Managing Versioned Web Resources in the File System

Leon Müller and ✉ Lars Gleim

Databases and Information Systems, RWTH Aachen University, Germany
firstname.lastname@rwth-aachen.de

Abstract. While the WebDAV standard provides a well-established read/write mechanism for Web resources, as well as version management through its Delta-V extension, the complexity of the underlying protocol limits its practical adoption. The W3C Linked Data Platform (LDP) more recently provides an alternative approach for simultaneous resource and semantic metadata management on the Web. In combination with the HTTP Memento protocol, it has recently been successfully employed in the context of interoperable data management. Inspired by file system interfaces for WebDAV, we present *factFUSE*, the – to our knowledge – first user-space application for the joint management of arbitrary computer files and semantic RDF data & metadata in the file system based on the LDP and HTTP Memento Web standards.

Keywords: Linked Data Platform · Semantic Data Management · FUSE · File System · Version Management · HTTP Memento Protocol

1 Introduction

As the volume and variety of data that is produced every year keep growing, the challenge of efficient data management is becoming increasingly relevant. Especially since the beginning of the Corona pandemic, the need for distributed data management systems has become more apparent than ever. While popular distributed cloud storage services like Dropbox or Google Drive are readily available, they are not based on open Web standards and lack important interoperability features. Inspired by these services and version control systems such as Git, we developed factFUSE: a data management system based upon the W3C Linked Data Platform (LDP) [6] standard enabling the unified management of RDF and binary resources through a simple hierarchical mapping of Web resources into the classical file system, while providing simple version control mechanisms based upon the HTTP Memento protocol.

2 Mapping LDP Resources into the File System

Besides standardized HTTP CRUD operations for reading and writing Web resources, the W3C LDP specification [6] provides primitives to hierarchically

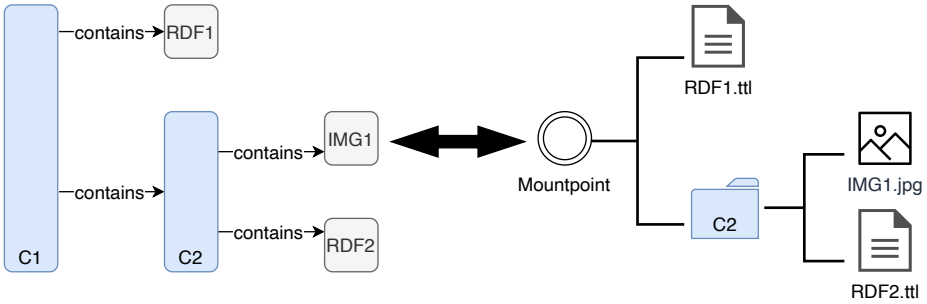

Fig. 1. Using the structural components of the Linked Data Platform Specifications, we can directly translate the containment relations in the LDP (left) to directory-child relations in the file system (right).

structure sets of RDF Linked Data resources, as well as arbitrary binary data, using containers that can contain RDF resources, binary resources and other containers. Based on the similarity of this organizational pattern with the traditional directory structure of hierarchical file systems, we define a straightforward mapping of LDP resources into the file system, as illustrated in Fig. 1. Containers are directly mapped to directories and other resources to corresponding file representations in the file system. While RDF resources are represented as text files containing a Turtle serialization of their triples, binary resources are displayed in their respective file format, as determined through their respective MIME-type.

Semantic Metadata. Each binary resource may further be augmented with RDF metadata which can contain semantic context information and metadata, stored in a dedicated metadata resource discoverable through a `describedBy` relation in the HTTP link header (implementing the LDP specification). Subsequently, this approach enables the flexible and extensible semantic enrichment of arbitrary binary resources and allows users to manage both RDF as well as arbitrary binary Web resources and their metadata inside the file system.

Persistent Identification. To allow for the persistent identification and referencing of individual resource revisions and keep a record of the resource revision history, we employ timestamped URLs according to the FactID scheme [3,1,2] to identify individual resource revisions and the HTTP Memento protocol [5], a standardized HTTP extension enabling time-based content negotiation to retrieve historical resource states, for version discovery and retrieval.

