Semantic Data Management in the File System through UI Extensions

✉ Lars Gleim[1], ✉ Leon Müller[1], Thomas Schemmer[2], Florian Brillowski[3], and Stefan Decker[1,4]

[1] Databases and Information Systems, RWTH Aachen University, Germany
[2] Chair of Communication Science, RWTH Aachen University, Germany
[3] Institute of Textile Technology, RWTH Aachen University, Germany
[4] Fraunhofer FIT, Sankt Augustin, Germany

Abstract. The importance of efficient data management keeps growing along with the increasing amount of data produced every year. Nevertheless, especially in the context of desktop applications, significant portions of data are collected, stored, and retrieved in traditional hierarchical file systems – without any explicit data management strategy nor adequate semantic metadata. Similarly, semantic data frequently remains inaccessible through traditional desktop applications. In practice, these shortcomings limit data findability, accessibility, interoperability, and reusability. In this work, we propose and investigate the usage of file system UI extensions to bridge the gap between the classical hierarchical management of computer files and semantic data management in RDF Graphs. We present factFUSE, a file system in user-space which transparently integrates computer files into a revisioned and provenance-linked RDF knowledge graph while maintaining compatibility with traditional file-oriented desktop applications. Inspired by the GIT commit system, we provide a user-friendly solution for semantic data management by extending standard file system functionality with extensible user interfaces for metadata management. Our quantitative evaluation in a user study shows that domain experts in textile engineering, without prior knowledge or experience in semantic data management, can successfully employ the system to manage semantic data and metadata, attesting excellent usability. We conclude that factFUSE is a suitable tool and a promising enabler for the sustainable adoption of user-friendly semantic data management in the industry.

Keywords: Semantic Data Management · File System · Provenance · Version Management · FAIR Data · Desktop Computing · Usability · User Study · FactDAG · FactStack · FUSE · Linked Data Platform

1 Introduction

As the amount of data produced continues to grow year by year, its management poses increasingly complex challenges, as the human capacity to reason about them does not scale accordingly [8].While there exist effective solutions for semantic data management [11,21], massive amounts of data are still managed through legacy approaches such as simple hierarchical file systems. Especially in the context of desktop applications, large quantities of data in the form of *computer files* are produced every day without proper semantic annotation or context embedding. We hypothesize that the low adoption of in many ways superior semantic data management solutions is partially due to the lack of tooling, reducing the cognitive load and overhead of semantic data management, as well as the lack of integration between such solutions and day-to-day computer file handling, especially in direct comparison with file system interfaces. Inspired by the global success of the Git [23] system as unified data management and version mismanagement system in software development (across vendors, programming languages, operating systems, and geographical borders), we explore how to integrate semantic data management directly and transparently into the file system. Therefore, we employ suitable *user interface* (UI) extensions and directly integrate these into the operating system's file system explorer.

Design Goals. We strive to fulfill the following design goals with our system:

- G1 *Low adoption barrier* for users that are already familiar with hierarchical data management in the file system.
- G2 *Interoperability* with existing desktop applications, which provide almost universal support for interfacing with local files and file systems, to enable the semantic management of arbitrary computer files.
- G3 *Transparent adoption* of the fundamental *FAIR principles* [31] of scientific data management.
- G4 Implementation of best practices for data management throughout the data life-cycle, notably including resource *versioning*, revision *immutability* and *unique referenceability* [15].
- G5 *Semiautomatic semantic data annotation*, combining automatic metadata collection with contextual user prompts if applicable.
- G6 Configurability, customization and *extensibility* of the collected semantics.

Contributions. (i) our vision of a unified semantic data management system integrating RDF, arbitrary data types, and computer files in a fundamentally provenance-linked knowledge graph, (ii) an extensible concept for the semantic management of computer files in the file system through UI extensions, (iii) an open-source implementation of this concept for both Linux and macOS, (iv) a quantitative user study evaluating this implementation based on a practical data management use case in the manufacturing domain, and (v) a qualitative discussion of the strengths, weaknesses, opportunities and threats of this model fundamentally based on a Think-Aloud Study [19].

Paper Structure. The remainder of this paper is structured as follows: Section 2 provides an overview of foundations and related approaches for semantic and hierarchical data management. Section 3 then introduces the concept of semantic data management in the file system through contextual UI extensions. Section 4 presents our open-source implementation of this model for both Linux and Mac OS, called *factFUSE*, and Section 5 discusses the findings of our quantitative and qualitative evaluation of this implementation. Conclusively, we summarize our contributions, findings and directions for future work in Section 6.

2 Related Work

Much of the data that is generated in production environments and systems focused on desktop computing, is still hierarchically stored and managed in the form of computer files. While being familiar to all kinds of users and a predominant data organization pattern since the inception of personal computing, file systems are traditionally limited to conveying only hierarchical relationships between resources. With the rise of distributed file systems and cloud storage systems, extended forms of data management, such as version-control and distributed authoritative attribution, have found their way into mainstream computer file management. Systems like Dropbox [9] enable users to store and share their hierarchical data in a distributed manner and deploy versioning mechanisms that attach useful metadata information to the resources. Nevertheless, more advanced forms of semantic metadata management and solutions capturing relations between different resources have seen rather limited adoption compared to the classic computer-file based data management.

