LinkedPipes Applications - Automated Discovery of Configurable Linked Data Applications*

Jakub Klímek[0000−0001−7234−3051], Altynbek Orumbayev, Marzia Cutajar, Esteban Jenkins, Ivan Latták, Alexandr Mansurov, and Jiří Helmich[0000−0001−5048−5366]

Department of Software Engineering, Faculty of Mathematics and Physics, Charles University, Malostranské náměstí 25, 118 00 Praha 1, Czech Republic
klimek@ksi.mff.cuni.cz
https://jakub.klímek.com

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.

Keywords: linked data · consumption · visualization · application · solid

1 Introduction

Nowadays, there are many linked data (LD) sources available, either within enterprise knowledge graphs, or as linked open data (LOD) publicly available on the Web. There has been a lot of research done on how to create LD. It is created either by transforming legacy data, e.g. by defining ETL processes [9] or by building declarative mappings [3], or by producing LD natively, e.g. from IoT sensors in a form of RDF streams [1], etc. Note that an ETL (extract-transform-load) process is a data transformation paradigm, where data is first *extracted*

* This work was supported by the Czech Science Foundation (GAČR), grant number 19-01641S.

from its original location, *transformed* by the data processing system and *loaded* to the target database. Once a LD dataset is created and accessible, either as an RDF dump, in a SPARQL endpoint, through IRI dereference or as a Linked Data Fragments server [13], there is an issue regarding its discoverability, i.e. how do the users find it. Partially, the issue of discoverability is tackled by, e.g. the Linked Open Data Cloud[1], or by a recently initialized list of known SPARQL endpoints[2]. This is, however, still far from ideal, as we show in [10]. Let us now assume that the potential consumer of the published dataset manages to find it.

Once an LD dataset is found by the consumer, they need to figure out how to work with it. This may include studying the dataset documentation, if available, or browsing through the dataset using one of the publisher provided data browsers such as the OpenLink Virtuoso Faceted Browser[3] or the more sophisticated Linked Data Reactor [8], if deployed. Typically, the process of getting to know a new dataset comes down to querying the data using dataset-independent exploratory SPARQL queries, such as:

1. Discover classes used in the data
2. Pick a few instances of the interesting classes
3. Discover predicates used by the picked instances

Only after the user gets to know the data, they are able to use it, e.g. in their own applications or visualizations. This is a critical part of the dataset consumption process when the user needs to extract the data relevant for the desired application and match the format to their tool of choice. However, there is information present in an LD dataset, such as the used classes and predicates, which could be exploited by an LD-enabled tool to match the data to its possible visualizations or other usages automatically.

In this demonstration, we present LinkedPipes Applications (LP-APPs) - a platform offering an alternative approach to the discovery of visualization applications applicable to a given LD data source, exploiting semantic information present in LD datasets and descriptions of expected input by applications in the platform. The discovered applications can be configured for better presentation to end users and published or embedded in a web page. The visualization configuration itself is stored in Solid PODs, which we exploit to provide collaboration capabilities. In case the source data changes over time, it might be necessary to periodically re-create the visualizations to keep them up to date, which is also supported by the platform. A Solid POD [11] is a personal online datastore, based on standard web technologies and separated from any applications, which can access and modify the stored data based on the users' permissions.

2 LinkedPipes Applications (LP-APPs)

The LP-APPs platform has a goal of easing creation and configuration of LD based visualizations in form of interactive applications. It supports two principal

[1] `https://lod-cloud.net/`
[2] `https://github.com/OpenLinkSoftware/lod-cloud`
[3] `http://vos.openlinksw.com/owiki/wiki/VOS/VirtFacetBrowserInstallConfig`

user roles. The first user role is the analyst, or journalist, who wants to prepare an LD based interactive application, which can be published or embedded in a news article. The second user role is the end user who can view the application and play with its interactive controls, such as filters. While the applications are created to be used by the end users, the platform focuses mainly on the analysts, to whom it offers features easing the creation of such interactive applications based on LD. LP-APPs builds on our previous research regarding automatic discovery of ways to visualize a given LD datasource [6,12], and focuses on the application configuration and publishing part of the process. The LP-APPs platform is open-source, hosted on GitHub[4] and can be run simply via Docker compose. From the point of view of the analyst, this is the typical workflow using the platform:

1. WebID login Since all the visualization configurations are stored in the analysts' Solid PODs, the user needs to do login using their WebID first.

