Tagging Might Not Be Slower Than Filing in Folders

Karl Voit
Institute for Software
Technology (IST)
Graz University of Technology
Austria
Karl.Voit@IST.TUGraz.at

Keith Andrews
Institute for Information
Systems and Computer Media
(IICM)
Graz University of Technology
Austria
kandrews@iicm.edu

Wolfgang Slany
Institute for Software
Technology (IST)
Graz University of Technology
Austria
Wolfgang.Slany@TUGraz.at

CHI'12, May 5–10, 2012, Austin, Texas, USA.
ACM 978-1-4503-1016-1/12/05.

Abstract

Tagging is a promising method for organising and refinding information. However, studies comparing tagging mechanisms to organising information in folder hierarchies are relatively scarce. A study with the software framework *tagstore* shows that tagging does not necessarily mean slower filing performance. For experienced users, tagging required less time, fewer mouse clicks, and had very high acceptance rates.

Author Keywords

tagging, tagstore, TagTrees, framework, information retrieval, re-finding, file storage, hierarchical structures, personal information management (PIM)

ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation]: User interfaces—evaluation/methodology.

Introduction

Despite the popularity of web-based storage, users still keep the vast majority of their personal files on local hard drives. Successfully managing and accessing this data is crucial. As the sheer amount of information increases, finding the right information is becoming increasingly difficult. Information access methods can be categorised into two general kinds: *navigation* (or *browsing*) methods, where the user typically steps through a hierarchy of folders, and *search* (or *teleporting)* 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 [2]. 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 [7]. However, studies like those of [1] or [2] have shown that users prefer navigation over searching. Hence, there is a need for renewed research into the use of navigational methods for local file retrieval.

	multiple item types	folder hierarchy	supports browsing	easy to use
Pak [9]	✘	✓	✘	✘
Civan [3]	✘	✘	✘	✘
Ma [8]	✘	✓/✘	✘	✘
Rodden [10]	✘	✘	✘	✓
tagstore	✓	✓	✓	✓

Table 1: Overview of systems used in related studies.

It is clear that, for the majority of users, topical hierarchical folder structures do not work well [5] and that alternative navigational structures need to be investigated [4, 6]. 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. Several reasons for this are described in [14].

This paper describes a new software framework called *tagstore* for testing issues related to tagging files and folders on the local hard disk. The framework is available for every major operating system, and can be flexibly configured to support a range of tagging strategies.

Related Work

Only very few studies have addressed the topic of whether storing items in folder hierarchies or storing items using labels or tags results in better organising and refinding performance, and their results have been largely inconclusive. Pak et al. [9] conducted two experiments which produced contradictory results. In the first experiment, users were asked to organise 60 photographs on a desk without any computer. Organising by assigning tags was significantly faster than grouping into a hierarchy. In the second experiment, the same 60 photographs were organised either into categories or by assigning tags using a forms-based web interface on a computer. In this case, organising by grouping into categories was significantly faster. There were no significant differences in refinding times. A serious drawback of this study is that a non-standard (web form) interface was used for the categories rather than using standard folders and an explorer. Civan et al. [3] asked users to compare organising emails within the Hotmail web interface (folders) and Google Gmail (tagging). Although the main focus of the study was on qualitative measures, no significant differences were found in refinding performance. Ma and Wiedenbeck [8] compared tagging and folder structures (and a hybrid with both folders and tags) using a Firefox add-on. No significant differences in performance were found between tagging and folders for either organising or refinding tasks. Rodden and Leggett [10] describe the new Google Gmail interface, which provides both labels and a limited form of folders, but do not report any kind of comparative evaluation.