Semantic data management (SDM) refers to approaches that focus on maintaining and using data whose meaning is made explicit in the form of meta-data. While the challenge of data and knowledge management is a well-known issue, today's notion of SDM – inspired by Vannevar Bush's vision of *associative indexing* and *trails of references* [8], as well as J. C. R. Licklider's 1965 models for networking information and services [22] – has been particularly successful in the scientific research data management domain. [6] In conjunction with the proposal of the FAIR principles of scientific data management and stewardship [31] – promoting data findability, accessibility, interoperability and reusability – SDM approaches have recently seen increased adoption to other domains, such as manufacturing technology [20,15].

Today, the arguably most prominent approach to represent and manage semantic data and metadata is based on the *Resource Description Framework* (RDF) [32] and a family of related standards proposed by the *World Wide Web Consortium* (W3C), e.g., SPARQL, RDFS, and OWL. RDF is a directed graph-based data model, employing *Uniform Resource Identifiers* (URIs) to uniquely identify resources and concepts as well as the foundation of *Linked Data* (LD). These standards and their various software implementations are considered sufficiently mature for practical use and employed infrastructure management and cultural heritage domain (e.g. in criminal intelligence). [17]

An important type of semantic metadata for data management is the concept of data provenance, capturing the origins, processes, influences, agents, and history of changes that led to its current state. Antoniou et al. [1] summarize, that it is particularly relevant when integrating data from heterogeneous sources, especially when under diverse ownership, in order to support trust mechanisms, policies for privacy and rights management, as well to include information on processing or reasoning operations carried out on the data. In the scope of our research, provenance plays an important role since its a form of metadata that can be applied to arbitrary data types, including classic computer-files. Tracking and managing provenance information for data stored in file systems allows us to apply semantic data management to files and elevate previously unstructured sets of data to semantic graphs.

Some approaches have been proposed that aim to publish sets of file system resources as Linked Data. Schandl and Popitsch [28] identify the lack of metadata support in the classical file system as the main reason for the lack of integration of file system knowledge into semantic webs. They present TripFS, an approach that expresses existing metadata, e.g., access and modification times, size or permissions, in RDF. Files are assigned HTTP URIs and can be linked to external web-sources, but not to other files or directories. Solutions like F2R[18] and VDB-FilePub [29] also extract existing file metadata and assign HTTP URIs to files for identification. The metadata can be enriched with user-defined RDF triples linking either to labels or external sources. LinkZoo [24] enables users to collect and share resources collaboratively and enrich them with attributes from external ontologies. A folder-based explorer is used to display resources that can be queried based on their annotations.

Semantic file system solutions extend the functionalities of a classical file system with the ability to add semantic annotation, tags or links to resources. Prior research by Gifford et al. [13] recognized the advantage of semantically annotated resources to provide alternative access to files other than hierarchical browsing. Eck and Schaefer [10] present a semantic file system in which users can tag their files with metadata and provide tag-based browsing and a query mechanism in an explorer interface. Linking files to other files through relationships is not possible. Similarly, TagFS [5] implements the tagging of arbitrary metadata information to files and allows users to query the file system based on tags. Direct relationships between files can not be made.

In summary, most approaches towards the semantic enrichment of file systems and semantic data management for computer-files extract existing file metadata or enable users to manually and semantically annotate files. The resulting semantic graphs on top of the file system can then be used to query the data set as a way of alternative file access. Generally, the above-mentioned approaches do not integrate computer-files into existing semantic graphs, but rather create a new graph around the resources.

3 Concept

In the following, we propose to map Linked Data into the file system and, vice versa, file system resources to Linked Data representations. We enable the management of semantic metadata through user interface extensions. The resulting extended file system acts as an interface for Linked Data sets and facilitates basic semantic data management in the form of file modification. Further semantic management aspects, such as metadata collection and visualization, are enabled through the use of corresponding specialized user interface extensions.

Mapping Linked Data into the File System. To provide the basic mapping between graph-structured RDF data and a tree-structured file system, we rely upon the hierarchical organization of resources provided through the

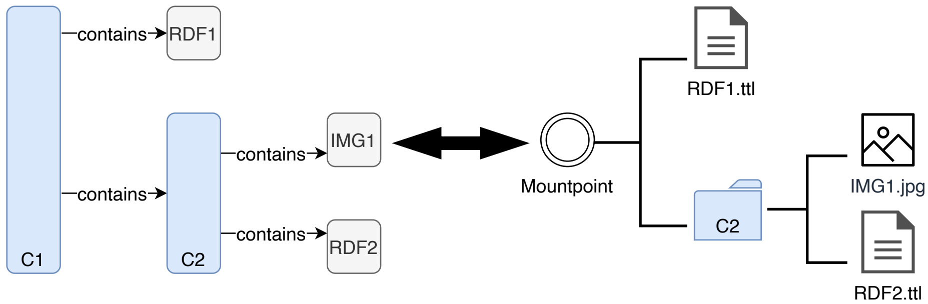

Fig. 1. Using the structural components of the Linked Data Platform Specifications, we can directly translate the containment relations in the LDP (left) to directory-child relations in the file system (right).

containment-relationship of the W3C *Linked Data Platform* (LDP) [30]. Besides providing resource *containers*, the standard further provides convenient access and management of both *RDF* and *Non-RDF* resources using CRUD operations through a standardized REST API. Additionally, it provides a mechanism to maintain dedicated metadata records for arbitrary Web resources, connected through the HTTP `rel="describedby" Link` header. A detailed introduction to the LDP can be found in [30].