3 Implementation

factFUSE[1] is a NodeJS application implementing a custom user-space file system driver through JavaScript FUSE bindings[2]. The system is available as open-source

[1] https://git.rwth-aachen.de/i5/factdag/factfuse
[2] https://github.com/fuse-friends/fuse-native/

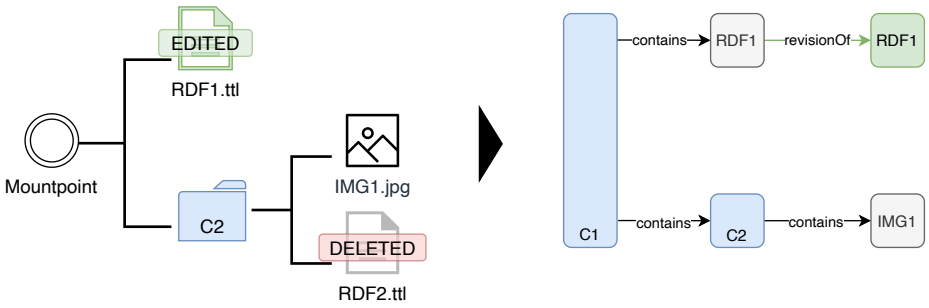

Fig. 2. Modifications to file representations are applied to the original LDP resource. Changes to resources are tracked by creating a list of revisions.

for both macOS and Linux environments. It communicates with LDP servers through the *factlib.js*[3] library, which is part of the FactStack data management system [3]. CRUD interactions with resources in the file system are directly translated to corresponding LDP HTTP calls. factFUSE further supports subscribing to server-provided change-notifications in Activity Streams 2.0 format via WebSockets (as implemented by the FactStack system) to be informed about resource updates on the LDP and to apply them to the local file representations in near real-time. This makes factFUSE a tool that can be used to interface with LDPs, manage data sets containing both RDF resources and binary resources, and integrate previously unconnected data into the Web of Linked Data.

Versioning. factFUSE transparently handles resource versioning using the Memento Protocol [5] and assigns a persistent Memento URL – a FactID – to each revision according to the FactDAG data interoperability layer model [2] and the FAIR data principles [7]. New revisions of a resource are connected to their predecessor through a W3C PROV-O [4] `revisionOf` relation, which is added to the RDF metadata of the new revision. Our custom user-space file system supports all regular CRUD file system interactions, such as writing, creating, removing or renaming files. All interactions are translated into according HTTP calls and so changes that were conducted in the file system representation are applied to the LDP side. Users may discover and download previous resource revisions through a file context menu. Therefore, factFUSE transparently discovers the version history of the underlying Web resource from its respective TimeMap provided by the LDP HTTP server in accordance with the HTTP Memento specification.

Example. As described in Section 2, factFUSE maps LDP Resources and Containers to files and directories in the local file system. A selected LDP Container is mapped to a virtual root directory – the mount point – within the local file system. Once mounted, LDP resources are exposed below this root directory. In an example scenario, as seen in Fig. 1, after connecting factFUSE to the LDP the user can interact with the user-space file system and modify

[3] https://git.rwth-aachen.de/i5/factdag/factlibjs

the resources with, or use them in, any desktop application. Fig. 2 shows the state of the LDP after editing one, and deleting another file in the factFUSE file system representation and the changes have been applied to the LDP side. The modified resource is stored as a new revision, with a `revisionOf` link to the original resource.

4 Conclusion

In this paper, we have presented factFUSE, a user-space application for the joint management of computer files and semantic data and metadata. It is based upon the LDP, PROV and HTTP Memento Web standards and offers a novel solution for distributed data management and version control in a familiar environment – the file system. Through the mapping of LDP resources into the local file system, factFUSE offers a first step towards bridging the gap between traditional file system-based data management and semantic web management solutions.

factFUSE is available[1] as open source software under the GNU AGPLv3 license to be evaluated by the community. The repository includes detailed installation and usage instructions, including screenshots of all the implemented features and interfaces in in-use scenarios, as well as pre-built binaries for macOS and the Ubuntu Linux distribution.

In future work, we plan to further extend upon the factFUSE concept in order to integrate flexible semantic metadata management into the traditional file system paradigm, ultimately working towards a tighter integration of Web resources, computer files and semantic knowledge graphs.

Acknowledgments Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC – 2023 Internet of Production – 390621612.

References

1. Gleim, L., Decker, S.: Timestamped URLs as Persistent Identifiers. MEPDaW@ISWC (2020)
2. Gleim, L., Pennekamp, J., Liebenberg, M., Buchsbaum, M., Niemietz, P., Knape, S., Epple, A., et al.: FactDAG: Formalizing Data Interoperability in an Internet of Production. IEEE Internet Things Journal **7**(4) (2020)
3. Gleim, L., Pennekamp, J., Tirpitz, L., Welten, S., Brillowski, F., Decker, S.: FactStack: Interoperable Data Management and Preservation for the Web and Industry 4.0. In: BTW 2021. Gesellschaft für Informatik, Bonn (2021), Preprint
4. Lebo, T., Sahoo, S., McGuinness, D.: PROV-O: The PROV Ontology. W3C Rec. (2013)
5. Van de Sompel, H., Nelson, M., Sanderson, R.: HTTP Framework for Time-Based Access to Resource States – Memento. IETF RFC 7089 (2013)
6. Speicher, S., Arwe, J., Malhotra, A.: Linked Data Platform 1.0. W3C Rec. (2015)
7. Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., et al.: The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data **3**, 160018 (2016)