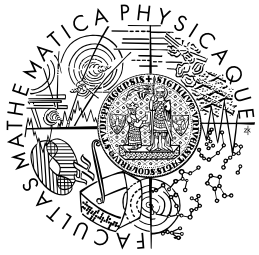

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

HABILITATION THESIS

Jakub Klímek

Improving data accessibility and interoperability using linked data

Prague 2022

I would like to thank my colleagues, especially **Martin** and **Petr**, for being such a great team, and my family, especially **Eliška** and **Čeněk**, for being there for me.

Title: Improving data accessibility and interoperability using linked data

Author: Jakub Klímek

Abstract: In this habilitation theses we describe our contributions to the field of data exchange using Linked Data with the aim of improving data accessibility and interoperability. We show the contributions from the points of view of a data provider and a data consumer. The data provider point of view focuses on software support for Linked Data publication, where we describe the LinkedPipes ETL tool for publication and consumption of Linked Data, including its optimizations and use cases. We also include a demonstration of usefulness of the Linked Data approach on two specific real world use cases from the Czech public administration, where we utilize the previously developed tool. The data consumer point of view shows how Linked Data published on the web can be consumed and how the data consumers can be helped by tools utilizing the basic Linked Data principles, including the possibilities of automation of the consumption process. This point of view includes a comprehensive survey of existing approaches.

Keywords: linked data, consumption, provision, data interoperability, life cycle support

Contents

Introduction

The value and importance of data rises in the recent decades. More and more companies make their business based on data, while governments and companies alike need data to make better decisions. Therefore, quite rightfully, data is being called the 21st century oil.[1]

However, the quantity of available data also gives rise to issues associated with its usage. This is especially true for usage of data coming from different data sources. We are talking about data quality, which is often defined simply as *fitness for use*.

Data quality is a very broad term which comprises many quality dimensions, where each dimension has a multitude of metrics used to measure it. Examples of data quality dimensions include accessibility, accuracy, validity, completeness, conciseness, consistency, interoperability, relevance, trustworthiness, versatility, etc. In this thesis, we will mainly focus on *accessibility* and *interoperability*.

Today's data scientists spend 80% of their time dealing with interoperability issues such as searching for data, getting access to the data, cleaning the data or reorganizing it, and they only spend 20% of their time analyzing or visualizing the data.[2] This clearly indicates that focusing on these issues is critical in today's data driven society.

The first big milestone addressing data, or more precisely at that time, document accessibility was the creation of ARPANET, the predecessor of the Internet, in 1969. Thanks to the standardization efforts regarding the low-level computer interface interoperability, documents could be sent over the computer network instead of being transported on physical data media such as magnetic tapes, making them more accessible. However, based on this newly gained improvement in computer interface interoperability, a multitude of protocols, formats and architectures emerged, causing new interoperability and accessibility issues on a new level.

The next milestone in addressing document interoperability and increasing their accessibility was the invention of the World Wide Web (WWW) in 1989 by Sir Tim Berners-Lee. The WWW is a layer of standards on top of the Internet which utilizes URLs [4] for the identification and localization of documents, now called web pages, the HTTP protocol [3] for transportation of the web pages across the Internet, HTML[3] as a single format for web pages and, most importantly, a standardized way of interlinking the web pages via the HTML's anchors (`<a>` elements) in a way where the web page author did not have to negotiate the link with the author of the target web page. The WWW, propelled by the network effect, changed the world of information exchange just by introducing a few simple standards and proof-of-concept technologies using them. However, just like it was with the advancement in computer interface interoperability, based on this newly gained improvement in document accessibility and interoperability,

[1] `https://economictimes.indiatimes.com/magazines/panache/data-is-the-21st-centurys-oil-says-siemens-ceo-joe-kaeser/articleshow/64298125.cms`

[2] `https://www.infoworld.com/article/3228245/the-80-20-data-science-dilemma.html`

[3] `http://info.cern.ch/hypertext/WWW/MarkUp/Tags.html`

a multitude of new ways of exploiting it arisen. Among them, data started to be exchanged in addition to the human readable web pages. And again, a multitude of data formats, such as XML, CSV, XLS, etc. started to be exchanged on the Web, causing interoperability issues on a new level. As it was before the WWW for documents, uniform identification of data entities was missing, as was a common data format and a standardized way of interlinking related data entities. Therefore it became hard to answer even a simple question like *Is a company referred to in one file the same as the company referred to in another file?*

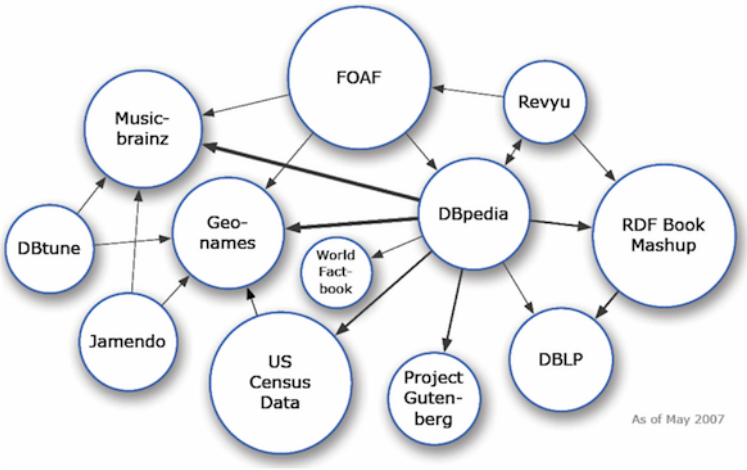

Figure 1: The LOD cloud in 2007 - 12 datasets

In 2006, Sir Tim Berners-Lee introduced the 4 principles of Linked Data[4] with the aim of helping with the data interoperability issues. These principles are based on the same principles which made the WWW, the web of documents, so successful. According to the principles, data producers should identify their data entities, such as companies, or people, using HTTP URLs, the same kind of identifiers they are used to use for their web pages. In addition, the Resource Description Framework (RDF) [5] should be used for data representation, as it naturally supports both the URL-based entity identification and standardized interlinking of related data entities. Finally, the data entities should link to related data entities using their URLs so that the data consumers can follow the links and be able to get more related data.

While the principles undoubtedly do improve data interoperability, especially in cases of working with multiple data sources, the adoption of this way of exchanging data on the web is slow. This can be illustrated by comparing the diagrams depicting the state of the so called linked open data cloud in 2007 (Figure 1) and in 2021 (Figure 2). However, the linked data principles also have other use cases besides publishing open data on the Web. They are also applied in the industry, where they power the so called knowledge graphs, and recently, they started to play a big role in the area of personal data protection on the Web, giving rise to the Solid ecosystem[5] of decentralized personal online datastores, or *pods*.

[4] `https://www.w3.org/DesignIssues/LinkedData`
[5] `https://solidproject.org`

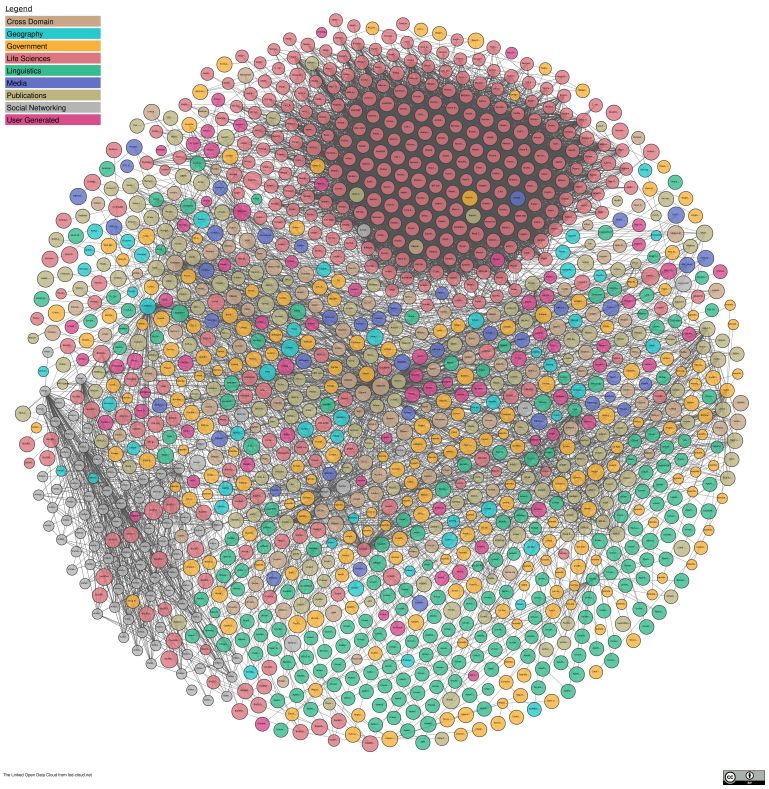

Figure 2: The LOD cloud in 2021 - 1301 datasets

Our contributions presented in this theses are:

- Identification and description of key steps in Linked Data provisioning and consumption processes
- Design and implementation of open-source tools supporting some of the steps of the processes
- Application of the implemented open-source tools in real-world scenarios, e.g. the Czech National Open Data Catalog run by the Ministry of the Interior of the Czech Republic, or the Linked Data production system implemented by the Czech Social Security Administration

The rest of this thesis is structured as follows. First, we introduce the used terminology. Then we introduce the data exchange processes using Linked Data (LD), their individual steps and context. We split the description into the LD provisioning process employed by data providers and the LD consumption process employed by LD consumers, and we state our contributions supporting these processes. The connecting point of those two processes is the data catalog hierarchy used by data providers to register provided data and by data consumers to discover data to be used. Note that in the context of open data[6], data provisioning is also called data publication, as the openness of data relates to the ability to reuse it freely. Most of our contributions were applied in the context of open data. However, since our approach is not exclusive to open data, we use the term data provisioning in this thesis, as that applies also to closed, e.g. enterprise, environments.