Through the LDP, Linked Data and Web resources can be managed similarly to regular files in a local file system while enabling the augmentation of arbitrary resources with semantic metadata. Building upon this foundation, we employ the following bijective mapping to align LDP and file system concepts:

– LDP *containers* are file system *folders*,
– *RDF* resources are *text files* with their respective RDF Turtle serialization,
– *Non-RDF* resources are corresponding *files* with a type and file extension suitable for their respective MIME-type [12].

An illustration of this mapping can be found in Fig. 1. Additionally, RDF *metadata* (part of the RDF resource or the LDP metadata resource of Non-RDF resources) is represented, visualized and managed through corresponding UI extensions. These include context-specific prompts, UI elements, and visualizations directly integrated into the file system explorer of the operating system.

In the context of this work, we focus on the management of provenance information, as this type of metadata allows for the interlinking of diverse data types, even across organizational boundaries, capturing processes, semantic contexts, and influences throughout its formation history [15], as well as the management and explorations of resource versions. To embed metadata management into the traditional workflow of managing files in the file system, we employ (i) a *Commit System* inspired by the distributed version control system GIT [23], which may expose user interfaces for metadata collection during the process of persisting resource modifications, and (ii) extensible *Metadata Interfaces* that may be opened directly from the file system explorer (either through context menus

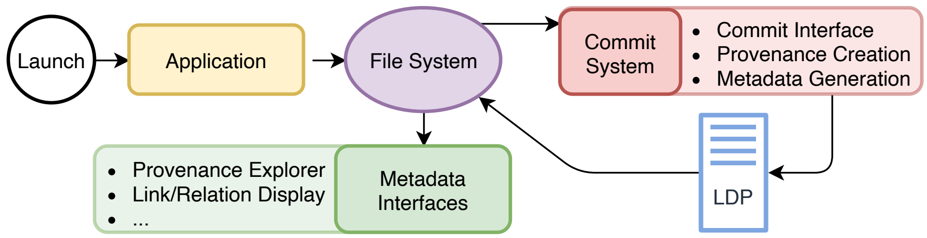

Fig. 2. The conceptual architecture of factFUSE, integrating semantic data management into the file system by providing metadata collection and management through UI extensions.

or by adding buttons to the operating system's file system explorer itself) to manage and display metadata of the selected resources, as illustrated in Fig. 2).

Building upon this conceptual architecture, we present our open-source implementation factFUSE as an extensible bridge for the integration of traditional, hierarchical computer file organization and semantic data management.

4 Implementation

factFUSE is a NodeJS application, providing a custom user-space file system through JavaScript FUSE bindings[5]. It communicates with LDP servers through the *factlib.js*[6] library, which is part of the FactStack data management system [14]. The library automatically handles resource versioning and assigns a persistent URI – called FactID – to each revision according to the FactDAG data interoperability layer model [15] and the FAIR data principles [31]. Additionally, it provides helper functions for provenance collection, management and preservation according to W3C PROV standard [16,4]. The PROV model expresses data provenance through *entities*, *activities* and *agents*. Entities are used and generated by activities with responsible agents. Entities can be revisions of one another, with each revision being the result of an activity. Tracking provenance information for data gives insight into the history of resources and can help to assess quality and trustworthiness or even be used to trace errors back to their origin.

Managing Linked Data in the File System. Based on the LDP to file system mapping described in Section 3, 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 within the local file system. Once mounted, LDP resources are exposed below this root directory. Subsequently, they can be used just like any other locally stored resources and can therefore

[5] `https://github.com/fuse-friends/fuse-native/`
[6] `https://git.rwth-aachen.de/i5/factdag/factlibjs`

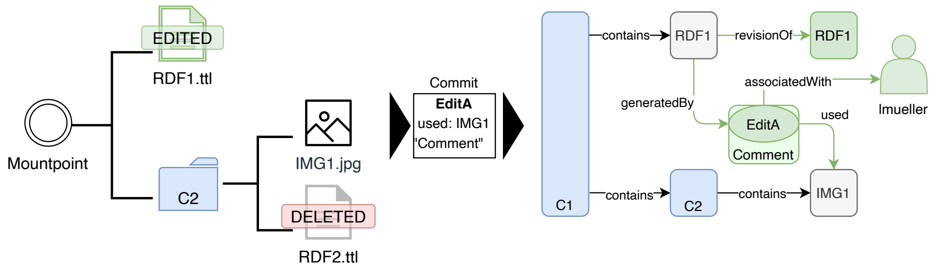

Fig. 3. Modifications in the local file system are enriched with semantic metadata collected through UI extensions. Data and semantic metadata are then jointly committed to the LDP server.

be easily integrated and processed with traditional desktop applications or processing workflows. If the factlib.js library detects a change to a relevant remote LDP resource, the local file system representation is automatically updated accordingly. Changes made to resources in the file system are tracked in an internal cache and can be committed to the LDP via the Commit System. All committed changes generate provenance information.

