provides an overview of 16 existing tools in three categories, classified according to 16 criteria. The first category represents the Linked Data browsers, the second category represents domain-specific and cross-domain recommenders, which should be considered for integration into LDCP regarding e.g. our Requirement 3. The third category represents the exploratory search systems (ESS), which could be considered for LDCP regarding Requirement 2. The surveyed systems help the user to find the data he is looking for. However, in the scope of LDCP, this is only the beginning of the consumption, which is typically followed by processing of the found data leading to a custom visualization or data in a format for further processing.

A recent survey on Ontology Matching [22] indicates there is a growing interest in the field, which could prove very useful when integrated into LDCP, covering Requirement 16 and Requirement 17.

A survey on Quality Assessment for Linked Data [29] defines 18 quality dimensions such as availability, consistency and completeness and 69 finer-grained metrics. The authors also analyze 30 core approaches to Linked Data quality and 12 tools, which creates a comprehensive base of quality indicators to be used to cover Requirement 15. One of the quality dimensions deals with licensing, which can be a good base for Requirement 26, which is, so far, left uncovered.

A survey on Linked Data Visualization [6] defines 18 criteria and surveys 15 Linked Data browsers, which could be used to cover the discovery Requirement 2 and the visualization requirements (Requirement 27 and Requirement 28).

In Exploring User and System Requirements of Linked Data Visualization through a Visual Dashboard Approach [17] the authors perform a focus group study. They aim for determining the users' needs and system requirements for visualizing Linked Data using the dashboard approach. The authors utilize their Points of View (.views.) framework for various parallel visualizations of Linked Data. They also emphasize the heterogeneity of Linked Data and the need for highly customized visualizations for different kinds of Linked Data, which supports our Requirement 10, Requirement 11 and Requirement 28.

6. CONCLUSIONS

In this paper we identified the lack of user friendly software for Linked Data consumption, which causes unfulfilled user expectations as to the usability and usefulness of Linked Data, especially compared to the comfort when working with 3-star open data such as CSV or XML files. We identified 40 technical user requirements on the Linked Data Consumption Platform and surveyed 7 existing tools which cover parts of these requirements and seem to be good candidates for establishing the way to LDCP. It is clear that in order to fulfill the user expectations, an integrated platform implementing the identified requirements is needed. Lots of those requirements are already covered by separate tools, which are, unfortunately, hard to integrate with each other, prohibiting their easy integration in such a platform. This is often caused by the fact that those tools were created as part of a now finished research project where they served their specific purpose and are now no longer maintained. It seems that the ideal way would be if such tools provided a simple REST API and were configurable via RDF, as it is e.g. with Apache Jena Fuseki[44] triplestore or R2RML[45] implementations.

7. ACKNOWLEDGMENTS

This work was supported in part by the Czech Science Foundation (GAČR), grant number 16-09713 and in part by the project SVV-2016-260331.

8. REFERENCES

[1] S. Araújo, J. Hidders, D. Schwabe, and A. P. de Vries. SERIMI - resource description similarity, RDF instance matching and interlinking. In P. Shvaiko, J. Euzenat, T. Heath, C. Quix, M. Mao, and I. F. Cruz, editors, *Proceedings of the 6th International Workshop on Ontology Matching, Bonn, Germany, October 24, 2011*, volume 814 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2011.

[2] G. A. Atemezing and R. Troncy. Towards a linked-data based visualization wizard. In *Workshop on Consuming Linked Data*, 2014.

[3] M. Bastian, S. Heymann, and M. Jacomy. Gephi: An Open Source Software for Exploring and Manipulating Networks, 2009.

[4] C. Bizer and A. Schultz. The R2R Framework: Publishing and Discovering Mappings on the Web. In *Proceedings of the First International Workshop on Consuming Linked Data, Shanghai, China, November 8, 2010*, 2010.

[5] M. Cheatham, Z. Dragisic, J. Euzenat, D. Faria, A. Ferrara, G. Flouris, I. Fundulaki, R. Granada, V. Ivanova, E. Jiménez-Ruiz, et al. Results of the ontology alignment evaluation initiative 2015. In *10th ISWC workshop on ontology matching (OM)*, pages 60–115. No commercial editor., 2015.

[6] A. Dadzie and M. Rowe. Approaches to visualising linked data: A survey. *Semantic Web*, 2(2):89–124, 2011.

[7] I. Ermilov, M. Martin, J. Lehmann, and S. Auer. Linked Open Data Statistics: Collection and Exploitation. In P. Klinov and D. Mouromtsev, editors, *Knowledge Engineering and the Semantic Web*, volume 394 of *Communications in Computer and Information Science*, pages 242–249. Springer Berlin Heidelberg, 2013.

[42] https://www.r-project.org/

[43] http://aksw.org/Projects/ALOE.html

[44] https://jena.apache.org/documentation/serving_data/

[45] https://www.w3.org/TR/r2rml/

[8] P. Haase, C. Hütter, M. Schmidt, and A. Schwarte. The Information Workbench as a Self-Service Platform for Developing Linked Data Applications. *WWW 2012 Developer Track*, pages 18–20, 2012.

[9] R. Isele, J. Umbrich, C. Bizer, and A. Harth. LDspider: An Open-source Crawling Framework for the Web of Linked Data. In *Proceedings of the ISWC 2010 Posters & Demonstrations Track: Collected Abstracts, Shanghai, China, November 9, 2010*, 2010.

[10] V. Ivanova, P. Lambrix, and J. Åberg. Requirements for and evaluation of user support for large-scale ontology alignment. In F. Gandon, M. Sabou, H. Sack, C. d'Amato, P. Cudré-Mauroux, and A. Zimmermann, editors, *The Semantic Web. Latest Advances and New Domains*, volume 9088 of *Lecture Notes in Computer Science*, pages 3–20. Springer International Publishing, 2015.

