

# TagTree: Storing and Re-finding Files Using Tags

Karl Voit<sup>1</sup>, Keith Andrews<sup>2</sup>, and Wolfgang Slany<sup>1</sup>

<sup>1</sup> Institute for Software Technology (IST),  
Graz University of Technology,  
Inffeldgasse 16b/2, A-8010 Graz, Austria

`Karl.Voit@IST.TUGraz.at`, `Wolfgang.Slany@tugraz.at`

<sup>2</sup> Institute for Information Systems and Computer Media (IICM),  
Graz University of Technology,  
Inffeldgasse 16c/1, A-8010 Graz, Austria  
`kandrews@iicm.edu`

**Abstract.** Although desktop search engines are now widely available on the computers of typical users, navigation through folder hierarchies is still the dominant mode of information access. Most users still prefer to store and search for their information within a strict hierarchy of folders.

This paper describes TagTree, a new concept for storing and retrieving files and folders using tagging and automatically maintained navigational hierarchies. TagTree is compatible with all currently prevalent software environments. A prototype implementation called *tagstore* provides a flexible framework for experimentation and a testbed for both usability studies and longer term field tests.

Preliminary test results show a very positive user acceptance rate of using TagTrees for storing and re-finding files.

**Keywords:** Tags, information re-finding, information architecture, folders.

## 1 Introduction

Users keep the vast majority of their personal files on local hard disk drives, even though network storage is becoming more popular. Successfully managing and accessing this data is crucial: finding the right information becomes increasingly difficult, as the sheer amount of information increases.

There are two kinds of information access methods: *navigation* (or *browsing*) methods, where the user typically steps through a hierarchy of folders, and *search* methods, where the user is magically transported to the item of interest. Much recent research has concentrated on search methods, in the form of desktop search engines and their query interfaces [1]. Navigational methods for local file retrieval have changed little from the principles described in the 1960s: files are placed into a hierarchy of folders. Cross-links to other destinations are hardly used by users, and operating systems make it hard for users to create and

maintain symbolic links [2]. However, studies like those of Barreau [3], Teevan et al. [4], Bergman et al. [1], or Alvarado et al. [5] have shown that users prefer navigation over searching. The method and software discussed in this paper therefore focuses on navigation and not search.

Unfortunately, hierarchical folder structures do not scale well [6, 7], suggesting that alternative navigational structures need to be investigated. Since the 1980s, numerous research tools for personal information management (PIM) have been developed, but none of them have made it onto the typical user's desktop of today: some reasons for this are described in Voit et al. [8].

This paper describes a new concept for storing and retrieving local files called TagTree. TagTree takes user-supplied tags and automatically generates and maintains a navigational tree (folder) structure of tags. The TagTree concept is implemented in a prototype called *tagstore*, which serves as a flexible testbed framework.

## 2 Related Work

WorkspaceMirror [9] introduced unified hierarchies across email folders, web page bookmarks, and personal document folders, allowing the user to benefit by reusing the same mental model to navigate through distinct information spaces.

Most research software products introduce some kind of new file browsing interface to the desktop. Interfaces providing spatial cues for orientation within a users' information space draw on real-world spatial metaphors. However, Lansdale [10] states that: "assertions made about the inherent value of visio-spacial information represent a simplistic view of human cognition and no guarantee of good design". He further remarks that pictures are less well-suited for recall than for differentiation between items.

Besides the definition of arbitrary attributes per item, the Presto system [11] allowed users to place documents in collections and workspaces. Presto also provided dynamic collections based on queries, later introduced in Apple's Mac OS X as "smart folders". In addition to its main interface based on Java Swing, Presto provided NFS-based shared network folders, which mapped collections for legacy applications.

With the File Attribute Browser [12], users can navigate files by selecting properties such as "modified this week", by selecting certain file types, or by specifying document sizes.

The Phlat interface presents a desktop search window containing the user's email, calendar events, local documents, and web history. The user is able to sort query results by columns such as title, date, author, email recipient, and so forth.

Feldspar [13] allows users to define relations to search within emails, files, web history, calendar events, and more. This approach supports the expression of access criteria such as the first email from a certain person met at a particular conference last May.