Generating Provenance Information. A Commit carries information on the time and content of changes, a title, its author, a message, and possibly resources that were involved. Therefore, a Commit can be directly mapped to an activity in the PROV model with the current user as the responsible agent. To enable manually issued Commits and custom metadata collection, we designed a GUI that lists all detected changes. It offers UI elements to modify the aspects of provenance and other metadata that can be included in a Commit. Fig. 3 shows an example, in which changes made to resources in the file system representation are committed to the LDP. The Commit includes automatically collected provenance information, as well as custom metadata, specified by the user in the CommitGUI. The LDP representation on the right shows the applied changes and the generated provenance information based on the users' input in the GUI. The CommitGUI further provides an important extension point for prospective additional metadata collection, possibly depending on the type and context of the modified resources. Depending on the use case, manually committing changes each time modifications to resources occur, may not be practical. Manual committing and customization of provenance information can greatly improve the level of detail in metadata but is time-consuming and work-intensive. Thus, factFUSE also implements an automatic synchronization mode. When enabled, local changes are automatically committed, but manual metadata collection through prompts is disabled. In the automatic synchronization mode, only automatically collected metadata (such as revision information) and preconfigured defaults (such as the identity of the agent responsible for the update) are included.

Displaying and Exploring Metadata. In order to enable the semantic exploration and management of existing resources, a set of user interfaces has

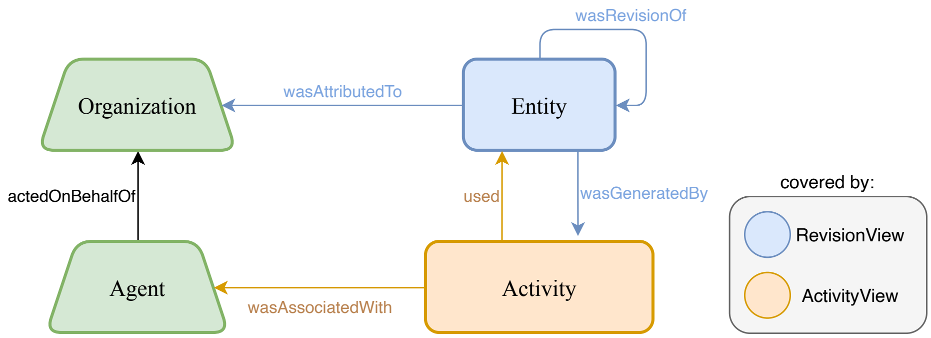

Fig. 4. Provenance graph exploration with our User Interfaces.

been designed and implemented extending the file system's functionality. The interfaces are accessible inside the file system through the right-click context menu or buttons in close proximity to selected files. For the management of provenance information, we focus on two interfaces: one for information on entities and one for information on activities in the PROV model, which allow for exploration along the edges of the provenance graph (see Fig. 4): the *RevisionView* and the *ActivityView*.

RevisionView: The *RevisionView* (Fig. 5) displays a history of all existing revisions of a resource, the FactIDs, and each revisions generating activity. Additionally, UI elements to download or restore revisions are provided. Inspecting a revisions generating activity allows further inspection along the edges of the PROV graph, by opening the *ActivityView*.

ActivityView: An activity's *ActivityView* (Fig. 5) holds information on its responsible Agent, the time of execution, and a list of resources that have been used during the activity. Additionally, if the Activity corresponds to a Commit issued through the factFUSE system, the interface may contain the commit message. The used resources can again be further inspected by following a link to their *RevisionView* interface.

This approach of providing context-dependent UI extensions to manage and visualize metadata, in alignment with individual ontologies and best practices of specific application domains, allows for the progressive and configurable adoption of semantic data management to the traditional file system. While the two described interfaces only provide access to a limited subset of the potentially much more extensive semantic context of the inspected resource, they already provide a tangible benefit for users of traditional file systems, as validated by our evaluation in Section 5.

