RDF-based Deployment Pipelining for Efficient Dataset Release Management

Claus Stadler[1], Lisa Wenige[1], Sebastian Tramp[2], Kurt Junghanns[1], and Michael Martin[1]

[1] Institute for Applied Informatics (InfAI), Goerdelerring 9, Leipzig, Germany, D-04109
E-mail: (wenige, stadler, junghanns, martin)@infai.org
[2] eccenca GmbH, Hainstrasse 8, Leipzig, Germany, D-04109
E-mail: info@eccenca.com

Abstract. Open Data portals often struggle to provide release features (i.e., stable versioning, up-to-date download links, rich metadata descriptions) for their datasets. By this means, wide adoption of publicly available datasets is hindered, since consuming applications cannot access fresh data sources or might break due to data quality issues. While there exists a variety of tools to efficiently control release processes in software development, the management of dataset releases is not as clear. This paper proposes a deployment pipeline for efficient dataset releases that is based on automated enrichment of DCAT/DATAID metadata and is a first step towards efficient deployment pipelining for Open Data publishing.

Keywords: Deployment, Open Data, DCAT, Data Quality

1 Introduction

With the advent of the Open Data movement, a multitude of datasets have been made available on public repositories.[3,4] In terms of the organizational surroundings, researchers have also developed methodologies in order to manage the data publishing process efficiently [4, 8, 10, 14]. In a more technical line of research, software tools have been developed that (semi-)automatically assist data publishers during data conversion [1, 16], quality assurance (i.e., checking the syntactic validity and semantic accuracy of datasets [5, 17], linking [12, 15], metadata enrichment and data provision [9, 13, 14].
However, the integration of these separate publishing phases has not been entirely addressed by academia. Instead, the combination of the required tasks is often handled by individuals who need to manually execute hand-crafted transformations to make up for the missing technical links between the existing data processing units. This hinders frequent releases of fresh datasets as well as the

[3] http://lodstats.aksw.org/
[4] https://datahub.io

reuse of data publishing pipelines across different application domains. Since quality-assurance, timeliness and discoverability are among the most important pre-conditions for stakeholders and applications to actually consume the data [10], these issues should have priority when publishing data collections. Hence, efficient software tools need to be in place to speed up publishing workflows and minimize manual adaptations thereby reducing error-proneness. In this context, the provision of interfaces and automatic tools for bridging the phases of data conversion/pre-processing and the final dataset release is particularly relevant. Here, the methods in the related and well-researched field of software engineering, such as *versioning, stable download links, automated testing and metadata enrichment* can serve as guiding best practices. However, the process of data publication also exhibits some specifics that need to be considered when developing deployment tools. In this paper, we present a methodology and a prototypical implementation of a deployment pipeline for automated release management of data collections based on RDF. We use a strict definition of *dataset* as the foundation for data management: A dataset is an instance of a datamodel. Hence, any procedures that yield (syntactic) representations of that instance, such as a CSV-to-RDF mapping and its materialization, are considered as different distribution forms of the *same* dataset.

2 Related Work

The research on efficient data release management is still in its infancy. In their survey paper on (linked) data consumption platforms, Klimek et al. rightly point out that there exist only a few tools that deal with dataset publication from an integrated perspective [9]. Some works present technological platforms to manage (linked) data lifecycles. For instance, Rojas Melendez et al. introduce a Linked Data publication platform that is based on RDF Streams and that provides data on the availability of parking lots in the Flanders region (Belgium) in an ad-hoc fashion [13].

The DataGraft application by Roman et al. enables data conversion, publication and reusability of transformation pipelines [14]. However, this tool misses the perspective that one dataset, although being issued by the same publishing body and containing the same kind of data, might evolve over time and should therefore be available in different versions that can be determined from a dataset catalog.