2. Choice of LD data source From the dashboard, the user starts with selecting a LD data source. This can be either a SPARQL endpoint or an RDF dump - either via a direct file upload, or via a URL. There is also a set of showcase data sources for various application types.

3. Automatic discovery of supported visualization techniques In this step, the platform uses the automated discovery process described in [12]. This is why, in addition to the data source, a small, but structurally representative data sample is required. For small datasets, the sample can be the same as the dataset, which is enough for the purpose of this demonstration. For larger datasets, the data sample can be created manually e.g. by separating one representative entity instance from the dataset. It could also be obtained automatically using techniques for dataset profiling [4], but this is not implemented yet. This step could also be substituted for a step generating an artificial data sample based on pre-existing metadata describing the dataset, e.g. using the VoID Vocabulary [5]. However, these descriptions, when present, are often inaccurate and outdated.

4. Choice of desired technique in case of multiple options When multiple applications can be applied to the selected data source, the analyst chooses the desired one. This step is skipped, if there is only one application supported.

5. Actual data transformation using LinkedPipes ETL Now that it is known which application is to be applied to the input data, a set of LinkedPipes ETL [9] data transformation pipelines is constructed automatically. Each pipeline takes the input data, and applies a set of data transformations, which lead to data compatible with the chosen application. If there is more than one possibility of how to transform the input data for the selected application, the user chooses one. This step is skipped, if there is only one transformation available.

[4] `https://github.com/linkedpipes/applications`

6. Initial visualization in selected application Once the input data is transformed to a form consumable by the chosen application, the application is opened and the analyst can see the data in it.

7. Configuration of application In this phase, the analyst can specify, which interactive controls such as filters and their values will be available to the end users. This is because a generic visualization might need to be, for instance, limited, to show only what is relevant to the end users. The application configuration is stored in the analyst's Solid POD [11], which also facilitates collaboration with other analysts and propagation of updates of the application configuration to the end users.

8. Application publication or embedding Once the analyst is happy with the application configuration, they can publish it, i.e. get its public URL, or generate an embedding snippet using HTML's `iframe`.

9. Data refresh scheduling If the source data is regularly updated, and the analyst wishes to update the resulting application accordingly, they can schedule regular updates of the underlying data via scheduling of the LinkedPipes ETL pipeline.

3 Demonstration

The demonstration will be performed on our live demo instance running on `https://applications.linkedpipes.com/`. The source code is hosted on GitHub `https://github.com/linkedpipes/applications`.

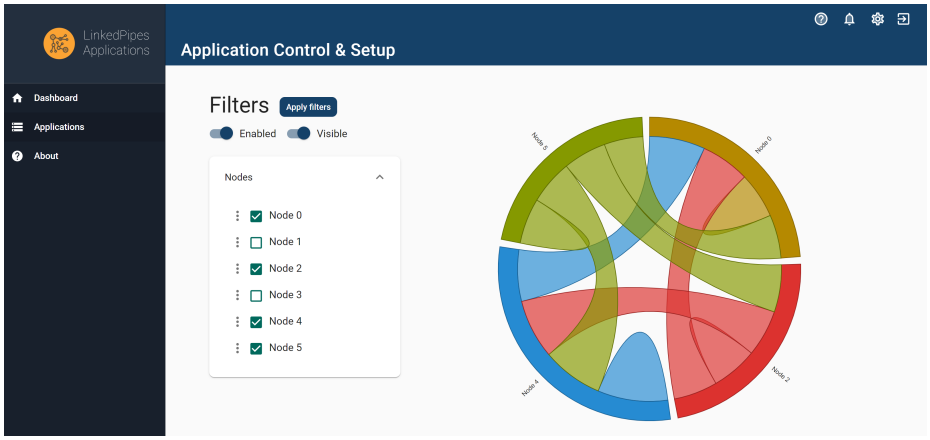

Fig. 1. Chord visualization application in LinkedPipes Applications

We will go through the basic workflow described in section 2, resulting in applications in a form of chord visualization (see Figure 1), Treemap visualization and a Timeline visualization.

In addition, we will demonstrate the benefits of the application configuration being stored in a Solid POD. Besides the usual benefits of the decentralized storage approach, we will show how the analysts can collaborate on configuration of a single visualization application thanks to the configurations being stored in Solid PODs.

4 Conclusions and Future Work

This demonstration shows how LP-APPs can be used to provide configurable visualization applications based on data in a linked data source. However, the platform is not limited to visualization applications. Using the same mechanism, any application such as a contact list manager or statistical data cube tool can be matched to a given data source. This is, as in the demonstration, provided that the data source contains data, which can either directly, or through a data transformation pipeline work with the application. A similar approach could also be adapted to match applications to data in Solid PODs.

