SemDAV: A File Exchange Protocol for the Semantic Desktop

Bernhard Schandl

University of Vienna
Department of Distributed and Multimedia Systems
bernhard.schandl@univie.ac.at

Abstract. The traditional file system is an integral part of how users interact with their desktop computers: To a great extent, user content (e.g. business documents, holiday pictures, project plans, etc.) is stored in a more or less unstructured manner. In the file system, this content is organized only using hierarchical directories and file names, which lack advanced expressivity in comparison to ontology-based classification schemes. As a result, traditional file systems do not provide sufficient means for organizing and annotating such content, especially when multiple users access the same file inventory. In this position paper, we introduce the SemDAV project[1], in which we aim to research technologies that are able to semantically enrich storage of such unstructured content. We give an outline of the SemDAV project, explain the motivation behind it, and discuss how SemDAV can contribute to the Semantic Desktop movement.

1 Introduction

As we can observe from everyday experience, personal desktop computers must provide quick, straightforward and traceable means to store user data, e.g. business documents or holiday pictures. We denote such data as *unstructured content*, since its inner structure is usually not well-defined, but created according to the current situation and user needs. Since years, the traditional hierarchical file system serves as such a storage facility: It is possible to store every kind of data with only very limited organizational procedures. After editing a document, it is saved by entering a few letters as file name, and it is organized into a purely user-defined and -managed directory hierarchy. In this way, the document can be uniquely addressed (on this machine) and be found by the user and applications.

However, these mechanisms entail all data organization and management effort to the user. Machines cannot understand the meaning of directory or file names, and file systems do not provide means for storage of advanced metadata, e.g. relations between files, or complex attributes. Analysis of content and

[1] The SemDAV project is partially founded by the FIT-IT Semantic Systems program of the Austrian Federal Ministry of Transport, Innovation and Technology, project number 812513.

user interaction on the personal desktop computer is in its infancy and mostly restricted to full-text indexing (cf. current desktop search engines like Google Desktop Search or Apple Spotlight).

We envision a file system for desktop computers that facilitates interoperable, open means for users and applications to not only store unstructured content, but also complex metadata and annotations, which can be created both manually and automatically. We envision applications that make extensive use of these options and store as much data and metadata as possible in formats supported by the file system. We envision browsing and searching interfaces for such file systems that are able to perfectly support the user's information needs.

In this paper, we introduce the SemDAV project, which will be carried out at the University of Vienna and the Research Studio Digital Memory Engineering. With SemDAV, we aim to contribute to these visions by extending the WebDAV protocol[5] with semantic features. WebDAV is by itself an extension to the HTTP protocol[3] and defines methods to transfer binary content and associated metadata in the form of attribute/value pairs[10]. WebDAV supports the creation and management of *collections* which are comparable to directories in a file system, and has built-in mechanisms for document version tracking. Many applications make use of WebDAV, especially calendar software through the CalDAV protocol[2], and WebDAV is (at least partially) supported by all major operating systems. By extending WebDAV, we can provide a smooth transition path for users and applications to semantically enriched file repositories without the need to immediately give up all known tools, procedures and practices.

2 A Semantic-Enabled File Storage

The primary goal of SemDAV is to extend the well-known WebDAV protocol with semantic features. However, such a protocol makes no sense without a repository which is able to store and manage content and associated metadata. Also, user clients and applications that are aware of the semantic features and utilize them to store and retrieve content-related metadata are needed. We envision to make the step from files, that are pure collections of bits, to *intelligent data objects*, that can be used to represent knowledge in machine-processable form. In this section, we describe features of such a object storage architecture, based on Semantic Web technology and accessible via the SemDAV protocol, and point out how these features overcomes problems and limitations imposed by traditional, hierarchical file systems.