[11] T. Käfer, J. Umbrich, A. Hogan, and A. Polleres. DyLDO: Towards a Dynamic Linked Data Observatory. In C. Bizer, T. Heath, T. Berners-Lee, and M. Hausenblas, editors, *WWW2012 Workshop on Linked Data on the Web, Lyon, France, 16 April, 2012*, volume 937 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012.

[12] E. Kalampokis, A. Nikolov, P. Haase, R. Cyganiak, A. Stasiewicz, A. Karamanou, M. Zotou, D. Zeginis, E. Tambouris, and K. Tarabanis. Exploiting linked data cubes with opencube toolkit. In *International Semantic Web Conference (ISWC)*, 2014.

[13] J. Klímek and J. Helmich. Vocabulary for Linked Data Visualization Model. In *Proceedings of the Dateso 2015 Annual International Workshop on DAtabases, TExts, Specifications and Objects, Nepřívěc u Sobotky, Jičín, Czech Republic, April 14, 2015.*, pages 28–39, 2015.

[14] J. Klímek, J. Helmich, and M. Nečaský. Use Cases for Linked Data Visualization Model. In C. Bizer, S. Auer, T. Berners-Lee, and T. Heath, editors, *Proceedings of the Workshop on Linked Data on the Web, LDOW 2015, co-located with the 24th International World Wide Web Conference (WWW 2015), Florence, Italy, May 19th, 2015.*, volume 1409 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2015.

[15] T. Knap, P. Škoda, J. Klímek, and M. Nečaský. UnifiedViews: Towards ETL Tool for Simple yet Powerfull RDF Data Management. In *Proceedings of the Dateso 2015 Annual International Workshop on DAtabases, TExts, Specifications and Objects, Nepřívěc u Sobotky, Jičín, Czech Republic, April 14, 2015.*, pages 111–120, 2015.

[16] N. Marie and F. L. Gandon. Survey of Linked Data Based Exploration Systems. In D. Thakker, D. Schwabe, K. Kozaki, R. Garcia, C. Dijkshoorn, and R. Mizoguchi, editors, *Proceedings of the 3rd International Workshop on Intelligent Exploration of Semantic Data (IESD 2014) co-located with the 13th International Semantic Web Conference (ISWC 2014), Riva del Garda, Italy, October 20, 2014.*, volume 1279 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2014.

[17] S. Mazumdar, D. Petrelli, and F. Ciravegna. Exploring User and System Requirements of Linked Data Visualization Through a Visual Dashboard Approach. *Semant. web*, 5(3):203–220, July 2014.

[18] P. N. Mendes, H. Mühleisen, and C. Bizer. Sieve: Linked Data Quality Assessment and Fusion. In *2nd International Workshop on Linked Web Data Management (LWDM 2012) at the 15th International Conference on Extending Database Technology, EDBT 2012*, page to appear, March 2012.

[19] J. Michelfeit and T. Knap. Linked data fusion in odcleanstore. In B. Glimm and D. Huynh, editors, *Proceedings of the ISWC 2012 Posters & Demonstrations Track, Boston, USA, November 11-15, 2012*, volume 914 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012.

[20] K. Nguyen, R. Ichise, and B. Le. Slint: a schema-independent linked data interlinking system. *Ontology Matching*, page 1, 2012.

[21] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello. Sindice.com: a document-oriented lookup index for open linked data. *IJMSO*, 3(1):37–52, 2008.

[22] L. Otero-Cerdeira, F. J. Rodríguez-Martínez, and A. Gómez-Rodríguez. Ontology matching: A literature review. *Expert Systems with Applications*, 42(2):949 – 971, 2015.

[23] A. Schultz, A. Matteini, R. Isele, P. N. Mendes, C. Bizer, and C. Becker. LDIF - A Framework for Large-Scale Linked Data Integration. In *21st International World Wide Web Conference (WWW2012), Developers Track*, page to appear, April 2012.

[24] D. Schweiger, Z. Trajanoski, and S. Pabinger. SPARQLGraph: a web-based platform for graphically querying biological Semantic Web databases. *BMC Bioinformatics*, 15(1):1–5, 2014.

[25] K. Thellmann, M. Galkin, F. Orlandi, and S. Auer. LinkDaViz – Automatic Binding of Linked Data to Visualizations. In *The Semantic Web - ISWC 2015*, volume 9366 of *Lecture Notes in Computer Science*, pages 147–162. Springer International Publishing, 2015.

[26] K. Thellmann, F. Orlandi, and S. Auer. LinDA - Visualising and Exploring Linked Data. In *Proceedings of the Posters and Demos Track of 10th International Conference on Semantic Systems - SEMANTiCS2014*, Leipzig, Germany, 9 2014.

[27] P. Vandenbussche and B. Vatant. Linked Open Vocabularies. *ERCIM News*, 2014(96), 2014.

[28] J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov. Discovering and maintaining links on the web of data. In A. Bernstein, D. Karger, T. Heath, L. Feigenbaum, D. Maynard, E. Motta, and K. Thirunarayan, editors, *The Semantic Web - ISWC 2009*, volume 5823 of *Lecture Notes in Computer Science*, pages 650–665. Springer Berlin Heidelberg, 2009.

[29] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, and S. Auer. Quality assessment for Linked Data: A Survey. *Semantic Web*, 7(1):63–93, 2016.

[30] X. Zhang, D. Song, S. Priya, Z. Daniels, K. Reynolds, and J. Heflin. Exploring linked data with contextual tag clouds. *Web Semantics: Science, Services and Agents on the World Wide Web*, 24:33 – 39, 2014. The Semantic Web Challenge 2012.