The factFUSE Application. factFUSE is bundled, distributed, and installed as a traditional desktop application. This provides a familiar way for users to launch the system, configure settings, and allows us to make most of the features accessible from central control elements. While initialization and configuration are handled in a designated interface, most of the systems func-

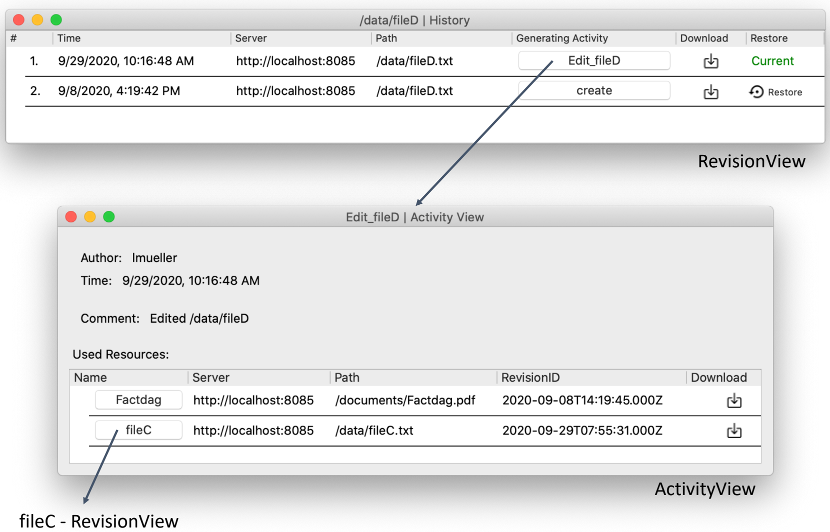

Fig. 5. The *RevisionView* interface shows the revision history of a resource and lets users interact with it. Inspecting a revisions generating activity opens its *ActivityView*. The *ActivityView* provides information on a specific activity and its used resources.

tionalities and metadata management interfaces can be accessed from the system tray menu of the respective operating system.

factFUSE is available as open source software[7] for both macOS and Linux environments, enabling the management of Semantic Data and extending the widely used Finder and GNOME file systems with metadata management interfaces.

5 Evaluation

To validate the user-friendliness of the factFUSE system, a user study was conducted evaluating the management and exploration of semantic provenance metadata and resource versions. To determine the difficulty of specific real use cases as well as the overall usability, the system's evaluation was focused on task and on test level.

Setup. The evaluation was carried out with a group of 10 researchers and engineers at the Institut für Textiltechnik (ITA), RWTH Aachen University. Participants were asked to complete a set of six tasks that each represented a core functionality as well as real-world use case of the system. As an introduction, the participants were given a short tutorial on the basic concepts of the PROV model, as well as the factFUSE application itself. The user study was conducted

[7] https://git.rwth-aachen.de/i5/factdag/factfuse

according to the protocol of a Think-Aloud Study [19], which encourages participants to vocalize their thought process throughout the test, to gain more detailed insight into possible difficulties. The subjects were asked to conduct the following 6 tasks using the v0.1.5 release[8] of factFUSE on macOS:

Task 1. Focused on the system configuration. Participants were asked to launch the system, configure a connection to an LDP server and mount the file system to a local directory.

Task 2. Introduced the provenance exploration interfaces. The tasked specified to retrieve information on a resources revision count and the responsible author of the newest revision.

Task 3. Involved the modification of files in automatic synchronisation mode. It aimed to show the participants that the file system could be used to interact with LDP resources as with regular files, while the system automatically synchronized the changes and tracked provenance.

Task 4. Laid out a scenario where a resource contained wrong information. The participants had to use the UI extensions to identify the responsible author of the generating activity, all used resources in that activity and restore the resource to the last correct revision.

Task 5. Introduced the concept of manual committing and provenance configuration. Resources were modified in manual commit mode and participants provided additional customized provenance information on their changes before committing them.

Task 6. Asked the participants to find and inspect the activity in the UI extensions, that was generated by their custom Commit in Task 5.

Two different standardized usability scales were used in the evaluation. A Single Ease Question (SEQ) [26] asking participants how difficult a task was on a scale from 1 (very difficult) to 7 (very easy) was presented after each of the 6 tasks. Upon completion of the test, the System Usability Scale (SUS, a 10 item questionnaire) [7] was handed to the participants. The SUS is a well-established method to measure the overall usability of a system [3].

Quantitative Results. The SEQ results, as seen in Fig. 6, show the degree of perceived difficulty for each task. With the exception of Task 2, all tasks yielded average results above 6. Scores above 5 generally indicate satisfying usability of the system in that specific use case [25]. The problems that some participants encountered in Task 2 were due to unfamiliarity with the PROV concepts, especially the relationship between revisions and their generating activities. In particular, retrieving information on the responsible agent from the activity rather than from the entity itself was not intuitive for most participants. This has become evident during the Think-Aloud Study in which the participants vocalized their confusion. Task 4 again asked the participants to interact with the interfaces and PROV relations in a similar way to Task 2. This time, the relationship between revisions and activities was clear to all participants and the task did not receive any problematic SEQ scores. Using the system, all

[8] `https://git.rwth-aachen.de/i5/factdag/factfuse/-/releases#v0.1.5`

participants had learned previously unknown concepts in Task 2 and were able to apply their knowledge in Task 4, within their first use of the system. Based on the high average results of the tasks, we see that factFUSE can indeed enable domain experts to manage provenance information without prior learning or major difficulties.

After completion of the tasks, all participants were asked to fill out the ten items SUS questionnaire. The answers were analyzed and converted into the according SUS scores that can range from 0 to 100 points. Research from Sauro and Lewis [27] has shown that the average SUS score of a system is about 68 points. As seen in Fig. 6, the factFUSE system received an average score of 83.25, a well above average result. Based on a grading scale proposed in [27], we can interpret our SUS result as a Grade A in usability and consider the factFUSE application's usability to be in the percentile range between 90 and 95.