Other research tools provide an overview of user data using a timeline. Lifestreams [14] presented a visual timeline of every document and email a user ever send, received, or edited. MyLifeBits [15] provides an advanced interface which tries to combine as much personal information as possible. It even

includes telephone conversations, radio and television programs, mouse and keyboard events, audio- and video recordings, and periodic screenshots of the desktop. The goal of MyLifeBits is to record the whole life of a user and to visualise all the accumulated data on a single timeline.

PersonalBrain [16] allows the users to link between documents. In contrast to other mind mapping software products, PersonalBrain is not limited to strictly hierarchical structures, but items can be linked together in arbitrary ways.

The Haystack project [17] provided a framework to define semantic relations between objects, more fine-grained than files. The relation structure was kept separately from its user interface. An interface called Ozone provided navigation in the object space and users could define collections like in Presto.

Nepomuk [18] attempted to realise a social semantic desktop using the Resource Description Framework (RDF). Besides technological challenges, users do not seem quite ready for semantic technologies. RDF notation requires a radically new way of thinking and semantic tools like Nepomuk might only become of general interest in the far future.

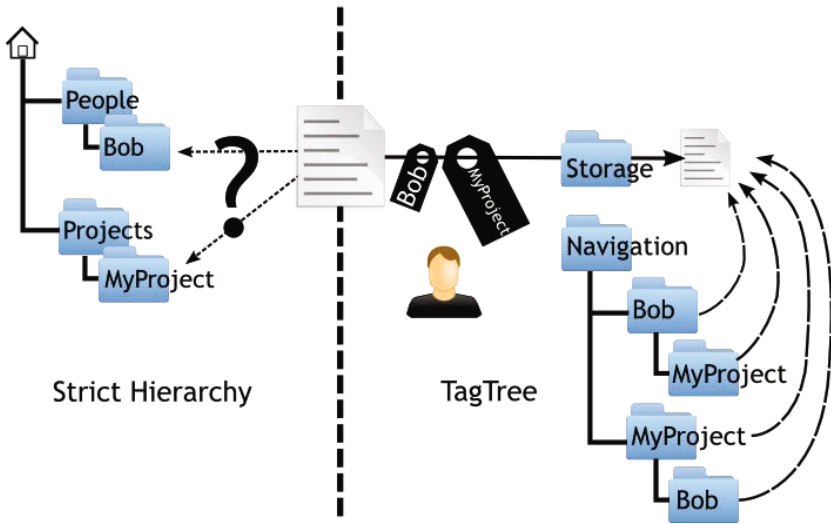
There are also some lower-level attempts to provide better user experience in information access. The idea of a semantic file system was discussed in Gifford et al. [7]. It introduced an NFS-based file system which allowed interactive search requests to be represented by dynamic folders. The semantic metadata of files was automatically extracted using format-dependent transducers. This work was remarkable in its approach, because there was no need for a new interface: the semantics of the file system and its folder structure were simply extended with a new concept, from which all existing applications could benefit without modification.

A more recent attempt to develop a semantic file system like Gifford et al. [7] is TagFS [19]. Using RDF, TagFS provides a semantic framework which allows tags to be attached to files within a special file system.

### 3 Tagging

Files in strict hierarchies of folders are insufficient for modern demands (Seltzer and Murphy [20], Fertig et al. [21], Shirky [22], Freeman and Gelernter [23]). The file system is the basic common interface shared by all software products. Although several projects promote semantic relations like Nepomuk [18], users are not yet ready to embrace such a radically new concept. However, in the context of Web 2.0, users have already widely adopted the practice of adding metadata to information in the form of tags. Studies show that tagging is a promising approach to handling multiple contexts related to information [7, 22, 24]. Tagging can be considered as a kind of semantic relation, but with only one fixed predicate “has\_tag”.

TagTree is a new concept for storing and retrieving files and folders using a tagging mechanism mapped to the file system hierarchy. Certain folders in the local folder hierarchy of the user’s computer have alternative semantics conferred upon them, in a manner similar to Gifford et al. [7]. The TagTree



**Fig. 1.** The TagTree concept when storing a file Bob’s ideas about MyProject.txt

concept is implemented in a software framework called *tagstore*. Instead of providing a special search interface, TagTree provides an alternative navigational folder structure based on user-supplied tags.