Each of the previously mentioned studies has some issues related to its test environment. Most studies organised only one specific item type (only emails or only photographs) rather than heterogeneous items. The capabilities of folders varied greatly. Some studies used an interface showing the contents of each folder, but without allowing a browsing/navigation process at all. Other systems did not provide for subfolders, thus neglecting the essential aspect of hierarchy. Most of the systems required the user to type in every character of the tags without any

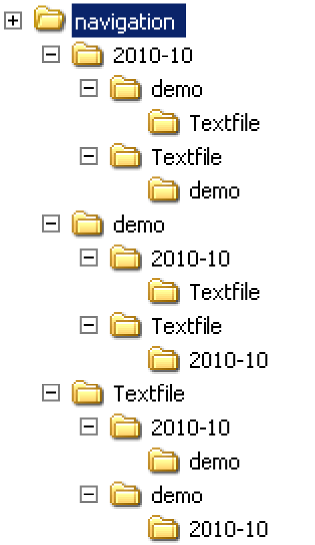
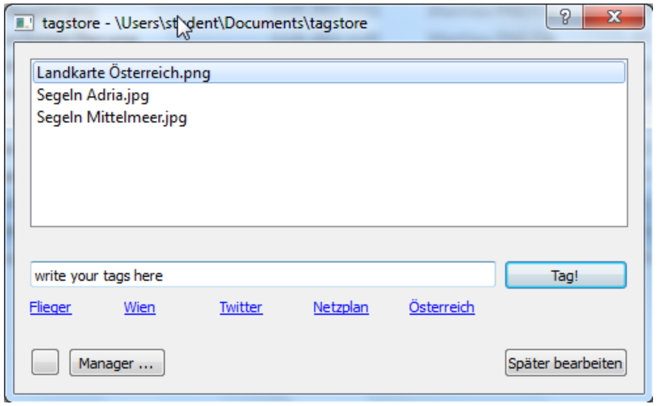


Figure 1: A sample tagstore TagTree of a single item tagged with the three tags "2010-10", "demo", and "textfile" shown in Windows Explorer.

tag completion mechanism. This is a major disadvantage for tagging and does not allow fair comparison. Table 1 gives an overview of these implications related to the studies mentioned above.



Landkarte Österreich.png
Segeln Adria.jpg
Segeln Mittelmeer.jpg

Flieger Wien Twitter Netzplan Österreich

Figure 2: The tagstore dialog window. For the experiment, the tagstore was configured with one tag-line, no controlled vocabulary, no pre-filled datestamp, and tag recommendations active.

tagstore and TagTrees

Files and folders (henceforth "items") appear only in one single place in a traditional file system[1]. The *tagstore* framework [13] provides an extension layer on top of the file system, where items can be tagged during the storing process: the user adds items to a *tagstore* by saving them to one central folder called storage, which preferably should be easily reached. Once the item is placed in the storage folder, the *tagstore* tagging dialog window appears, as shown in Figure 2. Convenience features such as pre-filled tags, tag completion, and a tag recommender system supports the user while assigning one or more suitable tags.

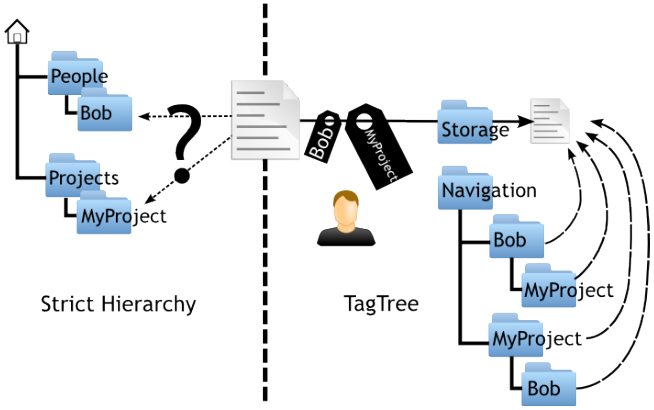


Figure 3: Folder hierarchy and TagTree concept using an example file `Bobs_ideas_about_MyProject.txt`