[6] `https://opendatahandbook.org/guide/en/what-is-open-data/`

Terminology

In this section we introduce the terminology used in this thesis.

Resource Description Framework

The Resource Description Framework (RDF) [5] is a graph data model. Technically, it is an oriented labelled multigraph. Entities, such as books or people, are represented as nodes in the graph and usually identified using an IRI such as `https://jakub.klímek.com/#me`. This IRI then identifies a person, Jakub Klímek. Relations of entities are represented as orientated edges in the graph, connecting two nodes. The edges are also labelled by IRIs identifying the meaning of the edge, such as `https://schema.org/worksFor`. RDF can then be viewed as a set of triples s, p and o, or *subject*, *predicate* and *object*, respectively, where s and o are entities and p is their relation. For example, the *RDF triple*

```
https://jakub.klímek.com/#me
https://schema.org/worksFor
https://www.cuni.cz
```

means that Jakub Klímek works for the Charles University. There is also another type of object that can be used in RDF triples called a literal, containing a simple data value along with its datatype or language specification. For example, the triple

```
https://jakub.klímek.com/#me
http://xmlns.com/foaf/0.1/givenName
"Jakub"@cs
```

means that Jakub Klímek has a given name, and that name is Jakub in Czech. Like this, any data can be represented using RDF. Also, the entities and their relations are identified globally and uniquely, thanks to the usage of IRIs. This, along with proper usage of vocabularies introduced later in the terminology section, helps greatly with interoperability of data from multiple data sources, especially when compared to traditional data formats such as CSV, JSON or XML.

Since RDF is a data model, it needs specifications of how to represent it in text, e.g. for saving it to disk or for network transfer. There are standard serializations of RDF defined, like RDF Turtle [17] or JSON-LD [16].

All the approaches presented in this thesis are based on working with RDF as a data model. Note that we omit some details of RDF like blank nodes and named graphs here, as they are not essential for the thesis.

Linked Data

The term Linked Data was introduced by Sir Tim Berners-Lee in 2006[7]. It is a set of four rules similar to those powering the Web of Documents, ensuring accessibility and interoperability of data on the Web. The rules are

1. Use URIs as names for things

[7] `https://www.w3.org/DesignIssues/LinkedData.html`

2. Use HTTP URIs so that people can look up those names

3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)

4. Include links to other URIs. so that they can discover more things

Note that in these original rules, URIs[8] are used, whereas nowadays, we use IRIs[9] in RDF.

These rules encourage unique global identification of entities and relations in the data, usage of RDF as a standard for exchanging data on the Web, and linking data in one datasource to data in other known and relevant data sources, so that applications, as well as people, can navigate the data on the Web using links the same way we know from the traditional Web of Documents.

This is crucial when working with data from multiple data sources. If those data are published using the Linked Data rules, i.e. already interlinked, all the data consumers are saved the effort of integrating those data sources themselves.

The approaches in this theses support data providers and data consumers in working with Linked Data, helping them achieve better data accessibility and interoperability.

Vocabularies

When we introduced RDF as a data model, we used an example where we said that, e.g. `https://schema.org/worksFor` identifies a predicate which means that someone works for some organization. The question is how do we know that? The answer lies in what is called vocabularies. Vocabularies are human and machine readable definitions of types of entities, called *classes*, and relationships, called *predicates*, that can connect entities, typically in some domain, like publications, online profiles, e-commerce, etc. Each class and predicate definition specifies a globally unique IRI to be used to identify the class or predicate, its human readable names, descriptions, and relations to other classes and predicates.

For example, in the *schema.org* vocabulary, `https://schema.org/worksFor` is defined as a predicate with a label *worksFor* and a description *Organizations that the person works for.* In addition, it is specified, that this predicate is used on the type `https://schema.org/Person`, and that the values, or objects, are expected to be of the type `https://schema.org/Organization`.

There is a vocabulary catalog called Linked Open Vocabularies[10] (see [20]) providing an overview of vocabularies used on the Web of Data. Data publishers are then encouraged to first look there for already defined classes and predicates corresponding to the meaning of their data to be published as Linked Data, and only when they do not find a suitable candidate, define their own, again increasing interoperability of their data.

In this thesis we present approaches to Linked Data visualization that exploits the fact that well known vocabularies are often used in Linked Data sources and therefore can be used by software to provide at least some data visualization automatically.

[8] `https://www.ietf.org/rfc/rfc3986.txt`
[9] `https://www.ietf.org/rfc/rfc3987.txt`
[10] `https://lov.linkeddata.es/dataset/lov/`

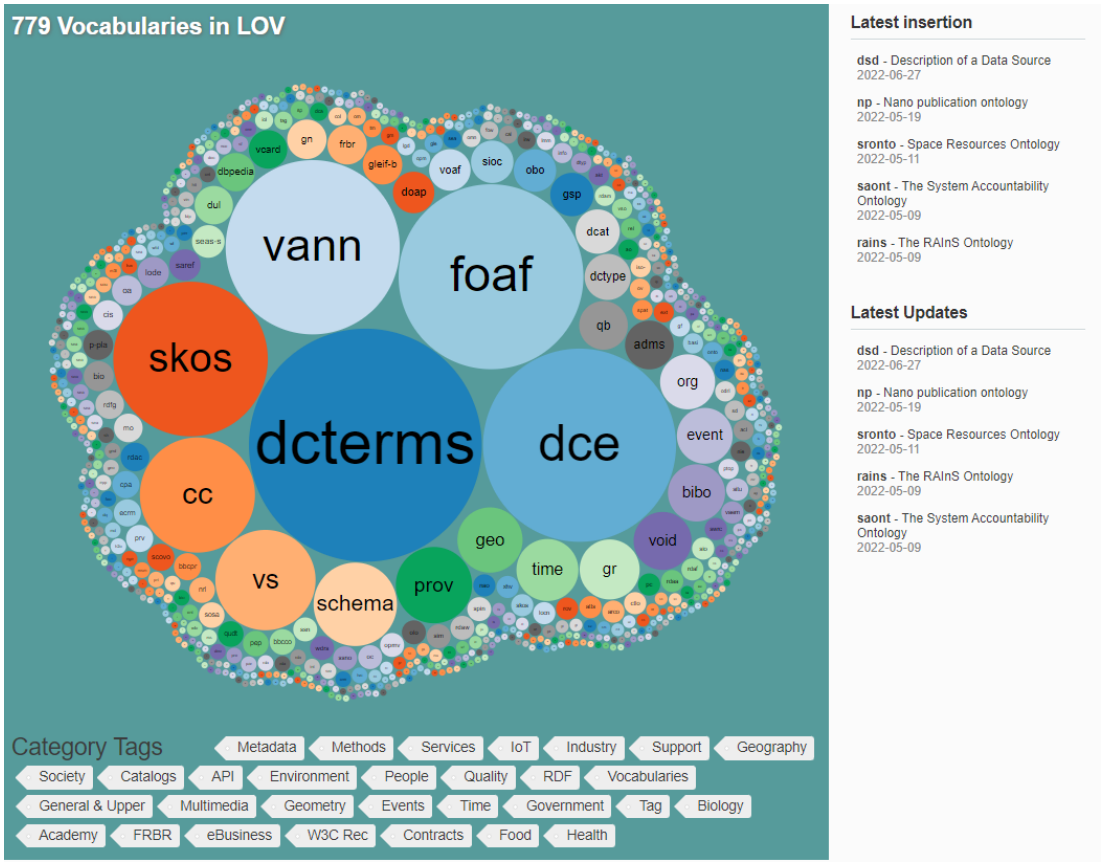

Figure 3: Linked Open Vocabularies

Open Data

While Linked Data is a set of rules about how to represent data on the Web technically, the term *Open data* deals not only with the technical maturity of data, but also with with the legal side of publishing data on the Web. For example, when a potential data consumer finds a data file freely downloadable on the Web, there is no legal assurance that they can, e.g. use, process, or further share the data. The mere fact that the data was freely accessible through the Web gives no such legal assurance. If the data is indeed intended to be used and re-used freely, it is necessary to accompany it with a legal document saying so, giving the consumers the legal assurance.

There is a quite well-known 5-star deployment scheme for Open data, see Figure 4. A data source is awarded one star, if their data, in whatever technical quality, is accompanied by an open license.

A second star is awarded to data available in a structured data format. For instance, if the original data to be published was available as an Excel file, it will also be published as structured data. It will not be published as, e.g. an image of an Excel table, making it very hard to process.

The third star is awarded for data published in open formats. An *open format* is a data format with an openly available specification so that anyone can freely create a software for processing data in that format. This prohibits usage of data formats which require, e.g. certification, or reverse-engineering, in order for the data in that format to be processable.

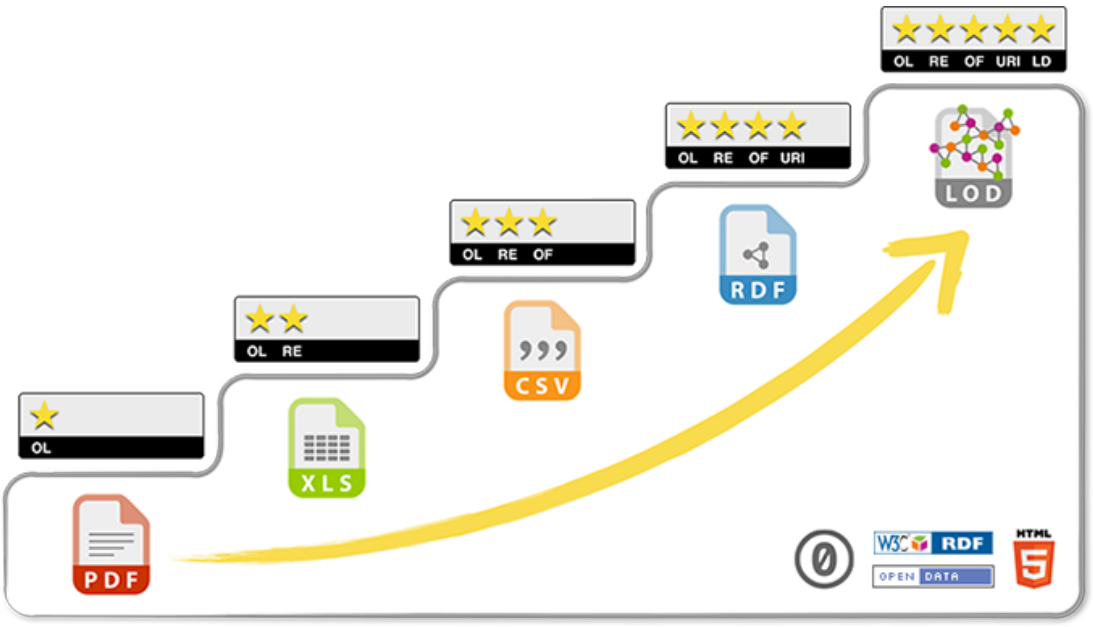

Figure 4: 5* Open Data

The fourth star is awarded for identification of data entities using IRIs, i.e. in a globally unique way. Currently, this means also using the RDF data model to represent the data, as at the moment there is no other data format natively supporting the identification of entities using IRIs.