Relations instead of hierarchies – The full path of a file (i.e. its directory path, file name, and extension) is its system-wide identifier and, simultaneously, a collection of user-defined metadata: The path can be interpreted as classification, the file name can be regarded as user annotation, summary, or keywords, and the extension may give a hint on the inner structure of a file. As a consequence, references to a file become invalid when the user decides to modify

[2] CalDAV: `http://www.caldav.org`

this organization structure. Moreover, it is impossible to attach more expressive metadata, possibly in different languages, to a file. We propose to solve the problems imposed by this fact by replacing file names with globally unique identifiers (most likely URIs), and adding descriptive data (ontology-based classification, user tagging, summary, keywords, etc.) as external metadata.

Cross-application metadata integration – Hierarchical file systems consist of a single-rooted tree wherein files can be placed at exactly one node. There exists no way to define multiple disjunct classification trees, or to place data objects to more than one location (except shortcuts, or aliases). As a consequence, applications tend to create their own data hierarchies in parallel to the user-defined data storage. Examples include e-mail clients or web browsers (bookmarks). To overcome this, a semantic file storage may not limit the classification of data objects to single hierarchies; instead, flexible assignment to classes and concepts must be possible, and these metadata must be formulated in an application-independent format. We believe that a Semantic Web technology-based file store would have the ability to bridge existing gaps between applications, and between applications and users.

Context persistence – A file's context, i.e. relations to other files, users, or workflows, is lost at the moment it is written to a storage medium, since current file systems treat files as stand-alone objects. This fact renders it impossible to associate files that are semantically related but stored at different places in a directory hierarchy or on different storage media. SemDAV will provide methods to store and retrieve such relations and other context metadata.

Efficient browsing and searching – Current file systems do not provide adequate support for browsing and searching. With the exception of full-text search engines, there exist no system-wide search facilities for user data in most operating systems, and metadata-based browsing tools are missing. Based on the SemDAV protocol, rich user clients can be implemented which make the metadata accessible to the user and are able to generate new metadata out of user interaction, content analysis, and manual annotation.

3 Example: A SemDAV-based Mail Application

To demonstrate the usefulness of semantically rich file systems, we plan to implement sample applications that utilize SemDAV to store their data and metadata. In this section, we describe features that we want to implement in a semantic e-mail client and compare them to existing applications.

Folder handling – All common mail applications support the management of mail messages using hierarchical folder trees. By doing so, these clients establish parallel, loose organizational structures within the user's data space. Our mail client does not define its own folder hierarchy, but uses the organizational structures of the underlying SemDAV system (which, as mentioned above, do not have to be hierarchical) to manage mail messages. This allows for unified handling of data objects, regardless of whether they have been created by the user, or are received by mail.

Tracking of mail access – Our mail application will be able to record all user interaction with mail messages, including *read*, *reply*, *forward*, *delete*, etc. By storing these access traces into the metadata-enabled file system, they will be made available for subsequent search queries. The experience of browsing the data repository is enriched since e.g. additional relations can be displayed. To increase interoperability and to make metadata interpretable for other applications, these traces will be stored according to a previously published ontology.

Attachment handling – In current mail software, handling of attachments is cumbersome. In our mail application, attachments will be stored as files, like all user documents. By doing so, the laborious task of storing attachments to folders, locating them, and re-attaching them, is avoided. Moreover, since in our system every file is identified by a globally unique identifier, our mail application may implement *download on demand*, instead of downloading all attachments by default.

Contact management – Contrary to current mail software, the semantic mail application will store all contacts as files with associated metadata in e.g. the FOAF format. Thus, contact data is accessible in SemDAV browsers, and can be treated like any other file: For instance, it can easily be sent to other users via e-mail.

Message searching – Our mail client does not need to implement additional searching capabilities, since the file system by itself will provide facilities to search content and associated metadata. This is supported by publishing the ontologies used for metadata to the file system management software.

Application integration – Since the mail application uses SemDAV to store all its metadata and publishes ontologies that are used for this, it is possible to integrate mail messages, attachments, and contacts with other applications, e.g. wiki software: We plan to integrate our mail client with Ylvi[7], developed at our research group, in order to allow mail messages and attachments to be published as wiki articles, and vice versa, with a single mouse click.