On the other hand, the approach has some drawbacks. Mainly, even if LP-APPs finds a match of an application to a data source based on its data sample, it does not always mean that the actual data contains what is necessary for the application to work. Also, given that the approach is based on ETL, it does not work well for larger datasets. An alternative approach could be based on rewriting of SPARQL queries. A query used by the application to query data could be gradually rewritten in the opposite direction of the ETL pipeline, to match the data in the original data source. This could allow us to process even larger datasets. Finally, the Solid specification and implementation is quite live at the moment, changing frequently, breaking existing implementations. This is, however, to be expected at this stage of development.

References

1. Barbieri, D.F., Braga, D., Ceri, S., Valle, E.D., Grossniklaus, M.: C-SPARQL: a continuous query language for RDF data streams. Int. J. Semantic Computing **4**(1), 3–25 (2010). https://doi.org/10.1142/S1793351X10000936
2. Bizer, C., Heath, T., Auer, S., Berners-Lee, T. (eds.): 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, CEUR Workshop Proceedings, vol. 1184. CEUR-WS.org (2014), `http://ceur-ws.org/Vol-1184`
3. Dimou, A., Sande, M.V., Colpaert, P., Verborgh, R., Mannens, E., de Walle, R.V.: RML: A generic language for integrated RDF mappings of heterogeneous data. In: Bizer et al. [2], `http://ceur-ws.org/Vol-1184/ldow2014_paper_01.pdf`
4. Ellefi, M.B., Bellahsene, Z., Breslin, J.G., Demidova, E., Dietze, S., Szymanski, J., Todorov, K.: RDF dataset profiling - a survey of features, methods, vocabularies and applications. Semantic Web **9**(5), 677–705 (2018). https://doi.org/10.3233/SW-180294
5. Hausenblas, M., Cyganiak, R., Alexander, K., Zhao, J.: Describing Linked Datasets with the VoID Vocabulary". W3C note, W3C (Mar 2011), `https://www.w3.org/TR/2011/NOTE-void-20110303/`

6. Helmich, J., Potoček, T., Klímek, J., Nečaský, M.: Towards easier visualization of linked data for lay users. In: Akerkar, R., Cuzzocrea, A., Cao, J., Hacid, M. (eds.) Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics, WIMS 2017, Amantea, Italy, June 19-22, 2017. pp. 12:1–12:9. ACM (2017). https://doi.org/10.1145/3102254.3102261
7. Indrawan-Santiago, M., Steinbauer, M., Salvadori, I.L., Khalil, I., Anderst-Kotsis, G. (eds.): Proceedings of the 19th International Conference on Information Integration and Web-based Applications & Services, iiWAS 2017, Salzburg, Austria, December 4-6, 2017. ACM (2017). https://doi.org/10.1145/3151759
8. Khalili, A., de Graaf, K.A.: Linked Data Reactor: Towards Data-Aware User Interfaces. In: Proceedings of the 13th International Conference on Semantic Systems. p. 168–172. Semantics2017, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3132218.3132231
9. Klímek, J., Škoda, P.: LinkedPipes ETL in use: practical publication and consumption of linked data. In: Indrawan-Santiago et al. [7], pp. 441–445. https://doi.org/10.1145/3151759.3151809
10. Klímek, J., Škoda, P., Nečaský, M.: Survey of tools for Linked Data consumption. Semantic Web **10**(4), 665–720 (2019). https://doi.org/10.3233/SW-180316
11. Mansour, E., Sambra, A.V., Hawke, S., Zereba, M., Capadisli, S., Ghanem, A., Aboulnaga, A., Berners-Lee, T.: A demonstration of the solid platform for social web applications. In: Proceedings of the 25th International Conference Companion on World Wide Web. p. 223–226. WWW '16 Companion, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2016). https://doi.org/10.1145/2872518.2890529
12. Nečaský, M., Helmich, J., Klímek, J.: Platform for automated previews of linked data. In: Indrawan-Santiago et al. [7], pp. 395–404. https://doi.org/10.1145/3151759.3151769
13. Verborgh, R., Sande, M.V., Colpaert, P., Coppens, S., Mannens, E., de Walle, R.V.: Web-Scale Querying through Linked Data Fragments. In: Bizer et al. [2], `http://ceur-ws.org/Vol-1184/ldow2014_paper_04.pdf`