The fifth and final star of the deployment scheme is for publishing the data as Linked Data, making it Linked Open Data (LOD), including links to other data sources, so that the consumers can navigate to related data.

It is worth noting here that the Linked Data rules can be applied both to open data, and to data in internal, e.g. enterprise, networks in the same way as there are Intranet pages. While most of the contributions mentioned in this thesis are demonstrated on Linked Open Data, they are directly applicable also to closed, i.e. not open, Linked Data.

Linked Data Provisioning Process

The Linked Data (LD) provisioning process (see Figure 5) starts with an internal representation of data to be provided or published as LD. If the data is not documented yet, which, unfortunately, often is the case, it needs to be analyzed first. The analysis results in data documentation containing a conceptual domain model showing what the data is about, and a technical part showing in which data formats, structures and in which data storage systems the data currently resides. This documentation along with the data itself serve as part of the inputs to the transformation process.

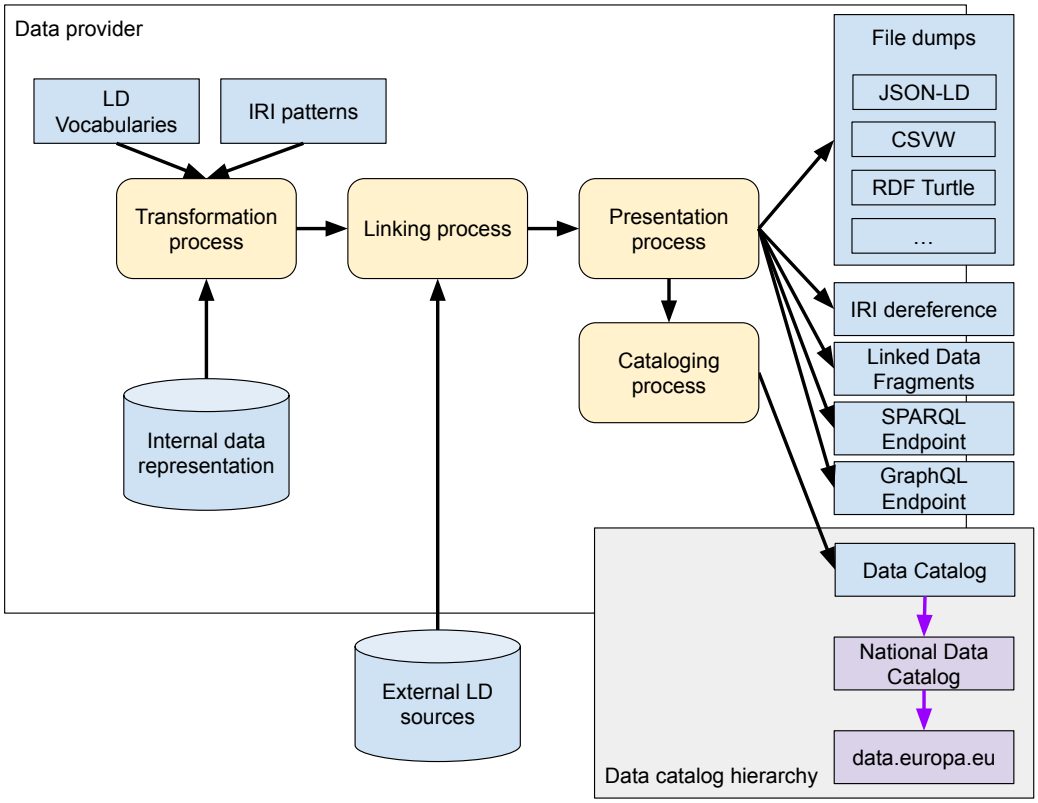

Figure 5: Linked Data Provisioning Process

Transformation process In this step, the data is transformed from the internal representation to an LD representation. In the case where the internal representation already is an LD representation, changes to that representation may still occurs as part of the provisioning process. In order to be able to define the transformation process, we need two additional inputs in addition to the internal data and its documentation. First, we need to have the classes and predicates to be used for the provided data entities and their properties either created in form of our own ontology or vocabulary, or chosen from an existing vocabulary, found, e.g., in Linked Open Vocabularies [20]. Second, we need IRI patterns[11] established for the actual data instances, i.e., decisions on how the individual parts of IRIs will look like, so that they are assigned consistently. Such IRI patterns can be very

[11] `https://patterns.dataincubator.org/book/identifier-patterns.html`

simple, e.g., https://mycompany.org/resource/<ClassName>/<instance-id>, but they are essential so that the data transformation designer knows how the output LD should look like. When all the inputs are defined, a data transformation process can also be defined and implemented.

Linking process One of the essential features of Linked Data is directly in its name. It is the possibility to provide context - links to related data. These links can be created either by the data provider, the data consumer or a third-party. However, the data provider is the one who understands their data the best, including related data sources. Therefore, the linking process is an essential part of the entire LD provisioning process, and involves defining links from the providers' data entities do related data entities, possibly provided by someone else. The data consumer can then decide whether or not they want to follow these links to get more related data.

Presentation process Once the data is transformed to its LD representation and augmented with links to related data, the last step is the presentation of the data using one or more LD interfaces.

The first basic LD interface type is a data dump, which allows users to download the entire dataset as one file. The file may be presented in various formats. Examples of such formats are RDF Turtle [17], a native RDF serialization, JSON-LD [16], another standard RDF serialization, which also allows JSON developers to work with the data without the knowledge of RDF, or CSV on the Web [19], a serialization of data in CSV files with an additional descriptor, allowing also developers used to CSV to work with the data without the knowledge of RDF.

The second basic LD interface type is a query service. The most prominent representative of this LD interface type is a SPARQL [18] endpoint, an RDF database interface allowing developers to ask queries and get query results through a web service. Another example of such a query service is a Linked Data Fragments [21] endpoint, or, a non-LD service GraphQL[12] endpoint.

The third basic LD interface type is IRI dereference, which is also the third Linked Data principle. It involves responding to requests to the IRIs identifying individual data entities with their representation in RDF.

Cataloging process When LD is provided via some of the LD interfaces, the final step in the provisioning process is making the data discoverable using a data catalog. Typically, the data provider runs their own data catalog with metadata describing the provided data. Such a data catalog should have a machine readable interface of its own, compliant with the Data Catalog Vocabulary (DCAT) W3C Recommendation [2], and, in Europe, its Application Profile[13]. In the case of the data being open data, the publisher's data catalog becomes part of a data catalog hierarchy, as it will get harvested typically by a national open data catalog, which, in turn, gets harvested by the Official portal for European data - `https://data.europa.eu`, making the metadata records discoverable internationally.

[12] `https://graphql.org/`

[13] `https://joinup.ec.europa.eu/collection/semantic-interoperability-community-semic/solution/dcat-application-profile-data-portals-europe`

Contributions Supporting the LD Provisioning Process

In this section we present our contributions supporting the steps of the LD provisioning process, and, by doing so, improving data accessibility and interoperability.

In LinkedPipes ETL: Evolved Linked Data Preparation [14], we present a tool called LinkedPipes ETL (LP-ETL) supporting the entire LD provision process from accessing the internal data, through its transformation, linking, presentation using various LD interfaces and with support for catalogization using DCAT. It contains a library of components reusable in a typical LD production and consumption processes and composable into data transformation pipelines. Based on our real-world experience, we optimize performance of some typical parts of the processes in Speeding up publication of Linked Data using data chunking in LinkedPipes ETL [12]. In LinkedPipes ETL in Use: Practical Publication and Consumption of Linked Data [11] we show how LP-ETL is applied in real world use cases.

LP-ETL is an open-source tool developed at GitHub `https://github.com/linkedpipes/etl` under the MIT license, with its documentation available at `https://etl.linkedpipes.com`. It was used in the H2020 project OpenBudgets.eu [14] to transform budget and spending data of various publishers to LD representation. LP-ETL is currently being used in the STIRData[15] CEF Telecom project to transform company data from the Czech Business Registry to LD representation.

In DCAT-AP Representation of Czech National Open Data Catalog and its Impact [7] we describe how we helped the Ministry of the Interior of the Czech Republic[16] to publish metadata from the Czech National Open Data catalog[17] as LD, again by applying LP-ETL. We show how this made them trendsetters in the publishing of metadata about national open data in the context of the European Union. The Ministry of the Interior of the Czech Republic is responsible for the agenda of eGovernment, which includes the coordination of publishing open data from various data sources in the public administration. This includes, among other activities, running the Czech National Open Data Catalog.

In addition, LP-ETL is in the process of being deployed in the same way as a new version of the Slovak National Open Data Catalog run by the Ministry of Investments, Regional Development and Informatization of the Slovak Republic[18].