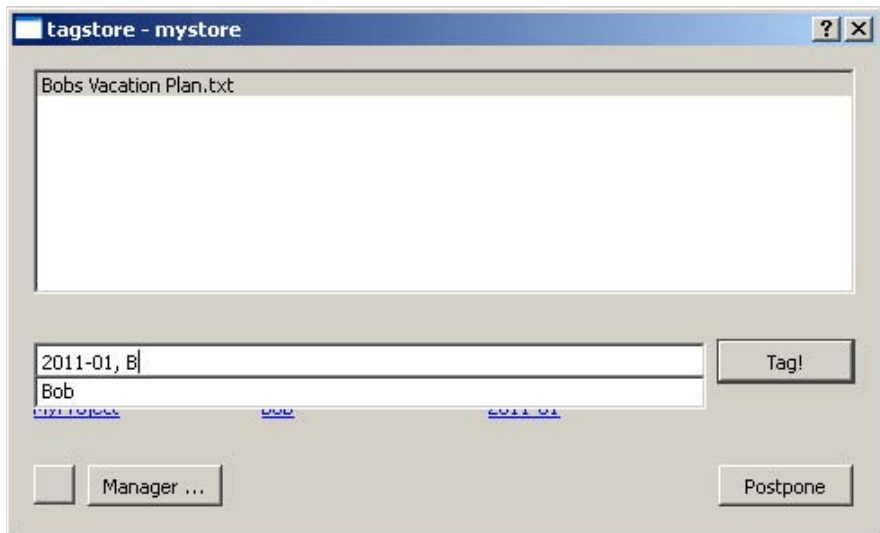
Since it is mapped to a standard folder hierarchy, TagTree is compatible with all standard applications, and the user is not confronted with a special navigational interface.

## 4 TagTree and tagstore

The concept of TagTree is illustrated in Figure 1. Rather than having to choose a destination folder within the folder hierarchy, the user stores all new items (files or folders) in a single folder, the central tagstore storage folder. When a new item is saved in this folder, a tagging dialog pops up and the user is able to enter one or more tags related to the item (see Figure 2).

After confirmation, tagstore creates a corresponding TagTree folder hierarchy in the tagstore navigation folder. The TagTree folder hierarchy consists of one folder path for each permutation of the tags associated with the item. Within each folder along each path, a symbolic link is created pointing to the original item stored in the central tagstore storage folder.

For example, to store a file Bob’s ideas about MyProject.txt in a tagstore, a user would first put the file into the central tagstore storage folder. In the tagging dialog, tags such as Bob and MyProject might be given to the file. Henceforth, the single target file may be found along any one of four possible paths Bob, MyProject, Bob/MyProject, and MyProject/Bob, as well as directly in the central tagstore storage folder. Figure 3 shows the corresponding, fully expanded TagTree.



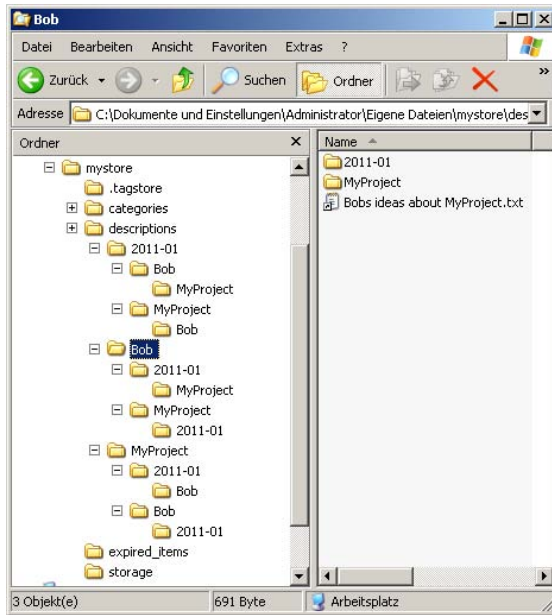
**Fig. 2.** The tagstore dialog window. The user has started typing a new tag beginning with “B”, so tagstore pops open a window showing all previously used tags with that prefix (here “Bob”)

When the user wants to access the item, it is not necessary to remember a single strict series of folder names. Navigation can begin using any of the associations connected with the item. The more tags the user can remember, the more specific the implicit query becomes, and the fewer items are presented at that level. The user does not even have to finish navigating the entire path down to a folder leaf: whenever an item in the current list of matching items is recognised, it can be accessed directly.