Both the task and test level results yielded above average results in a participant group of domain experts that had not used the system prior to the evaluation. Additionally, the participants' interactions with the system showed promising learning effects and improved understanding of the provenance relations, within the first usage of the system.

Qualitative Discussion. The user study and its results have shown that semantic data management through UI extensions in the file system can enable user-friendly management and exploration of Linked Data. All participants were able to interact and modify resources stored on an LDP, view and interpret provenance information and create new provenance information for their modifications, within their first use of the system. Although most of the participants had no experience with any metadata management tools prior to the study, both task and test level evaluations – conducted using standardized scales – yielded "excellent" [2] results that confirmed the good usability and user-friendliness of factFUSE. We thus deem design goal *Low adoption barrier* (`G1`) to be fulfilled by our system. Though some participants had initial difficulties with the relationship between entities and their generating activities (in Task 2), they quickly

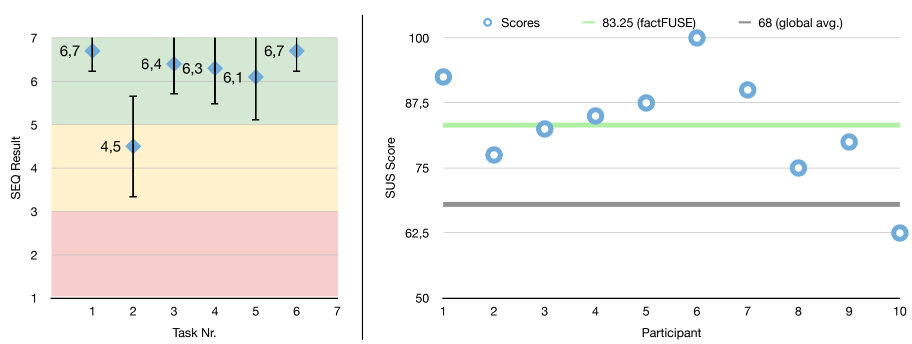

Fig. 6. The average SEQ results (left) and SUS results and average score (right), from the conducted user study.

grasped the concept and were able to apply the gained knowledge in Task 4. Therefore we can assume a promising learning effect for domain experts in the interaction with both the system and the concept of provenance.

During the Think-Aloud study and discussion after the evaluation, all users expressed their satisfaction with being able to browse and interact with the resources in the file system. Nine out of the ten participants stated that they generally preferred factFUSE in comparison to their existing tools employed for data management. 3 of the participants recognized the advantage of being able to import file system resources into most available desktop applications and legacy systems that rely on locally stored resources, fulfilling the design goal of *Interoperability* (G2). The most prominently mentioned advantage of the file system interface is the almost axiomatic familiarity of users with file system interfaces.

By building upon the FactStack data management system (as outlined in Section 4), factFUSE further benefits from its realization of the FAIR principles of scientific data management, as well as its adoption of further best practices, such as resource versioning, revision immutability and unique referenceability [14]. Thus, factFUSE also fulfills design goals for *Transparent adoption of FAIR principles* (G3) and (G4).

By using the custom FUSE file system as an interface for Linked Data Platforms, we can include new sets of files and resources into existing RDF graphs along with metadata information, by simply copying them into the mounted file system. As specified by the goals of *Semiautomatic semantic data annotation* (G5) and *extensibility of semantics* (G6), the system automatically collects essential provenance information that can be manually extended and configured by the user. This provides an intuitive way to connect semantic graphs to previously disconnected/unstructured sets of data. For binary resources (e.g. documents or images) that are typically difficult to include in semantic graphs, we automatically generate semantic resource representations. All generated metadata is then linked to the semantic representation, allowing us to seamlessly integrate all kinds of binary resources as well as mixed sets of semantic and binary data into RDF graphs.

While the hierarchical nature of classical file systems provides a straightforward primitive to map and display hierarchical LDP relations (see Fig. 1), other semantic relations between resources have largely remained inaccessible to users without the proposed UI extensions. As factFUSE relies on the hierarchical metadata provided by the LDP, mapping LD resources without corresponding LDP embedding remains outside of the scope of factFUSE system. At the current state of the implementation, factFUSE only supports a set of provenance relations to be managed and collected for resources. As of yet, no generic UIs for RDF metadata have been added. Due to these limitations of factFUSE to provenance metadata, the conducted evaluation is limited to the management of provenance information as well. We assume similarly satisfying results for the adoption of more generic metadata management, through the implementation of additional UI extensions in the future.