Moreover, in Publication and Usage of Official Czech Pension Statistics Linked Open Data [9] we describe how we helped the Czech Social Security Administration (CSSA) to publish official LD representation of their pension statistics data, applying LP-ETL in the process. The CSSA is the part of the Czech public administration responsible for the organization and implementation of social security. It has more than 8.9 million clients, pays more than 3.5 million pensions and around 280 thousand sickness insurance benefits monthly, according to data from 2022[19].

[14] `https://openbudgets.eu/`
[15] `https://stirdata.eu`
[16] `https://www.mvcr.cz/mvcren/`
[17] `https://data.gov.cz/english`
[18] `https://www.mirri.gov.sk/`
[19] `https://www.cssz.cz/web/lang/cssz`

In addition to production deployments, LP-ETL is also used as a teaching material in LD related courses at the Charles University, the Czech Technical University in Prague and the Prague University of Economics and Business.

Finally, even though we do not collect LP-ETL usage statistics, there is plenty of evidence in the GitHub repository that LP-ETL is used world-wide. This claim is supported by issues we receive from the community, and the occasional pull request with contributions to the code base.

Regarding the cataloging process part of LD provisioning, in LinkedPipes DCAT-AP Viewer: A Native DCAT-AP Data Catalog [13] we introduce the DCAT-AP Viewer, showing how LD representation of a data catalog using DCAT and DCAT-AP can be presented to end users. This tool is currently deployed in production at the Ministry of the Interior of the Czech Republic as the official user interface of the Czech National Open Data Catalog at `https://data.gov.cz/datasets`. It is also deployed as a user interface of the institutional catalog of the Ministry of the Interior of the Czech Republic at `https://data.mvcr.gov.cz`. In addition, it is currently used as a user interface of a data catalog listing datasets used within the STIRData CEF Telecom project at `https://stirdata.opendata.cz`.

Linked Data Consumption Process

The Linked Data (LD) consumption process (see Figure 6) is typically defined by the data consumer intending to use data for a certain task. The task might be, e.g., writing a data-based article in the case of a data journalist, synthesizing data from multiple sources in case of a data analyst, or simply keeping an up-to-date copy of a certain dataset. In some cases, the data consumer already knows where to find the data they need, but in many cases, the LD consumption process starts with the data consumer searching for data to be used.

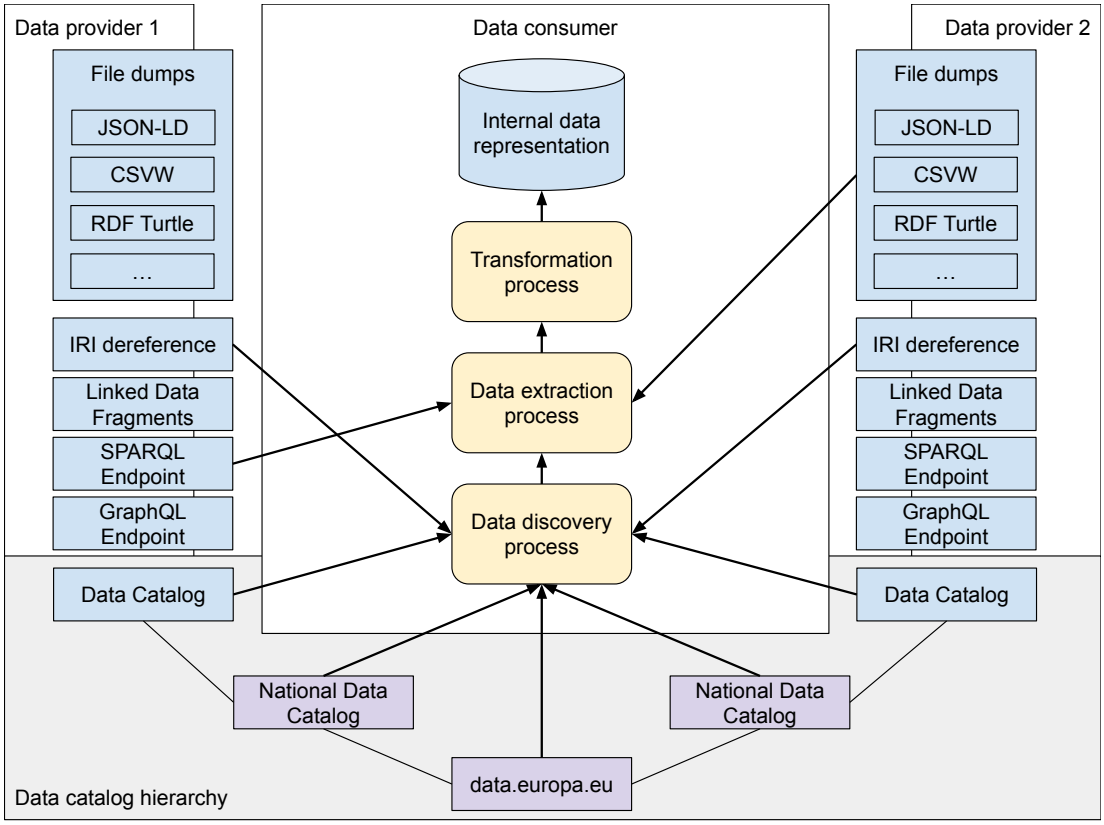

Figure 6: Linked Data Consumption Process

Data discovery process In this step, the data consumer typically accesses a data catalog in a data catalog hierarchy such as the Official portal for European data[20], the Czech National Open Data Catalog[21] or equivalent, e.g., in a closed, institutional environment, and uses its user interface to find data they need to use. Alternatively, the data catalogs provide an API - a machine-readable interface that can be used by the consumer or their software to help them search the metadata contained within the catalogs. Up to now, the process is the same for regular data and for LD. However, nowadays, data catalogs such as the two mentioned above already provide a multitude of LD interfaces such as data dumps, IRI dereference and SPARQL endpoints so that the metadata records contained within them are accessible as LD, and thus the data consumer is not limited by

[20] `https://data.europa.eu`
[21] `https://data.gov.cz`

what the user interface allows them to query. How such an LD interface of a data catalog can be built is described in DCAT-AP Representation of Czech National Open Data Catalog and its Impact [7].

Specifically in LD, the data can be discovered not only via a data catalog, but also directly, through following the IRI links in already known LD, a.k.a. IRI dereference. In fact, there are well-known techniques of discovering where an entire data dump or a queryable SPARQL endpoint is located based on just the IRI of a single data entity from the dataset [1].

Data extraction process Once the data of interest is located, possibly at multiple providers, it needs to be extracted, i.e., transferred from its origin to the data consumer's platform. With LD, this can happen in multiple ways. First, the data can be either queried directly, in case the data provider provides a query service such as a SPARQL endpoint or a Linked Data Fragments endpoint. Next, the data can be downloaded in bulk as a data dump. In the case of LD, the data dump will be typically in one of the standard RDF serializations such as RDF Turtle, JSON-LD or CSV on the Web. Lastly, the data can be accessed record by record through IRI dereference, which is desirable for cases where downloading the entire data dump would not be proportionate to the task being solved.

Transformation process Once the relevant data is extracted, the consumer can perform the desired task directly, e.g. by querying the gathered data using SPARQL. However, it is often the case that while LD is a suitable format for data exchange, it is not that suitable for certain, mainly larger-scale data operations such as aggregations over large datasets, geospatial querying, running graph algorithms, etc. Therefore, another way of processing the gathered data is transforming it to a different data format or representation suitable for further processing by the consumer's tool of choice or storing in the consumer's database, which might not be an RDF database. Regardless of the LD interface type used for extraction, the consumer needs to understand the data. Specifically, they need to understand the LD vocabularies used in the data to be able to query or transform it. However, as we show later in our contributions, it is possible to leverage the fact that LD providers are encouraged to re-use already existing and known LD vocabularies, and to automatically offer ways of processing or visualizing the data.

Note that some data consumers can at the same time also be data publishers. In that case, the LD provisioning process could start where the LD consumption process ended, i.e. with the internal data representation, which might or might not be LD representation.

Contributions Supporting the LD Consumption Process

In this section we present our contributions supporting the steps of the LD consumption process. By doing so, we also improve data accessibility and interoperability, because the data providers are incentivized to publish their data as LD when they can see the effect of doing so immediately, without waiting for some data consumer to acknowledge that working with the data in the LD representation was more convenient for them.

In LinkedPipes Visualization: Simple Useful Linked Data Visualization Use Cases [8] we show how visualization of properly published Linked Data can be automated using the LinkedPipes Visualization tool. The tool works by automatically discovering which LD vocabularies are used in the data provided, and for some of the most popular vocabularies offers automatic visualizations of the data according to the vocabulary. LinkedPipes Visualization is an open-source tool hosted on GitHub at `https://github.com/ldvm/LDVMi` under the Apache 2 license, documented at `https://visualization.linkedpipes.com/`.

A similar approach is presented in LinkedPipes Applications - Automated Discovery of Configurable Linked Data Applications [10], a recent evolution of the automated LD visualization approach, focusing on visualizations which are configurable by the LD providers, but remain interactive for LD consumers. In this way, the consumers do not see only what the data providers prepare for them and they can adjust the visualizations on their own. On the other hand, the data providers can suggest tailored ways of visualizing their data, as the purely automated visualizations are not always usable. LinkedPipes Applications is an open-source tool hosted on GitHub at `https://github.com/linkedpipes/applications` under the Apache 2 license, with an online demo available at `https://applications.linkedpipes.com/`. Another interesting aspect of the tool is that the visualization configurations are stored in the users' Solid PODs - LD platforms with read-write access and the user in control of their data.