## 5 The Benefits of tagstore and Tagging

The concept of storing items in TagTrees provides several improvements compared to classical folder hierarchies. Metadata is added by the user during the storing process, a time when the user has a maximum of contextual information about the item available. Using a tagging mechanism supports the expression of a rich variety of metadata. Typical folder structures and item names do not support a large variety of metadata.

The tagstore implementation supports the user with features such as automatic datestamps. If enabled, datestamps are written as default tags into the tagline. Similarly, the user is able to define expiry dates for items added to a tagstore. After an expiry date is reached, the corresponding item is moved automatically into a special folder called **expired items**. Users tend to keep files for a long time and do not delete them, even when they are only for temporary



**Fig. 3.** Windows Explorer with the expanded TagTree of the file *Bobs ideas about MyProject.txt* tagged with the tags 2011-01, Bob, and MyProject

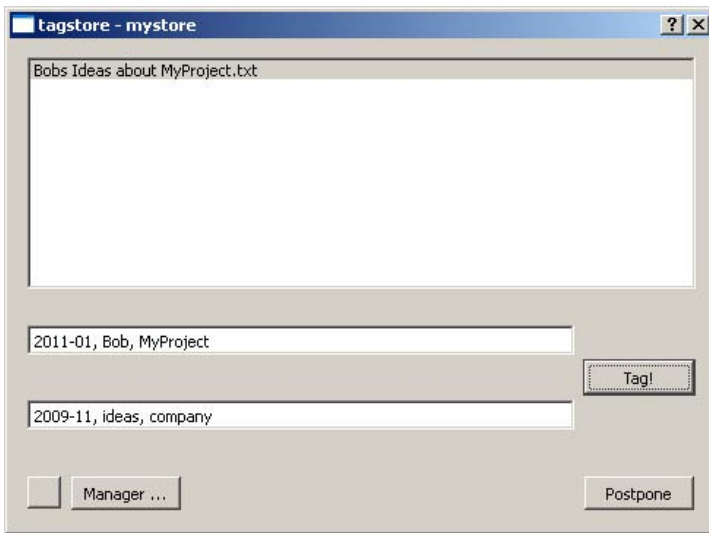
use [3]. With this feature, users are able to define explicit temporal ranges of interest for information [25].

Path names and file names mostly reflect a single dimension of contextual information, such as the document's title. Tagging offers the possibility to use multi-dimensional metadata, such as an arbitrary combination of dates, authors, project, events, and so forth.

## 6 Limitations of tagstore

The current implementation of tagstore has some technological limits. Items with many tags result in a large number of folders and links. Current file systems have a fixed number of possible file or folder entries (inodes) per hard disk partition. Therefore, a reasonable upper bound of items per tagstore is only a few thousand items. For testing purposes, this limit should be no problem.

There is an exponential relation between the number of tags of an item and the number of folders and links that have to be created. This results in a performance problem when tagging an item with many tags. On conventional hardware, a reasonable upper bound of tags per item is six. Currently, if a user enters more than six tags, a warning message appears and disables the Tag button. The button is re-enabled, when the user reduces the number of tags to six or fewer. Several studies have shown that the number of tags users assign to items is generally below six in the vast majority of tagging tasks [26, 27].



