Producing and Using Linked Open Government Data in the TWC LOGD Portal

Timothy Lebo, John S. Erickson, Li Ding,
Alvaro Graves, Gregory Todd Williams,
Dominic DiFranzo, Xian Li,
James Michaelis, Jin Guang Zheng, Johanna Flores, Zhenning Shangguan,
Deborah L. McGuinness and Jim Hendler

Tetherless World Constellation
Rensselaer Polytechnic Institute
110 8th St., Troy, NY 12180, USA

Abstract

As open government initiatives around the world publish an increasing number of raw datasets, citizens and communities face daunting challenges when organizing, understanding, and associating disparate data related to their interests. Immediate and incremental solutions are needed to integrate, collaboratively manipulate, and transparently consume large-scale distributed data. The Tetherless World Constellation (TWC) at Rensselaer Polytechnic Institute (RPI) has developed the TWC LOGD Portal based on Semantic Web principles to support the deployment of Linked Open Government Data. This chapter[1] introduces the informatic challenges faced while developing the portal over the past two years and describes the design solutions employed by the portal's LOGD production infrastructure.

Since substantial human effort is needed to make raw datasets comprehensible, only a small proportion of the government data available has been published in an easily-reusable form using open principles. To accelerate the progress of opening more government data, new approaches are required to produce Linked Open Government Data as quickly as possible while allowing for incremental improvements developed by a broad community with diverse knowledge, skills, and objectives. We present a reference model (**?**) for the practical application of Linked Data techniques to integrate disparate and heterogeneous government data.

We introduce six stages of dataset integration (*Name*, *Retrieve*, *Adjust*, *Convert*, *Enhance*, and *Publish*). Five of these stages are designed to minimize human effort for incorporating a new dataset as Linked Data, while the remaining stage enables data modelers to add well-structured and well-connected descriptions to the initial representation. We describe enhancement types that a data modeler is most likely to use, along with a selection of more advanced enhancement types that elucidate the diversity of structural schemes employed by tabular government datasets. We use portions of the White House Visitor Access Records[2] as a running example.

[1] http://purl.org/twc/id/pub/lebo2011publishing

[2] http://purl.org/twc/pages/whitehouse-visitor-access-records

We describe an extension of the VoID Dataset class to establish a three level dataset hierarchy (*abstract*, *versioned*, and *layer*) that accounts for the RDF data resulting from incremental activities when accumulating new datasets, enhancing existing datasets, and handling new releases of those datasets already accumulated. Further, we highlight the correspondence between a dataset's URI and its role within the three-level VoID hierarchy. We then describe how this same correspondence is reused in our design to populate a SPARQL endpoint's named graphs.

After applying the five stages to create initial Linked Data from an OGD dataset and taking advantage of a sixth stage to enhance its representation, we describe how to handle an inevitable situation: a source organization releases a new version of a dataset we have already incorporated, published – and are using in applications. We use this situation to highlight several data organization challenges and how we solve them using a three-level namespace decomposition that simultaneously supports naming entities within and across datasets, establishing vocabularies that apply at different breadths, and performing bottom-up incremental integration of diverse datasets within and across source organizations – and among the Web of Data.

Finally, we address the paradoxical issue that a content view integrated from disparate sources obscures important answers about *how* it came to be, i.e., *who*, *where*, and *when* it was obtained or derived. This is increasingly important when the sources vary significantly in degrees of authority or reputability. We highlight the workflow's transparency by describing the context captured at each integration stage.

References

Ding, L.; Lebo, T.; Erickson, J. S.; DiFranzo, D.; Williams, G. T.; Li, X.; Michaelis, J.; Graves, A.; Zheng, J. G.; Shangguan, Z.; Flores, J.; McGuinness, D. L.; and Hendler, J. 2011. Twc logd: A portal for linked open government data ecosystems. *Web Semantics: Science, Services and Agents on the World Wide Web* In Press, Accepted Manuscript.

Lebo, T.; Erickson, J. S.; Ding, L.; Graves, A.; Williams, G. T.; DiFranzo, D.; Li, X.; Michaelis, J.; Zheng, J. G.; Flores, J.; Shangguan, Z.; McGuinness, D. L.; and Hendler, J. 2011. Producing and using linked open government data in the twc logd portal (to appear). In Wood, D., ed., *Linking Government Data*. New York, NY: Springer.