To ease the consumption of LD to consumers with limited knowledge of RDF and SPARQL, but familiar with CSV, in Simplod: Simple SPARQL Query Builder for Rapid Export of Linked Open Data in the Form of CSV Files [6] we present a SPARQL query builder based on identification of a relevant subset of the data provided through a SPARQL endpoint, in a graphical representation. The identified subsets are again stored in Solid PODs, allowing providers and consumers to collaborate on them. From the identified subset of the data, a SPARQL query returning a corresponding CSV file is generated automatically. The primary use case is that the providers can pre-select the most relevant subset of their data, store the project in their Solid POD, and the data consumers can then further refine that subset to what is relevant to them, and have a SPARQL query generated automatically. Simplod is an open-source tool hosted on GitHub at `https://github.com/mff-uk/simplod` under the MIT license. A demo instance is running at `https://jaresan.github.io/simplod/build/index.html`.

Moreover, the already mentioned LinkedPipes DCAT-AP Viewer presented in LinkedPipes DCAT-AP Viewer: A Native DCAT-AP Data Catalog [13] can also be viewed as a LD consumption tool since it takes LD representation of data catalog records using DCAT and DCAT-AP on the input and visualizes it in the form of a user interface.

Finally, in Survey of Tools for Linked Data Consumption [15] we present a survey of existing tools for LD consumption and visualization using a set of criteria based on the LD end-users expectations, given that LD is perceived and presented as the superior way of providing data on the web, at least by the academic community. From the survey, requirements on a Linked Data Consumption Platform - a tool that would ease working with LD to the consumers, properly leveraging LD advantages over other data formats, are formulated, and serve as a base of our ongoing research.

Architecture and Tools Supporting Data Exchange Using Linked Data

In this section, we introduce a typical architecture of data exchange using Linked Data and we show, how our contributions support the individual actors and processes. The architecture diagrams uses the ArchiMate[22] language. In Figure 7 we can see the subset of ArchiMate components used to depict the architecture in this section.

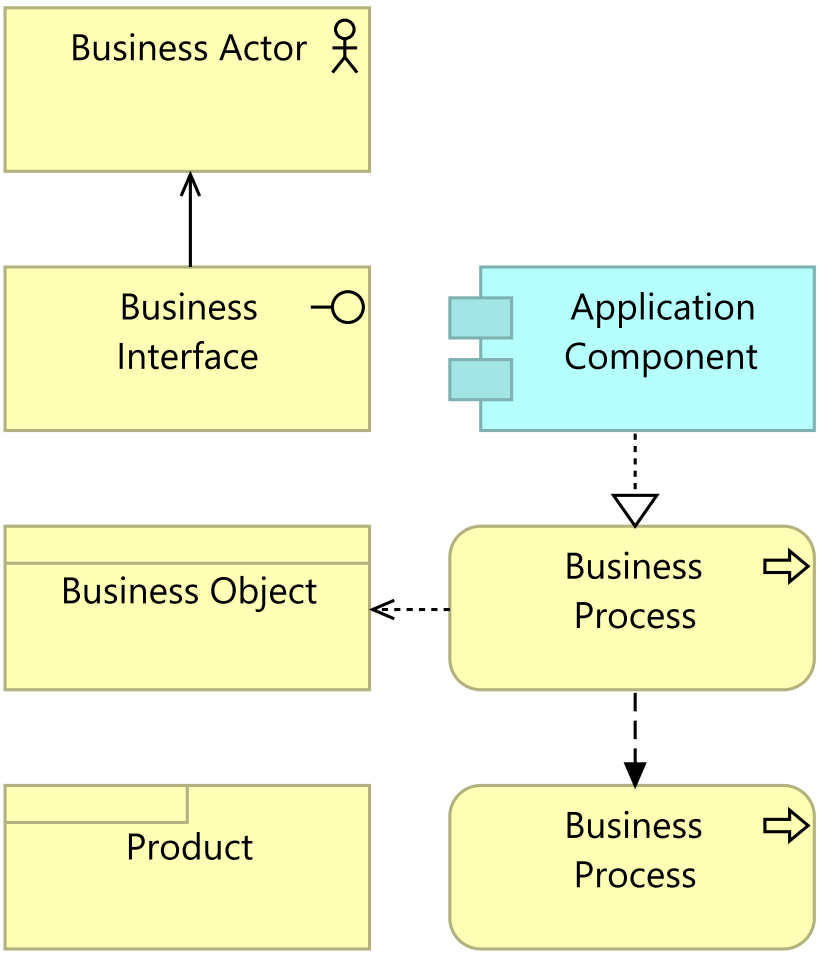

Figure 7: Overview of the used ArchiMate components

The yellow components are part of the ArchiMate business layer. We depict the data providers platform, the data consumers toolkit, the National Open Data Catalog and the Official portal for European data as *Products*. *Business Objects* are passive things, we use them to represent data. *Business processes* represent activities such as data transformations. They are *realized* by *Application components*, part of the Application layer, representing the software realizing the business process. They also *access* business objects, and can have the *flow* relation among themselves, representing their sequence. *Actors* represent our data providers, processors and consumers. Lastly, *Business intrefaces* represent the

[22] `https://www.opengroup.org/archimate-forum/archimate-overview`

interface where a service such as user interface or data interface is provided to be consumed by someone else, i.e. the Interface *serves* an Actor, a Business proces or an Application component.

Data provider

In Figure 8 we can see the the view of a potential LD provider's platform. It contains the business processes involved in provision of LD as described in the previous section. In addition, we can see how the business processes can be realized by the software described in the contributions sections of this thesis.

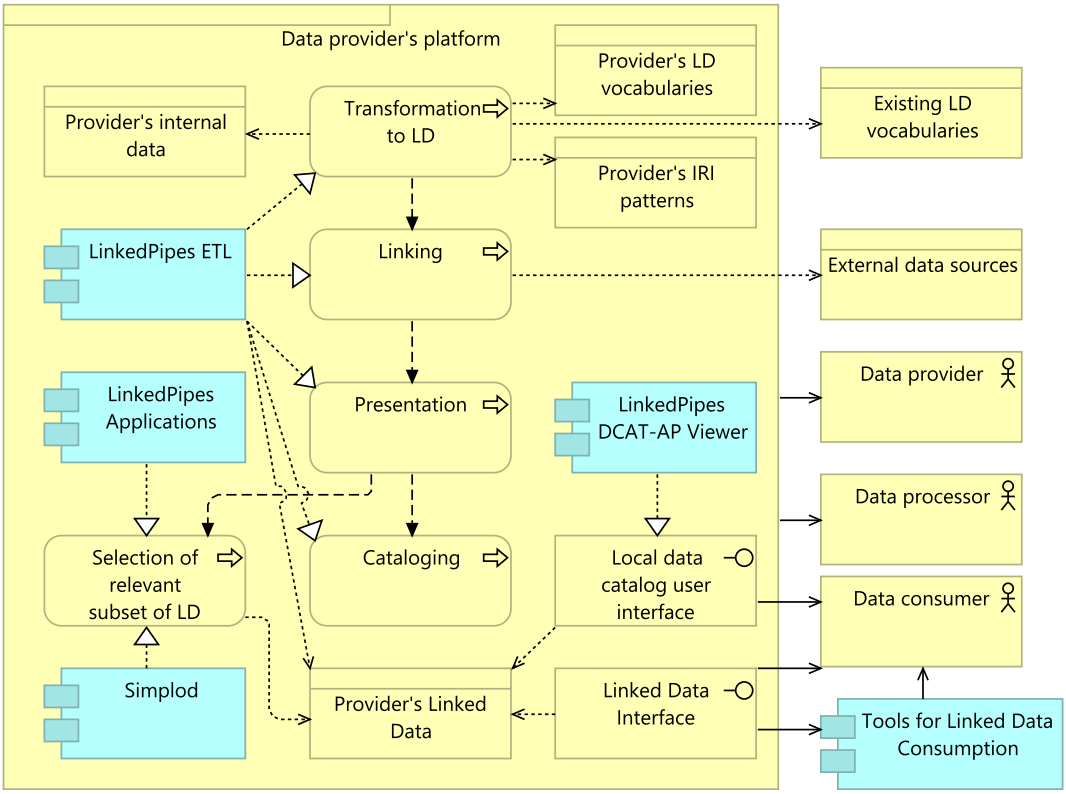

Figure 8: Architecture of data exchange using Linked Data and the approach presented in this thesis.

LinkedPipes ETL realizes the transformation, linking, presentation and cataloging processes. LinkedPipes DCAT-AP Viewer realizes the user interface of the provider's data catalog. A data consumer can use any tools for Linked Data consumption with the provided LD interfaces. Finally, the data provider can ease the process of consumption of their LD by providing selections of relevant subsets of their data using Simplod for easy generation of CSV files, or using LinkedPipes Applications to provide customizable visualizations to potential consumers. Note that a data processor is both a data provider and a data consumer.

Data consumer

In Figure 9 we can see the view of a potential LD consumer's toolbox.