**Fig. 4.** The tagstore dialog window showing two separate taglines for categorising tags and describing tags

## 7 Preliminary Results

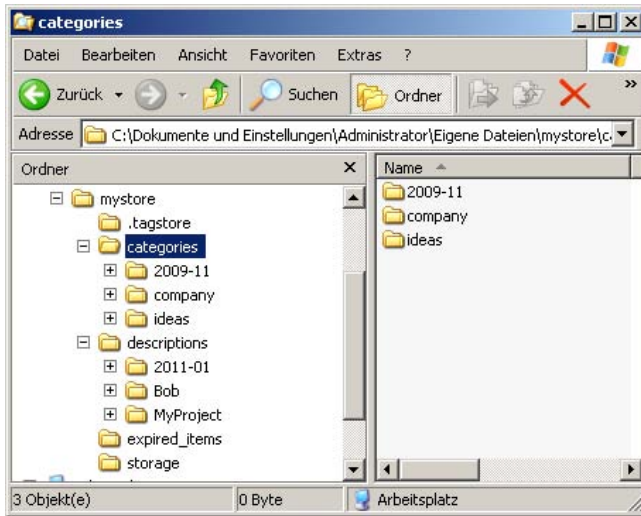
A formal experiment (counterbalanced  $2 \times 2$  repeated measures) was conducted in January 2011. A total of 18 participants compared storing and then re-finding files in Windows Explorer using folder hierarchies and with tagstore using tags.

Two task sets were prepared. In the first task set, the participant had to store thirty files of a general nature into a folder hierarchy also constructed by the user. In the second task set, the same set of files had to be added to a tagstore and appropriately tagged. The order of the two task sets was counterbalanced between two groups of 9 users each. The files consisted of ten PDF text documents, ten (computer generated) graphics, and ten photographs.

The dependent variables measured were time to file/tag, time to re-find, number of navigation subtasks to re-find files, depth of folder hierarchy per file, and the number of tags assigned.

The preliminary results appear to show no statistically significant differences for filing or re-finding files. The feedback questionnaire showed a statistically significant ( $p < 0.05$ ) subjective preference of the users for the tagstore method over the folder hierarchy: 14 users (77.8%) preferred tagstore and only 4 users (22.2%) preferred folders. This result is remarkable, since the experiment was the first time participants had seen tagstore or the TagTree concept and half of the users had previously never used any form of tagging.

Comments made by the participants suggest that field studies might show even better acceptance and re-finding performance. Several users suggested that



**Fig. 5.** Windows Explorer showing separate TagTrees for categorising tags and describing tags

using tagstore on their own files (rather than the generic files provided in the study) might provide even more benefit.

## 8 Future Work

The TagTree concept and its tagstore implementation are not an exhaustive solution for information storage and retrieval. As Lansdale [10] states, tools for information retrieval have to be constantly enhanced and can never be complete.

Further formative usability tests (thinking aloud tests, heuristic evaluations) are planned to improve the usability of tagstore. Further comparative studies (formal experiments) will test the effectiveness of controlled vocabularies, multiple taglines, and tag recommendations. In addition, it is intended to run longer-term field tests with tagstore, over a period of several weeks or months. An instrumented version of tagstore will allow log files to be generated and collected for analysis.

In the current tagging dialog, users are offered suggestions for tags based on the most recently used and most often used tags. In future, a more sophisticated recommender system will be used to generate tag suggestions.

Recent studies of social tagging systems [28] shows that tags might be able to be split up into categorizing tags and describing tags. The current implementation of tagstore also provides the possibility for multiple taglines to test the benefits of separate tag categories (Figures 4 and 5).

The problem with homonyms and synonyms of tags is an important issue for persons using tagging systems. Controlled vocabularies, where users are able to define their own set of allowed tags, can help to alleviate this problem.



## 9 Concluding Remarks

Information re-finding is a crucial part of our daily work. Studies show that navigation is preferred to search by users. Items located in strict hierarchies of folders are insufficient for modern demands.

TagTree is introduced as a new method to organize items (files or folders) for re-finding by navigation. Tags entered by the user generate navigational folder structures. Within TagTree structures, users are able to re-find items by association. This method is implemented in a research framework called tagstore.

Using tagstore as a research framework, a wide range of research topics can be investigated: tag management needs, tag vocabulary and tag taxonomy issues, the tagging behaviour of users over time, differences between users, and so forth.