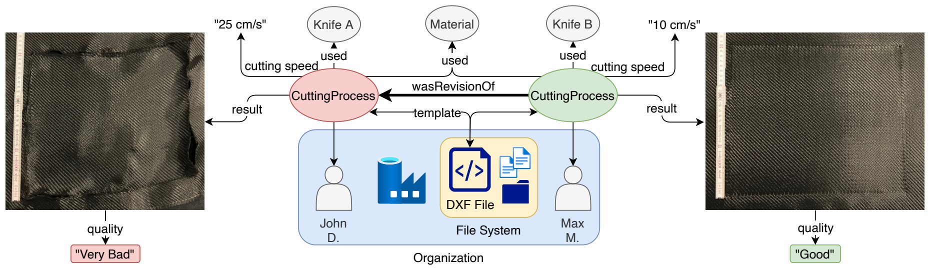

Fig. 7. Multiple executions (revisions) of the same Process (activity) with different configuration metadata and results. Namespaces have been disregarded for conciseness.

Practical Value for a Use Case in Manufacturing. Based on the conducted quantitative and qualitative evaluation, we provide a comprehensible real-world use-case within the composites industry, discerning the practical value of factFUSE for the industry. The manufacturing of carbon-fiber-reinforced plastics requires the cutting of carbon-fiber-textiles which is often done by designing a cutting template, which is then processed by a CNC cutting machine. These cutting templates are typically created in the form of DXF files, effectively a type of 2D-line image, which is maintained in the traditional file system. However, when providing this DXF File to a CNC cutter, the operator also needs to provide a number of machine parameters and other metadata are usually based on the experience of the machine operator and cannot be recorded within the DXF file. Workpiece quality may therefore vary significantly between workers and process parameters may have to be redetermined through trial and error, although similar parts were successfully produced in the past.

Fig. 7 shows the result of an identical template being cut twice by an automated CNC cutting machine. The only differences between both documented results were varying machine parameters. Due to their major influence on the cutting result, documenting parameter settings including cutting-speed, cutting-acceleration or the used knife-type is of utmost importance. By not maintaining these parameters in conjunction with the workpiece design, errors, unnecessary waste and bad quality results (e.g. Fig. 7, left side) are more likely to occur and to be repeated. Collecting and exposing such process metadata in digital semantic form and making it available through file system extensions could therefore significantly improve productivity and reduce manual errors in production.

Using semantic data management in the file system, as enabled through the usage of factFUSE, the cutting template could be linked to several useful metadata information, e.g. used material and machine parameters. The resulting cut could then be photographed, its quality be assessed, and then an image also be added to the cutting template and related through semantic metadata. The resulting metadata graph would then be made accessible in the form of UI Extensions within the file system. Different iterations of the process with individual results and configuration parameters could further be stored as revisions of the same process. This would allow other users to predict the quality of the cut based on previous results and their configurations. Repeated errors could be prevented and suitable machine parameter settings could be made available in direct proximity to the used template file.

With the example of provenance management, we have shown that forms of semantic data management can indeed be implemented through user interfaces that extend the classical file system functionality. Our evaluation has shown that the factFUSE system provides a user-friendly first step towards the application of metadata management in industrial applications that enables domain experts to utilize it.

6 Conclusion

In this paper, we presented a solution for semantic data management in the file system through the use of UI extensions. We introduced the embedment of semantic data, stored on LDPs, into the classical file system, supported by UI extensions for the exploration and management of semantic metadata. We presented factFUSE, our open-source implementation of this system, enabling the user-friendly joint management of RDF and Non-RDF data with semantic metadata in the file system through UI extensions. Our approach supports both the management of existing RDF graph resources as well as the elevation of previously unstructured data sets into revisioned and provenance-linked graphs. The system was evaluated in a quantitative user study based on a practical manufacturing use case, yielding excellent usability results and showing a quick adoption of principles by domain experts.

With the example of provenance management, we have shown that semantic data management can be brought to the file system, in a way that enables users without prior experience to utilize it. Thus, factFUSE is a promising step towards the application of semantic data management in industry and away from unstructured production data. Based on the successful evaluation of the solution in the context of provenance management, we propose the adoption of the same concept for the management of other types of semantic RDF metadata.

Our evaluation has shown that the file system can provide a familiar environment for domain experts to adapt to concepts of semantic data management. While other forms of data representation, e.g. graph-based visualizations, may provide richer (i.e., more expressive) representations of available semantic information, they are also more complicated for inexperienced users to adopt and provide less compatibility with existing desktop applications and legacy systems. Therefore, extending traditional hierarchically structured data storage and retrieval with functionality for semantic data management, supported by adequate user interfaces, could bridge the gap between traditional file-based data management and semantic data management for all kinds of resources and processes.

In the context of future work, we plan to continue the development of factFUSE and to implement domain specific interface extensions for the management and visualization of semantic metadata in the context of the factFUSE approach. Furthermore, we plan to implement and evaluate the system on a larger scale in conjunction with practical manufacturing processes at the ITA institute and its industrial partners.