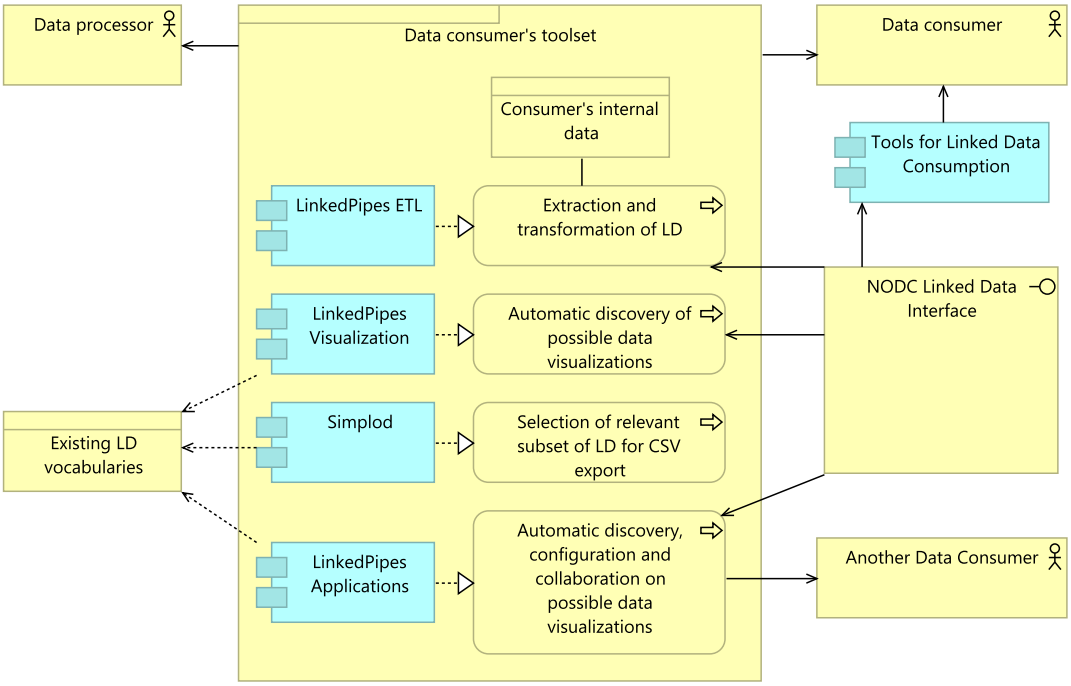

Figure 9: Architecture of a LD provider platform with tools supporting it

The consumer can use LinkedPipes Visualization to try to automatically visualize data served using a SPARQL endpoint. Using LinkedPipes Applications, they can either do the same, or start with a visualization provided by the data provider and refine it. Using Simplod, the consumer can easily get a CSV representation of the provided LD, ideally based on a project pre-created by the data provider. If the data consumer is an LD expert, they can also use LinkedPipes ETL to combine and transform the provided LD, producing an internal representation of the data in a different data format, possibly for further processing by non-LD tools. Finally, they might want to use any of the surveyed tools for LD consumption. Note that a data processor is both a data provider and a data consumer.

National Open Data Catalog

In Figure 10 we present the architecture of the Czech National Open Data Catalog (NODC).

It is built using LinkedPipes ETL and LinkedPipes DCAT-AP Viewer (see DCAT-AP Representation of Czech National Open Data Catalog and its Impact [7] for details) and it represents a typical LD processor. It both consumes LD on the input as it harvests local data catalogs, and produces LD on the output through the LD interface. The LD interface of NODC is used by data consumers and their tools to find relevant data in the catalog. One of those data consumers is the Official portal for European data (data.europa.eu), which collects metadata about data provided throughout Europe.

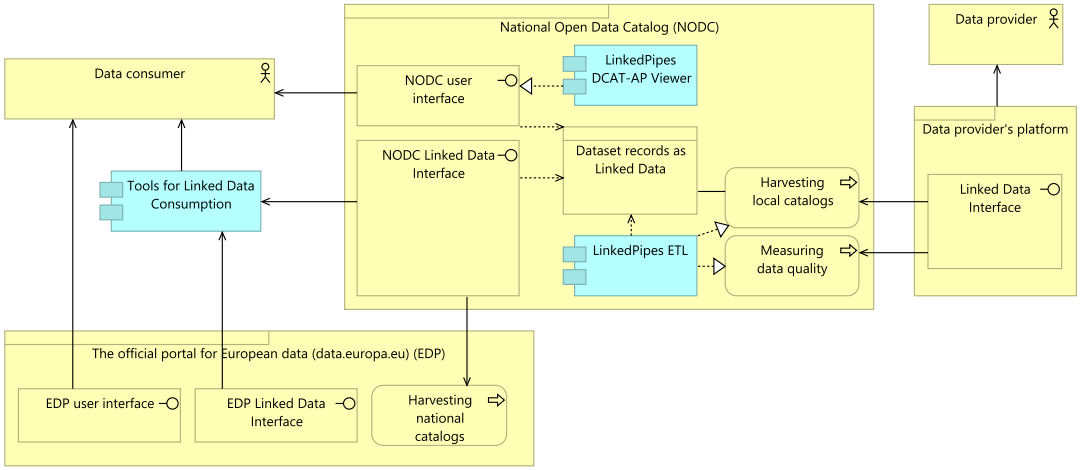

Figure 10: Architecture of the Czech National Open Data Catalog as an example of a data processor, with tools supporting it

Overall architecture

For the image of the LD data exchange architecture to be complete, in Figure 11 we present the entire architecture of Linked Data exchange among data providers, data consumers and data catalogs, including their interconnections.

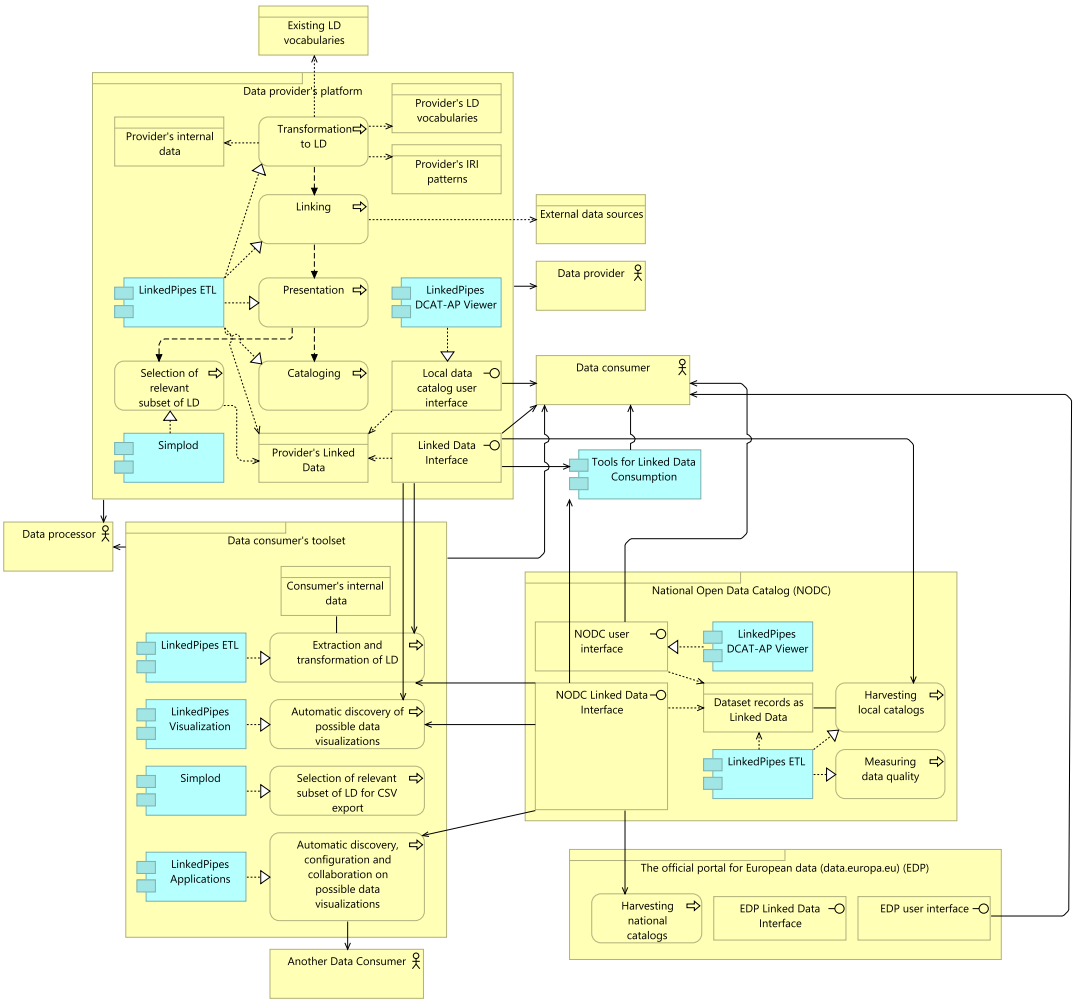

Figure 11: Overall architecture of data exchange using Linked Data with the supporting tools presented in this thesis

Bibliography

[1] Keith Alexander, Richard Cyganiak, Jun Zhao, and Michael Hausenblas. Describing Linked Datasets with the VoID Vocabulary. W3C note, W3C, March 2011. https://www.w3.org/TR/2011/NOTE-void-20110303/.

[2] Alejandra Gonzalez Beltran, Riccardo Albertoni, Peter Winstanley, Andrea Perego, Simon Cox, and David Browning. Data Catalog Vocabulary (DCAT) - Version 2. W3C Recommendation, W3C, February 2020. https://www.w3.org/TR/2020/REC-vocab-dcat-2-20200204/.

[3] Tim Berners-Lee, Roy T. Fielding, and Henrik Frystyk Nielsen. Hypertext Transfer Protocol – HTTP/1.0. RFC 1945, IETF, May 1996. https://datatracker.ietf.org/doc/html/rfc1945.

[4] Tim Berners-Lee, Larry Masinter, and Mark McCahill. Uniform Resource Locators (URL). RFC 1738, IETF, December 1994. https://datatracker.ietf.org/doc/html/rfc1738.

[5] Jeremy Carroll and Graham Klyne. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation, W3C, February 2004. https://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

[6] Antonín Jareš and Jakub Klímek. Simplod: Simple SPARQL Query Builder for Rapid Export of Linked Open Data in the Form of CSV Files. In Eric Pardede, Maria Indrawan-Santiago, Pari Delir Haghighi, Matthias Steinbauer, Ismail Khalil, and Gabriele Kotsis, editors, *iiWAS2021: The 23rd International Conference on Information Integration and Web Intelligence, Linz, Austria, 29 November 2021 - 1 December 2021*, pages 415–418. ACM, 2021. doi:10.1145/3487664.3487790.

[7] Jakub Klímek. DCAT-AP representation of Czech National Open Data Catalog and its impact. *J. Web Semant.*, 55:69–85, 2019. doi:10.1016/j.websem.2018.11.001.