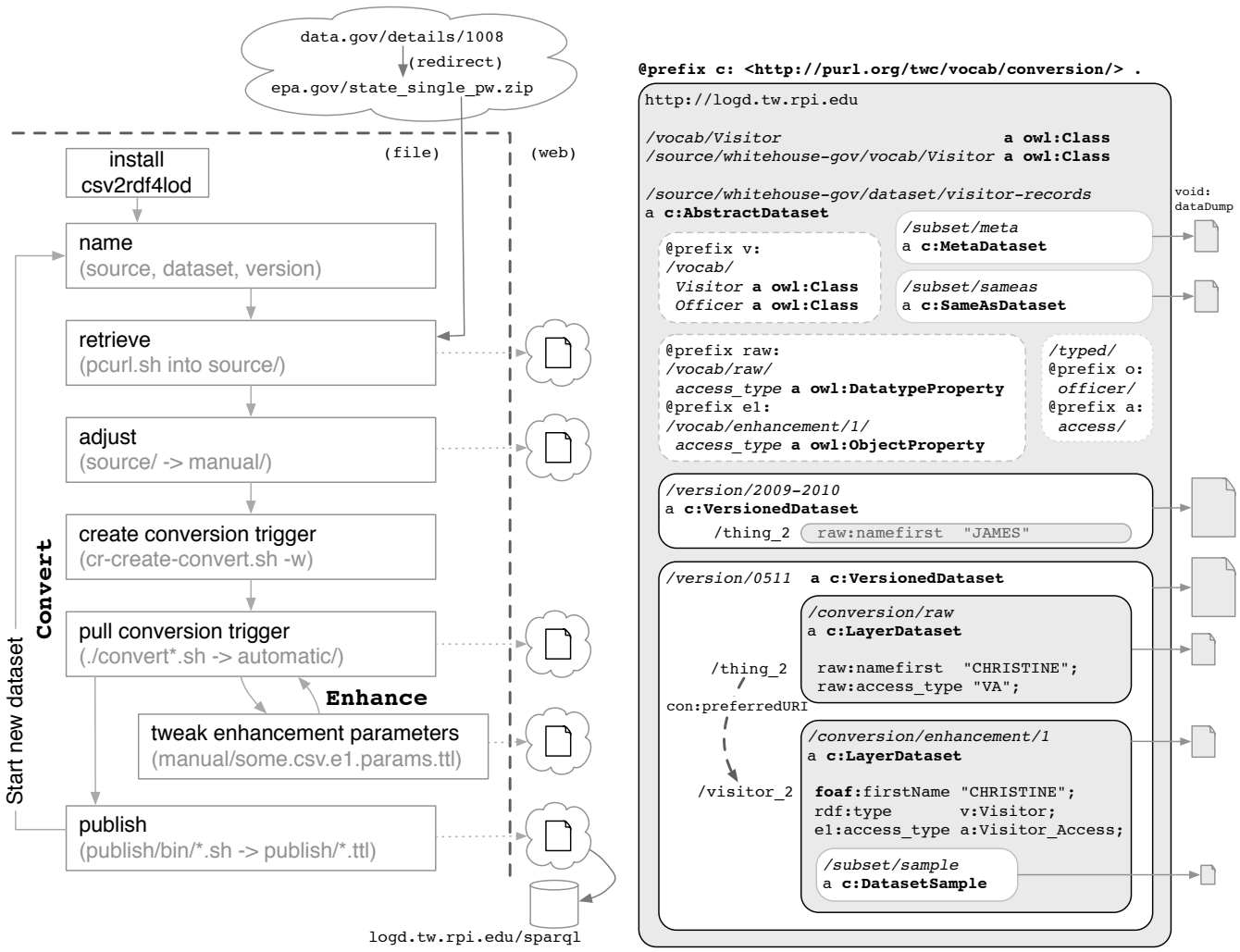

Figure 1: (**left**) The major stages of the LOGD production workflow are performed on the server, while the final and intermediate results are made available on the Web as dump files and through a SPARQL endpoint. Entities described in the SPARQL endpoint are available as resolvable Linked Data. Associations among the retrieved files and conversion results are encoded in RDF by provenance-aware utilities. Five of the six production stages require minimal human effort; the sixth enhancement stage can be performed as needed without disrupting applications built against earlier results. (**right**) Namespaces decompose according to *source*, *dataset*, and *version* identifiers assigned when retrieving data from other organizations; and *layer* identifiers assigned when interpreting it in different ways. Each step in the namespace decomposition corresponds to a void:Dataset URI that is a VoID superset of the datasets named within its namespace. URIs for entities, properties, and classes created from a dataset are named in namespaces corresponding to their breadth of applicability. Data integration is achieved incrementally by reinterpreting source data to use entities, properties, and classes from broader namespaces within this namespace decomposition or from existing vocabulary that is already used in the Semantic Web. For a grammar defining URIs, see (**?**).