It is hoped that the research results emanating from tagstore can lead to improvements in the user experience concerning personal information storage and retrieval. To have any widespread impact, though, these results will need to find their way into user applications, operating systems, and file systems of the future.

## References

1. Bergman, O., Beyth-Marom, R., Nachmias, R., Gradovitch, N., Whittaker, S.: Improved search engines and navigation preference in personal information management. *Transactions on Information Systems* 26(4), 1–24 (2008) ISSN 1046-8188, doi:10.1145/1402256.1402259
2. Gonçalves, D.J., Jorge, J.A.: An Empirical Study of Personal Document Spaces. In: Jorge, J.A., Jardim Nunes, N., Falcão e Cunha, J. (eds.) *DSV-IS 2003*. LNCS, vol. 2844, pp. 46–60. Springer, Heidelberg (2003), <http://virtual.inesc.pt/dsvis03/papers/05.pdf>, doi:10.1007/b13960
3. Barreau, D.: The persistence of behavior and form in the organization of personal information. *Journal of the American Society for Information Science and Technology* 59(2), 307–317 (2008) ISSN 1532-2882, doi:10.1002/asi.20752
4. Teevan, J., Alvarado, C., Ackerman, M.S., Karger, D.R.: The perfect search engine is not enough: a study of orienteering behavior in directed search. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2004*, pp. 415–422. ACM, New York (2004), <http://people.csail.mit.edu/teevan/work/publications/papers/chi04.pdf>, doi:10.1145/985692.985745
5. Alvarado, C., Teevan, J., Ackerman, M.S., Karger, D.: Surviving the information explosion: How people find their electronic information. *AI Memo AIM-2003-006*, MIT AI Laboratory, Department of Computer Science (2003), <http://hdl.handle.net/1721.1/6713>
6. Dourish, P., Edwards, W.K., LaMarca, A., Salisbury, M.: Presto: An experimental architecture for uid interactive document spaces. *Transactions on Information Systems* 6(2), 133–161 (1999), <http://www.dourish.com/publications/1999/tochi-presto.pdf>, ISSN 1073-0516, doi:10.1145/319091.319099

7. Gifford, D.K., Jouvelot, P., Sheldon, M.A., James, W., O'Toole, J.: Semantic file systems. In: Proc. 13th ACM Symposium on Operating Systems Principles (SOSP 1991), pp. 16–25. ACM (October 1991), <http://cgs.csail.mit.edu/history/publications/Papers/sfs.ps>, doi:10.1145/121132.121138
8. Voit, K., Andrews, K., Slany, W.: Why personal information management (pim) technologies are not widespread. In: ASIS&T 2009 Workshop on Personal Information Management (PIM 2009) (November 2009), <http://pimworkshop.org/2009/papers/voit-pim2009.pdf>
9. Boardman, R., Sasse, M.A., Spence, B.: Life beyond the mailbox: A crosstool perspective on personal information management. In: Proc. CSCW 2002 Workshop on Redesigning Email for the 21st Century. ACM (November 2002), <http://www.iis.ee.ic.ac.uk/~rick/research/pubs/email-cscw2002.pdf>
10. Lansdale, M.W.: The psychology of personal information management. *Applied Ergonomics* 19(1), 55–66 (1988), <http://simson.net/ref/1988/Lansdale88.pdf>, ISSN 0003-6870, doi:10.1016/0003-6870(88)90199-8
11. Dourish, P., Edwards, W.K., LaMarca, A., Salisbury, M.: Using properties for uniform interaction in the presto document system. In: Proc. 12th Annual ACM Symposium on User Interface Software and Technology (UIST 1999), pp. 55–64. ACM (November 1999), <http://www2.parc.com/csl/projects/placeless/papers/uist99-presto.pdf>, doi:10.1145/320719.322583
12. Marsden, G., Cairns, D.E.: Improving the usability of the hierarchical file system. In: Proc. Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on Enablement through Technology (SAICSIT 2003), South African Institute for Computer Scientists and Information Technologists (SAICSIT), pp. 122–129 (September 2003), <http://pubs.cs.uct.ac.za/archive/00000190/01/saicsit2003-dec.pdf>, ISBN 1581137745
13. Chau, D.H., Myers, B., Faulring, A.: What to do when search fails: Finding information by association. In: Proc. 26th SIGCHI Conference on Human Factors in Computing Systems (CHI 2008), pp. 999–1008. ACM (April 2008), <http://www.cs.cmu.edu/~dchau/feldspar/feldspar-chi08.pdf> doi:10.1145/1357054.1357208
14. Freeman, E., Fertig, S.: Lifestreams: Organizing your electronic life. In: AAAI Fall Symposium 1995. AAAI (November 1995), <http://www.aaai.org/Papers/Symposia/Fall/1995/FS-95-03/FS95-03-007.pdf>
15. Gemmell, J., Bell, G., Lueder, R.: Mylifebits: A personal database for everything. *Communications of the ACM* 49(1), 88–95 (2006), <http://research.microsoft.com/pubs/64157/tr-2006-23.pdf>, ISSN 0001-0782, doi:10.1145/1107458.1107460
16. Hugh, H.: Personal brain (November 2010), <http://www.thebrain.com/>
17. Huynh, D., Karger, D.R., Quan, D.: Haystack: A platform for creating, organizing and visualizing information using rdf. In: Proc. International Workshop on the Semantic Web (WWW 2002) (May 2002), <http://semanticweb2002.aifb.uni-karlsruhe.de/proceedings/Research/huynh.pdf>
18. Bernardi, A.: Nepomuk: The social semantic desktop (November 2010), <http://nepomuk.semanticdesktop.org/>