[8] Jakub Klímek, Jiří Helmich, and Martin Nečaský. LinkedPipes Visualization: Simple Useful Linked Data Visualization Use Cases. In Harald Sack, Giuseppe Rizzo, Nadine Steinmetz, Dunja Mladenic, Sören Auer, and Christoph Lange, editors, *The Semantic Web - ESWC 2016 Satellite Events, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected Papers*, volume 9989 of *Lecture Notes in Computer Science*, pages 112–117, 2016. doi:10.1007/978-3-319-47602-5_23.

[9] Jakub Klímek, Jan Kučera, Martin Nečaský, and Dušan Chlapek. Publication and usage of official Czech pension statistics Linked Open Data. *J. Web Semant.*, 48:1–21, 2018. doi:10.1016/j.websem.2017.09.002.

[10] Jakub Klímek, Altynbek Orumbayev, Marzia Cutajar, Esteban Jenkins, Ivan Latták, Alexandr Mansurov, and Jiří Helmich. LinkedPipes Applications - Automated Discovery of Configurable Linked Data Applications. In Andreas

Harth, Valentina Presutti, Raphaël Troncy, Maribel Acosta, Axel Polleres, Javier D. Fernández, Josiane Xavier Parreira, Olaf Hartig, Katja Hose, and Michael Cochez, editors, *The Semantic Web: ESWC 2020 Satellite Events - ESWC 2020 Satellite Events, Heraklion, Crete, Greece, May 31 - June 4, 2020, Revised Selected Papers*, volume 12124 of *Lecture Notes in Computer Science*, pages 146–151. Springer, 2020. doi:10.1007/978-3-030-62327-2_25.

[11] Jakub Klímek and Petr Škoda. LinkedPipes ETL in use: practical publication and consumption of linked data. In Maria Indrawan-Santiago, Matthias Steinbauer, Ivan Luiz Salvadori, Ismail Khalil, and Gabriele Anderst-Kotsis, editors, *Proceedings of the 19th International Conference on Information Integration and Web-based Applications & Services, iiWAS 2017, Salzburg, Austria, December 4-6, 2017*, pages 441–445. ACM, 2017. doi:10.1145/3151759.3151809.

[12] Jakub Klímek and Petr Škoda. Speeding up Publication of Linked Data Using Data Chunking in LinkedPipes ETL. In Hervé Panetto, Christophe Debruyne, Walid Gaaloul, Mike P. Papazoglou, Adrian Paschke, Claudio Agostino Ardagna, and Robert Meersman, editors, *On the Move to Meaningful Internet Systems. OTM 2017 Conferences - Confederated International Conferences: CoopIS, C&TC, and ODBASE 2017, Rhodes, Greece, October 23-27, 2017, Proceedings, Part II*, volume 10574 of *Lecture Notes in Computer Science*, pages 144–160. Springer, 2017. doi:10.1007/978-3-319-69459-7_10.

[13] Jakub Klímek and Petr Škoda. LinkedPipes DCAT-AP Viewer: A Native DCAT-AP Data Catalog. In Marieke van Erp, Medha Atre, Vanessa López, Kavitha Srinivas, and Carolina Fortuna, editors, *Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, USA, October 8th - to - 12th, 2018*, volume 2180 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2018. http://ceur-ws.org/Vol-2180/paper-32.pdf.

[14] Jakub Klímek, Petr Škoda, and Martin Nečaský. LinkedPipes ETL: Evolved Linked Data Preparation. In Harald Sack, Giuseppe Rizzo, Nadine Steinmetz, Dunja Mladenic, Sören Auer, and Christoph Lange, editors, *The Semantic Web - ESWC 2016 Satellite Events, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected Papers*, volume 9989 of *Lecture Notes in Computer Science*, pages 95–100, 2016. doi:10.1007/978-3-319-47602-5_20.

[15] Jakub Klímek, Petr Škoda, and Martin Nečaský. Survey of tools for Linked Data consumption. *Semantic Web*, 10(4):665–720, 2019. doi:10.3233/SW-180316.

[16] Dave Longley, Gregg Kellogg, and Pierre-Antoine Champin. JSON-LD 1.1. W3C Recommendation, W3C, July 2020. https://www.w3.org/TR/2020/REC-json-ld11-20200716/.

[17] Eric Prud'hommeaux and Gavin Carothers. RDF 1.1 Turtle. W3C Recommendation, W3C, February 2014. https://www.w3.org/TR/2014/REC-turtle-20140225/.

[18] Andy Seaborne and Steven Harris. SPARQL 1.1 query language. W3C Recommendation, W3C, March 2013. https://www.w3.org/TR/2013/REC-sparql11-query-20130321/.

[19] Jeni Tennison and Gregg Kellogg. Metadata Vocabulary for Tabular Data. W3C Recommendation, W3C, December 2015. https://www.w3.org/TR/2015/REC-tabular-metadata-20151217/.

[20] Pierre-Yves Vandenbussche and Bernard Vatant. Linked Open Vocabularies. *ERCIM News*, 2014(96), 2014. https://ercim-news.ercim.eu/en96/special/linked-open-vocabularies.

[21] Ruben Verborgh, Miel Vander Sande, Pieter Colpaert, Sam Coppens, Erik Mannens, and Rik Van de Walle. Web-Scale Querying through Linked Data Fragments. In Christian Bizer, Tom Heath, Sören Auer, and Tim Berners-Lee, editors, *Proceedings of the Workshop on Linked Data on the Web co-located with the 23rd International World Wide Web Conference (WWW 2014), Seoul, Korea, April 8, 2014*, volume 1184 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2014. http://ceur-ws.org/Vol-1184/ldow2014_paper_04.pdf.

1. LinkedPipes ETL: Evolved Linked Data Preparation

Abstract As Linked Data gains traction, the proper support for its publication and consumption is more important than ever. Even though there is a multitude of tools for preparation of Linked Data, they are still either quite limited, difficult to use or not compliant with recent W3C Recommendations. In this demonstration paper, we present LinkedPipes ETL, a lightweight, Linked Data preparation tool. It is focused mainly on smooth user experience including mobile devices, ease of integration based on full API coverage and universal usage thanks to its library of components. We build on our experience gained by development and use of UnifiedViews, our previous Linked Data ETL tool, and present four use cases in which our new tool excels in comparison.

Reference Jakub Klímek, Petr Škoda, and Martin Nečaský. LinkedPipes ETL: Evolved Linked Data Preparation. In Harald Sack, Giuseppe Rizzo, Nadine Steinmetz, Dunja Mladenic, Sören Auer, and Christoph Lange, editors, *The Semantic Web - ESWC 2016 Satellite Events, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected Papers*, volume 9989 of *Lecture Notes in Computer Science*, pages 95–100, 2016. doi:10.1007/978-3-319-47602-5_20

2. Speeding up publication of Linked Data using data chunking in LinkedPipes ETL

Abstract There is a multitude of tools for preparation of Linked Data from data sources such as CSV and XML files. These tools usually perform as expected when processing examples, or smaller real world data. However, a majority of these tools become hard to use when faced with a larger dataset such as hundreds of megabytes large CSV file. Tools which load the entire resulting RDF dataset into memory usually have memory requirements unsatisfiable by commodity hardware. This is the case of RDF-based ETL tools. Their limits can be avoided by running them on powerful and expensive hardware, which is, however, not an option for majority of data publishers. Tools which process the data in a streamed way tend to have limited transformation options. This is the case of text-based transformations, such as XSLT, or per-item SPARQL transformations such as the streamed version of TARQL. In this paper, we show how the power and transformation options of RDF-based ETL tools can be combined with the possibility to transform large datasets on common consumer hardware for so called chunkable data - data which can be split in a certain way. We demonstrate our approach in our RDF-based ETL tool, LinkedPipes ETL. We include experiments on selected real world datasets and a comparison of performance and memory consumption of available tools.

Reference Jakub Klímek and Petr Škoda. Speeding up Publication of Linked Data Using Data Chunking in LinkedPipes ETL. In Hervé Panetto, Christophe Debruyne, Walid Gaaloul, Mike P. Papazoglou, Adrian Paschke, Claudio Agostino Ardagna, and Robert Meersman, editors, *On the Move to Meaningful Internet Systems. OTM 2017 Conferences - Confederated International Conferences: CoopIS, C&TC, and ODBASE 2017, Rhodes, Greece, October 23-27, 2017, Proceedings, Part II*, volume 10574 of *Lecture Notes in Computer Science*, pages 144–160. Springer, 2017. doi:10.1007/978-3-319-69459-7_10

3. LinkedPipes ETL in Use: Practical Publication and Consumption of Linked Data

Abstract More and more companies and institutions realize the potential of Linked Open Data (LOD) and are motivated to start publishing their own data as LOD and consuming existing LOD datasets. However, e.g. in institutions of public administration, publishing LOD is still a challenging task. One of the main reasons is a lack of user friendly tooling which would properly support the whole LOD publishing process. The process typically consists of source data extraction, transformation to RDF, alignment with commonly used vocabularies, linking to other datasets, computing metadata, publishing on the web as a dump, loading into a triplestore and recording the dataset in a data catalog such as CKAN. In this paper we present LinkedPipes ETL, a tool for ETL like LOD publishing, which mainly focuses on supporting such LOD publishing workflows in a user friendly way. In addition, the tool also eases consumption of already existing LOD data sources as it addresses some of the practical issues associated with it. Finally, the tool itself uses Linked Data technologies for representation of the ETL processes. We describe LinkedPipes ETL and its main distinguishing features in context of the use cases in which the tool has already been deployed. They include an institution of public administration, a municipality, a university, a software company and an open data initiative.