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

References

1. Antoniou, G., Corcho, O., Aberer, K., Simperl, E., Studer, R.: Semantic Data Management (Dagstuhl Seminar 12171) **2**(4), 39–65 (2012)
2. Bangor, A., Kortum, P., Miller, J.: Determining what individual sus scores mean: Adding an adjective rating scale. Journal of usability studies **4**(3), 114–123 (2009)
3. Bangor, A., Kortum, P.T., Miller, J.T.: An empirical evaluation of the system usability scale. Intl. Journal of Human–Computer Interaction **24**(6), 574–594 (2008)
4. Belhajjame, K., B'Far, R., Cheney, J., Coppens, S., Cresswell, S., Gil, Y., Groth, P., Klyne, G., Lebo, T., McCusker, J., et al.: Prov-dm: The prov data model. W3C Recommendation (2013)
5. Bloehdorn, S., Görlitz, O., Schenk, S., Völkel, M., Karlsruhe, F.: Tagfs-tag semantics for hierarchical file systems. In: Proceedings of the 6th International Conference on Knowledge Management (I-KNOW 06), Graz, Austria. vol. 8, pp. 6–8 (2006)
6. Borgman, C.L.: Scholarship in the digital age: Information, infrastructure, and the Internet. MIT press (2010)
7. Brooke, J.: Sus: A quick and dirty usability scale. Usability evaluation in industry p. 189 (1996)
8. Bush, V., et al.: As We May Think. The Atlantic Monthly **176**(1), 101–108 (1945)
9. Drago, I., Mellia, M., M. Munafo, M., Sperotto, A., Sadre, R., Pras, A.: Inside dropbox: understanding personal cloud storage services. In: Proceedings of the 2012 Internet Measurement Conference. pp. 481–494 (2012)
10. Eck, O., Schaefer, D.: A semantic file system for integrated product data management. Advanced Engineering Informatics **25**(2), 177–184 (2011)
11. European Organization For Nuclear Research, OpenAIRE: Zenodo. `https://www.zenodo.org/` (2013)
12. Freed, N., Borenstein, D.N.S.: Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. RFC 2046 (Nov 1996), `https://rfc-editor.org/rfc/rfc2046.txt`
13. Gifford, D.K., Jouvelot, P., Sheldon, M.A., O'Toole Jr, J.W.: Semantic file systems. ACM SIGOPS operating systems review **25**(5), 16–25 (1991)
14. Gleim, L.: FactStack: Interoperable Data Management and Preservation for the Web and Industry 4.0. In: RDA 16th Plenary Meet. - Poster Sess. (2020)
15. Gleim, L., Pennekamp, J., Liebenberg, M., Buchsbaum, M., Niemietz, P., Knape, S., Epple, A., Storms, S., Trauth, D., Bergs, T., Brecher, C., Decker, S., Lakemeyer, G., Wehrle, K.: FactDAG: Formalizing Data Interoperability in an Internet of Production. IEEE Internet Things J. **7**(4) (2020)

16. Gleim, L., Tirpitz, L., Pennekamp, J., Decker, S.: Expressing FactDAG Provenance with PROV-O. In: MEPDaW @ ISWC (2020)
17. Hartig, O., Curé, O.: Semantic Data Management in Practice. 26th Int. World Wide Web Conf. 2017, WWW 2017 Companion pp. 901–904 (2019)
18. He, S., Li, J., Shen, Z.: F2r: Publishing file systems as linked data. In: 2013 10th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). pp. 767–772. IEEE (2013)
19. Jääskeläinen, R.: Think-aloud protocol. Handbook of translation studies **1**, 371–374 (2010)
20. Kasrin, N., Qureshi, M., Steuer, S., Nicklas, D.: Semantic Data Management for Experimental Manufacturing Technologies. Datenbank-Spektrum **18**(1), 27–37 (2018)
21. King, G.: An Introduction to the Dataverse Network as an Infrastructure for Data Sharing. Sociol. Methods Res. **36**(2) (2007)
22. Licklider, J.C.R., Clapp, V.W.: Libraries of the Future, vol. 2. MiT Press Cambridge, MA (1965)
23. Loeliger, J., McCullough, M.: Version Control with Git: Powerful tools and techniques for collaborative software development. O'Reilly Media (2012)
24. Meimaris, M., Alexiou, G., Papastefanatos, G.: Linkzoo: A linked data platform for collaborative management of heterogeneous resources. In: European Semantic Web Conference. pp. 407–412. Springer (2014)
25. Sauro, J.: 10 things to know about the single ease question (seq) (October 2012), `https://measuringu.com/seq10/`, [Online; posted 30-October-2012]
26. Sauro, J., Dumas, J.S.: Comparison of three one-question, post-task usability questionnaires. In: Proceedings of the SIGCHI conference on human factors in computing systems. pp. 1599–1608 (2009)
27. Sauro, J., Lewis, J.R.: Quantifying the user experience: Practical statistics for user research. Morgan Kaufmann (2016)
28. Schandl, B., Popitsch, N.: Lifting file systems into the linked data cloud with tripfs (2010)
29. Shen, Z., Hou, Y., Li, J.: Publishing distributed files as linked data. In: 2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). vol. 3, pp. 1694–1698. IEEE (2011)
30. Speicher, S., Arwe, J., Malhotra, A.: Linked Data Platform 1.0. W3C Rec. (2015)
31. 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. Sci. Data **3**, 160018 (2016)
32. Wood, D., Lanthaler, M., Cyganiak, R.: RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation, W3C (feb 2014)