In software development, strategies of continuous integration (CI) are widespread. In CI, code repositories are frequently checked for updates and troubleshooting operations are triggered in case errors are found. Thus, it is ensured that released software fulfills qualitative requirements and that other applications can make use of the most recent updates [7]. However, CI methods from software development can not be transferred to data publishing 1:1. Proposals for adaptations have been made by Cirulli et al., who suggest to maintain the code that is used for data conversion tasks on a Jenkins server [3]. Meissner et al. present an integration pipeline that gets triggered as soon as changes are commited to a

repository [11]. But these approaches are missing a clear versioning concept on the data level as well as methods to automatically add metadata to the different dataset releases. The issue of automatically enhancing metadata descriptions is addressed by Frey et al., who present a novel strategy of fusing and publishing large datasets with their FlexiFusion approach [6].
However, a holistic method for deployment pipelining on common RHS that considers all relevant aspects of the dataset release process has not yet been presented.

3 Use Case - The LIMBO Project

The German Federal Ministry of Transport and Digital Infrastructure is providing the mCLOUD Open Data portal, where currently more than 1070 datasets on roads, rails, air traffic and waterways can be found. Additionally, the platform contains collections of space, climate and weather data. The datasets are published by the ministry as well as by its associated publicly funded agencies. By providing an Open Data portal, the ministry intends to harness research and development projects working on novel navigational services, smart travel and route planning as well as applications for highly precise weather forecasting[5].
However, the data on the portal is provided in heterogeneous formats, is not semantically described and poorly integrated with other data collections. Hence, the research project *Linked Data Services for Mobility* (LIMBO) has been started to convert selected datasets from the mCLOUD portal to RDF and link them to other datasets on the Linked Open Data cloud [6]. However, as the original data is subject to changes, conversion processes are re-run frequently thus making efficient deployment and release pipelines necessary.

4 Dataset Deployment Pipeline

It is sensible to publish converted RDF datasets on a repository management service, since such services already provide numerous useful features, which can be configured for dataset publication. Currently, the LIMBO datasets including the deployment build tools can be accessed on the public gitlab instance.[7] New (versions of) datasets are published by running build commands in a broadly predefined, but configurable order of execution. The build process triggers *data transformation*, *RDFUnit-based testing*[8] and *metadata enrichment* tasks 1. For instance, during *metadata enrichment*, the process assembles a local DCAT model of the dataset project and allows modifications to be specified with SPARQL update statements utilizing the tool *Sparql-Integrate*.[9] This tool is a

[5] `https://www.mcloud.de/`
[6] `https://www.limbo-project.org/`
[7] `https://gitlab.com/limbo-project.org`
[8] `https://github.com/AKSW/RDFUnit`
[9] `https://github.com/SmartDataAnalytics/SparqlIntegrate`

thin command-line wrapper for Jena's *ARQ*[10] SPARQL engine which registers several extensions, such as for passing environment variables from shell scripts to SPARQL queries. Upon release, a git tag is created and local file references are converted to public download URLs, yielding a DCAT record ready for publishing in a dataset catalog identifying releases with maven-style group, artifact and version identifiers as proposed by [2] and [6].

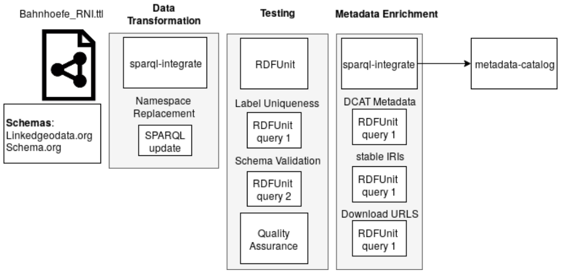

Fig. 1: Sparql-Integrate transformations

By this means, applications that rely on stable data can determine current releases from the metadata catalog.

5 Conclusion

In this paper, we have presented a git-based methodology on how to publish validated stable releases of datasets and corresponding DCAT records. Stable releases are beneficial for applications in many ways, such as when it comes to predictability reproducibility of results, testing, and caching of derived information for performance-reasons. Up to date, we published a couple of datasets of the mCLOUD with our approach. As this approach does not depend on hosting new custom services, maintenance overhead is minimized. The adoption of DataIDs and maven's artifact identification scheme promises future interoperability with other existing infrastructures for data asset management.