Reference Jakub Klímek and Petr Škoda. LinkedPipes ETL in use: practical publication and consumption of linked data. In Maria Indrawan-Santiago, Matthias Steinbauer, Ivan Luiz Salvadori, Ismail Khalil, and Gabriele Anderst-Kotsis, editors, *Proceedings of the 19th International Conference on Information Integration and Web-based Applications & Services, iiWAS 2017, Salzburg, Austria, December 4-6, 2017*, pages 441–445. ACM, 2017. doi:10.1145/3151759.3151809

4. Publication and Usage of Official Czech Pension Statistics Linked Open Data

Abstract Linked Open Data (LOD) principles are known for a decade now and there are thousands of LOD datasets of variable quality and importance. They often come from academic research projects, which show how LOD can be published, how it can be useful, etc. However, these projects last only for a few years and when they end, the datasets often cease to be maintained. What is needed is to convince the owner of the original data, e.g. an organization in public administration, to keep on publishing LOD on their own even when the project ends. Therefore, it is noteworthy and admirable, when this happens. In this paper we describe how the Czech Social Security Administration (CSSA) publishes official pension statistics as LOD as part of their day-to-day operation, which was jump-started by an applied research project. The data is modeled using the Simple Knowledge Organization System (SKOS) vocabulary and the RDF Data Cube Vocabulary (DCV). It is published as RDF data dumps, as a SPARQL endpoint and using IRI dereferencing for semantic web technologies power users. For journalists and other users without knowledge of these technologies, the data is also published as CSV files and visualizations generated from the LOD. We show how the data is reused in applications and how it contributes to statistical indicators in combination with other LOD.

Reference Jakub Klímek, Jan Kučera, Martin Nečaský, and Dušan Chlapek. Publication and usage of official Czech pension statistics Linked Open Data. *J. Web Semant.*, 48:1–21, 2018. doi:10.1016/j.websem.2017.09.002

5. DCAT-AP Representation of Czech National Open Data Catalog and its Impact

Abstract Open data is now a heavily discussed topic around the world and in the European Union. In the Czech Republic, open data is a term anchored in legislation, which includes the requirement of registration of all open data in the Czech National Open Data Portal (NODC). In this paper we describe the NODC, its architecture, dataset registration processes including the harvesting of Local Open Data Catalogs (LODCs), proprietary XML API and its obsolete dataset viewer. Next we describe the process of transformation of the NODC metadata to the DCAT-AP v1.1 RDF representation from the data model point of view and from the technical environment point of view. We describe the dataset quality measurements computed using the new data representation and its further impact on the Linked Open Data (LOD) environment including the harvesting of the metadata by the European Data Portal (EDP). Finally, we evaluate the data transformation and publishing environment from the usability, portability, availability and performance perspectives.

Reference Jakub Klímek. DCAT-AP representation of Czech National Open Data Catalog and its impact. *J. Web Semant.*, 55:69–85, 2019. doi:10.1016/j.websem.2018.11.001

6. LinkedPipes DCAT-AP Viewer: A Native DCAT-AP Data Catalog

Abstract In this demonstration we present LinkedPipes DCAT-AP Viewer (LP-DAV), a data catalog built to support DCAT-AP, the European standard for representation of metadata in data portals, and an application profile of the DCAT W3C Recommendation. We present its architecture and data loading process and on the example of the Czech National Open Data portal we show its main advantages compared to other data catalog solutions such as CKAN. These include the support for Named Authority Lists in EU Vocabularies (EU NALs), controlled vocabularies mandatory in DCAT-AP, and the support for bulk loading of DCAT-AP RDF dumps using LinkedPipes ETL.

Reference Jakub Klímek and Petr Škoda. LinkedPipes DCAT-AP Viewer: A Native DCAT-AP Data Catalog. In Marieke van Erp, Medha Atre, Vanessa López, Kavitha Srinivas, and Carolina Fortuna, editors, *Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, USA, October 8th - to - 12th, 2018*, volume 2180 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2018. `http://ceur-ws.org/Vol-2180/paper-32.pdf`

7. LinkedPipes Visualization: Simple Useful Linked Data Visualization Use Cases

Abstract There is a need for being able to effectively demonstrate the benefits of publishing Linked Data. There are already many datasets and they are no longer limited to research based data sources. Governments and even companies start publishing Linked Data as well. However, a tool, which would be able to immediately demonstrate the Linked Data benefits to those, who still need convincing, was missing. In this paper, we demonstrate LinkedPipes Visualization, a tool based on our previous work, the Linked Data Visualization Model. Using this tool, we show four simple use cases that immediately demonstrate the Linked Data benefits. We demonstrate the value of providing dereferenceable IRIs and using vocabularies standardized as W3C Recommendations on use cases based on SKOS and the RDF Data Cube Vocabulary, providing data visualizations on one click. LinkedPipes Visualization can be extended to support other vocabularies through additional visualization components.

Reference Jakub Klímek, Jiří Helmich, and Martin Nečaský. LinkedPipes Visualization: Simple Useful Linked Data Visualization Use Cases. In Harald Sack, Giuseppe Rizzo, Nadine Steinmetz, Dunja Mladenic, Sören Auer, and Christoph Lange, editors, *The Semantic Web - ESWC 2016 Satellite Events, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected Papers*, volume 9989 of *Lecture Notes in Computer Science*, pages 112–117, 2016. doi:10.1007/978-3-319-47602-5_23

8. Survey of Tools for Linked Data Consumption

Abstract There is a large number of datasets published as Linked (Open) Data (LOD/LD). At the same time, there is also a multitude of tools for publication of LD. However, potential LD consumers still have difficulty discovering, accessing and exploiting LD. This is because compared to consumption of traditional data formats such as XML and CSV files, there is a distinct lack of tools for consumption of LD. The promoters of LD use the well-known 5-star Open Data deployment scheme to suggest that consumption of LD is a better experience once the consumer knows RDF and related technologies. This suggestion, however, falls short when the consumers search for an appropriate tooling support for LD consumption. In this paper we define a LD consumption process. Based on this process and current literature, we define a set of 34 requirements a hypothetical Linked Data Consumption Platform (LDCP) should ideally fulfill. We cover those requirements with a set of 94 evaluation criteria. We survey 110 tools identified as potential candidates for an LDCP, eliminating them in 3 rounds until 16 candidates for remain. We evaluate the 16 candidates using our 94 criteria. Based on this evaluation we show which parts of the LD consumption process are covered by the 16 candidates. Finally, we identify 8 tools which satisfy our requirements on being a LDCP. We also show that there are important LD consumption steps which are not sufficiently covered by existing tools. The authors of LDCP implementations may use this survey to decide about directions of future development of their tools. LD experts may use it to see the level of support of the state of the art technologies in existing tools. Non-LD experts may use it to choose a tool which supports their LD processing needs without requiring them to have expert knowledge of the technologies. The paper can also be used as an introductory text to LD consumption.

Reference Jakub Klímek, Petr Škoda, and Martin Nečaský. Survey of tools for Linked Data consumption. *Semantic Web*, 10(4):665–720, 2019. doi:10.3233/SW-180316

9. LinkedPipes Applications - Automated Discovery of Configurable Linked Data Applications

Abstract Consumption of Linked Data (LD) is a far less explored problem than its production. LinkedPipes Applications (LP-APPs) is a platform enabling data analysts and data journalists to easily create LD based applications such as, but not limited to, visualizations. It builds on our previous research regarding the automatic discovery of possible visualizations of LD. The approach was based on the matching of classes and predicates used in the data, e.g. in a form of a data sample, to what an application or visualization expects, e.g. in a form of a SPARQL query, solving potential mismatches in data by dynamically applying data transformers. In this demo, we present a platform that allows a data analyst to automatically discover possible visualizations of a given LD data source using this method and the applications contained in the platform. Next, the data analyst is able to configure the discovered visualization application and publish it or embed it in an arbitrary web page. Thanks to the configuration being stored in their Solid POD, multiple analysts are able to collaborate on a single application in a decentralized fashion. The resulting visualization application can be kept up to date via scheduling an ETL pipeline, regularly refreshing the underlying data.

Reference Jakub Klímek, Altynbek Orumbayev, Marzia Cutajar, Esteban Jenkins, Ivan Latták, Alexandr Mansurov, and Jiří Helmich. LinkedPipes Applications - Automated Discovery of Configurable Linked Data Applications. In Andreas Harth, Valentina Presutti, Raphaël Troncy, Maribel Acosta, Axel Polleres, Javier D. Fernández, Josiane Xavier Parreira, Olaf Hartig, Katja Hose, and Michael Cochez, editors, *The Semantic Web: ESWC 2020 Satellite Events - ESWC 2020 Satellite Events, Heraklion, Crete, Greece, May 31 - June 4, 2020, Revised Selected Papers*, volume 12124 of *Lecture Notes in Computer Science*, pages 146–151. Springer, 2020. doi:10.1007/978-3-030-62327-2_25

10. Simplod: Simple SPARQL Query Builder for Rapid Export of Linked Open Data in the Form of CSV Files

Abstract In the last decade, linked open data (LOD) became the de-facto highest standard of publishing data on the Web, a.k.a. 5-star open data. One of the advantages of LOD is better data discovery thanks to the linkable nature of the data. Unfortunately, in the wider IT expert community, RDF and SPARQL is considered unnecessarily complex, hard to understand and hard to process, especially when transferring open data in bulk. However, it is this wider IT expert community which is the one supposed to both publish data produced in the public administration as open data, and to build consumer-facing applications on top of the data. In this paper, we propose an approach to making LOD easily accessible to the community used to CSV files. We propose a tool called Simplod focused on rapid, straightforward and customizable formulation of a SPARQL SELECT query intended for customizable bulk transformation of LOD published in SPARQL endpoints into CSV files. In addition, Simplod configurations can be stored in and shared via Solid pods.

Reference Antonín Jareš and Jakub Klímek. Simplod: Simple SPARQL Query Builder for Rapid Export of Linked Open Data in the Form of CSV Files. In Eric Pardede, Maria Indrawan-Santiago, Pari Delir Haghighi, Matthias Steinbauer, Ismail Khalil, and Gabriele Kotsis, editors, *iiWAS2021: The 23rd International Conference on Information Integration and Web Intelligence, Linz, Austria, 29 November 2021 - 1 December 2021*, pages 415–418. ACM, 2021. doi:10.1145/3487664.3487790