19. Bloehdorn, S., Görlitz, O., Schenk, S., Völkel, M.: Tagfs — tag semantics for hierarchical file systems. In: Proc. 6th International Conference on Knowledge Management (I-KNOW 2006), pp. 304–312 (September 2006), <http://triplei.tugraz.at/blog/wp-content/uploads/2008/11/37tagfs.pdf>
20. Seltzer, M., Murphy, N.: Hierarchical file systems are dead. In: Proceedings of the 12th Workshop on Hot Topics in Operating Systems HOTOS 2009, Monte Verita, Switzerland (May 2009)
21. Fertig, S., Freeman, E., Gelernter, D.: Finding and reminding. reconsidered. SIGCHI Bulletin 28(1), 66–69 (1996) ISSN 0736-6906, doi:10.1145/249170.249187
22. Shirky, C.: Ontology is overrated: Categories, links and tags (2005), <http://www.shirky.com/writings/ontologyoverrated.html>
23. Freeman, E., Gelernter, D.: Lifestreams: A storage model for personal data. ACM SIGMOD Bulletin 25, 80–86 (1996)
24. Quan, D., Bakshi, K., Huynh, D., Karger, D.R.: User interfaces for supporting multiple categorization. In: Proc. 9th IFIP TC13 International Conference on Human-Computer Interaction (INTERACT 2003), pp. 228–235. IOS Press (September 2003), <http://www.idemployee.id.tue.nl/g.w.m.rauterberg/conferences/INTERACT2003/INTERACT2003-p228.pdf>, ISBN 1586033638
25. Mayer-Schönberger, V.: Delete: The Virtue of Forgetting in the Digital Age. Princeton University Press (October 2009) ISBN 0691138613
26. Hsieh, J.L., Chen, C.H., Lin, I.W., Sun, C.T.: A web-based tagging tool for organizing personal documents on pcs. In: International Conference of Computer-Human Interaction 2008 (CHI 2008), Florence, Italy (April 2008), <http://works.bepress.com/lucemia/18/>
27. Pak, R., Pautz, S., Iden, R.: Information organization and retrieval: A comparison of taxonomical and tagging systems. Cognitive Technology 12(1), 31–44 (2007), <http://business.clemson.edu/Catlab/pubs/pak-pautziden-2007.pdf>
28. Strohmaier, M., Koerner, C., Kern, R.: Why do users tag? detecting users' motivation for tagging in social tagging systems. In: Proc. 4th International AAAI Conference on Weblogs and Social Media (ICWSM 2010). Association for the Advancement of Artificial Intelligence (May 2010), <http://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/viewFile/1497/1892>