4 Related Work

The enrichment of file systems with more expressive metadata has been subject of research both in the scientific area and in commercial operating systems. Gifford[4] introduced a semantic file system for UNIX operating systems. *Sedar*[6] provides an archival-oriented semantic file system that treats every update to a file as new version and provides snapshots for file metadata. In the Windows NTFS file system, *Alternate Data Streams*[1] allow to add arbitrary application-specific metadata to files; Apple supports multiple *forks* per file, and most UNIX file systems allow to store and retrieve *extended attributes* for files. However, none of these approaches provides platform-independent metadata management, which causes loss of information when metadata is exchanged across different platforms. Various approaches aim to increase usability of traditional file systems: *TagFS*[8] uses RDF to store user annotations for files, while *Connections*[9] and *Beagle++*[2] aim to improve file system search by tracing file system calls to

infer context information, and by applying sophisticated ranking mechanisms to desktop objects, respectively. SemDAV attempts to integrate some of these concepts; however, we target a more radical solution that may render, in a long-term perspective, the hierarchical file system obsolete.

5 Conclusion

In this paper, we introduced the SemDAV project, which will extend the existing WebDAV protocol by semantic features, and implement semantic-enabled file repositories and browsing and searching clients. We described which problems originate from traditional, tree-based file systems and outlined how these problems can be solved by extending file systems with semantic features, and how users and applications can utilize the features of SemDAV in order to create more semantic-oriented desktop environments.

References

1. Hal Berghel and Natasa Brajkovska. Wading into alternate data streams. *Commun. ACM*, 47(4):21–27, 2004.
2. Paul-Alexandru Chirita, Stefania Ghita, Wolfgang Nejdl, and Raluca Paiu. Semantically enhanced searching and ranking on the desktop. In *Proceedings of the Int. Semantic Web Conference Workshop on The Semantic Desktop*, 2005.
3. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1 (RFC 2616), 1999. available at `http://www.ietf.org/rfc/rfc2616.txt`, accessed 7 July 2006.
4. David K. Gifford, Pierre Jouvelot, Mark A. Sheldon, and Jr. James W. O'Toole. Semantic file systems. In *SOSP '91: Proceedings of the 13th ACM symposium on Operating systems principles*, pages 16–25, New York, NY, USA, 1991. ACM Press.
5. Y. Goland, E. Whitehead, A. Faizi, S. Carter, and D. Jensen. HTTP extension for distributed authoring – WebDAV (RFC 2158), 1999. available at `http://www.ietf.org/rfc/rfc2518.txt`, accessed 7 July 2006.
6. Mallik Mahalingam, Chunqiang Tang, and Zhichen Xu. Towards a semantic, deep archival file system. In *FTDCS '03: Proceedings of the The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'03)*, page 115, Washington, DC, USA, 2003. IEEE Computer Society.
7. Niko Popitsch, Bernhard Schandl, Arash Amiri, Stefan Leitich, and Wolfgang Jochum. Ylvi - multimedia-izing the semantic wiki. In *1st Workshop on Semantic Wikis - From Wiki to Semantics*, 2006.
8. Simon Schenk, Olaf Görlitz, and Steffen Staab. TagFS: Bringing semantic metadata to the filesystem. In *Poster at the 3rd European Semantic Web Conference (ESWC)*, 2006. available at `http://www.eswc2006.org/poster-papers/FP31-Schenk.pdf`, accessed 7 July 2006.
9. Craig A. N. Soules and Gregory R. Ganger. Connections: Using context to enhance file search. In *Proceedings of the 20th ACM Symposium on Operating Systems Principles*, 2005.
10. E. James Whitehead and Yaron Y. Goland. The WebDAV property design. *Softw. Pract. Exper.*, 34(2):135–161, 2004.