Acknowledgement

This work has been supported by the Federal Ministry of Transport and Digital Infrastructure (BMVI) under the grant number 19F2029.

References

1. Auer, S., Dietzold, S., Lehmann, J., Hellmann, S. and Aumueller, D. (2009, April). Triplify: light-weight linked data publication from relational databases. In Pro-

[10] `http://jena.apache.org/`

ceedings of the 18th international conference on World wide web (pp. 621-630). ACM.

2. Brümmer, M., Baron, C., Ermilov, I., Freudenberg, M., Kontokostas, D., Hellmann, S. (2014, September). DataID: Towards semantically rich metadata for complex datasets. In Proceedings of the 10th International Conference on Semantic Systems (pp. 84-91). ACM.
3. Cirulli, S. (2015). Continuous integration for XML and RDF Data. XML LONDON, 52-60.
4. COMSODE, A., Kučera, J., Nečaský, M., Klímek, J. and Chlapek, D. Open Data publication in a nutshell.
5. Dimou, A., Kontokostas, D., Freudenberg, M., Verborgh, R., Lehmann, J., Mannens, E., Hellmann, S. and Van de Walle, R. (2015, October). Assessing and Refining Mappings to RDF to Improve Dataset Quality. In International Semantic Web Conference (pp. 133-149). Springer, Cham.
6. Frey, J., Hofer, M., Hellmann, S., Obraczka, D. DBpedia FlexiFusion Best of Wikipedia¿ Wikidata¿ Your Data.
7. Fowler, M. and Foemmel, M. (2006). Continuous integration. Thought-Works. `https://www.martinfowler.com/articles/continuousIntegration.html`, 122?, 14?.
8. Klein, E., Gschwend, A. and Neuroni, A. C. (2016, May). Towards a Linked Data Publishing Methodology. In 2016 Conference for E-Democracy and Open Government (CeDEM) (pp. 188-196). IEEE.
9. Klímek, J., Skoda, P. and Necask, M. (2016). Requirements on Linked Data Consumption Platform. In LDOW at WWW.
10. Kucera, Jan, et al. Methodologies and Best Practices for Open Data Publication. DATESO. 2015.
11. Meissner, R., Junghanns, K. (2016, September). Using devOps principles to continuously monitor RDF data quality. In Proceedings of the 12th International Conference on Semantic Systems (pp. 189-192). ACM.
12. Ngomo, A. C. N. and Auer, S. (2011, June). LIMES a time-efficient approach for large-scale link discovery on the web of data. In Twenty-Second International Joint Conference on Artificial Intelligence.
13. Rojas Meléndez, J. A., Van de Vyvere, B., Gevaert, A., Taelman, R., Colpaert, P. and Verborgh, R. (2018). A Preliminary Open Data Publishing Strategy for Live Data in Flanders. In WWW2018, the International World Wide Web Conference (pp. 1847-1853). ACM Press.
14. Roman, D., Dimitrov, M., Nikolov, N., Putlier, A., Sukhobok, D., Elvester, B. and Petkov, Y. (2016, May). Datagraft: Simplifying open data publishing. In European Semantic Web Conference (pp. 101-106). Springer, Cham.
15. Schultz, A., Matteini, A., Isele, R., Bizer, C. and Becker, C. (2011, October). LDIF-linked data integration framework. In Proceedings of the Second International Conference on Consuming Linked Data-Volume 782 (pp. 125-130). CEUR-WS.org.
16. Unbehauen, J., Stadler, C. and Auer, S. (2012, December). Accessing relational data on the web with sparqlmap. In Joint International Semantic Technology Conference (pp. 65-80). Springer, Berlin, Heidelberg.
17. Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., and Auer, S. (2016). Quality assessment for linked data: A survey. Semantic Web, 7(1), 63-93.