After the user confirms the tagging dialog, *tagstore* automatically creates associated link structures, called *TagTrees*. In the navigation folder, each directory reflects the tags of the user. The user can navigate to each item using any (sub-)permutation of its related tags. Figure 3 illustrates an example TagTree for an item with two tags: `Bob` and `MyProject`. Figure 1 shows an expanded TagTree for only one item tagged with three tags. Items are actually stored in the single central `storage` folder and they can be found there as well as navigated to using the dynamically synchronised TagTrees in the `navigation` folder hierarchy. Within the TagTrees structure, users can re-find an item using any permutation of its associated tags. No TagTrees folder is ever empty. In this respect, TagTrees are similar to faceted navigation [11]. Further information about *tagstore* can be found in [13, 15, 16].

[1] Duplicated items and hardly used (symbolic) links neglected.

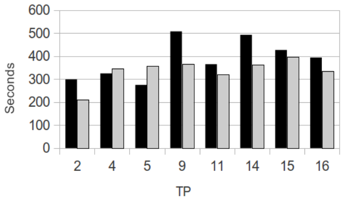


Figure 4: Filing performance into folder hierarchy (black) and tagstore (gray) of fast performers only.

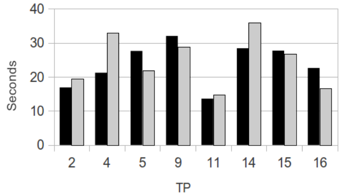


Figure 5: Re-finding performance within folder hierarchy (black) and tagstore (gray) of fast performers only.

Formal Experiment

A formal experiment (repeated measures, within-groups) was conducted with 18 test persons (TPs)[2] involving storing in *tagstore*[3] and re-finding within TagTrees. After a short period of familiarisation with the 30 test items[4], TPs had to file the items. Group 1 started by storing the documents into a folder structure they had to create themselves. Group 2 started by filing and tagging the items into *tagstore*. Afterwards, TPs performed filing with the other method as well. Then there was a break of at least 15 minutes, where TPs left the experiment room. After the break, TPs had to re-find a set of six test items. Once again, Group 1 started with the self-made folder structure, group 2 with the TagTrees. Afterwards, the TPs had to locate the same items using the other method as well. The TPs were given both specific and vague information about which item to find next. At the end, each TP was interviewed in both an open ("How was it?") and structured (questionnaire) form.

The whole experiment was filmed and *very* detailed, manual transcriptions (all timings for tasks, distractions, and so forth, feature usage, comments, even down to file management operations) were derived from the footage after the experiment. The raw data, materials, scripts, direct and indirect results are all available online [12].

Results

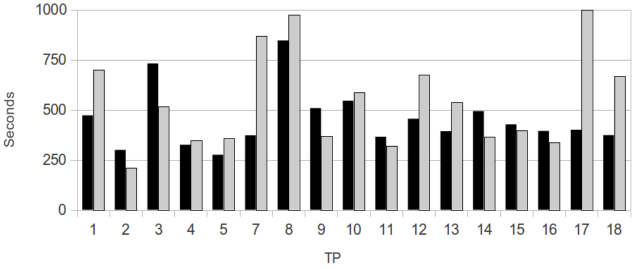


Figure 6: Filing performance into folder hierarchy (black) and tagstore (gray) of all test persons.

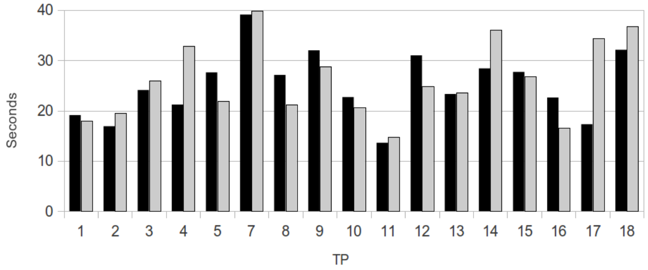


Figure 7: Re-finding performance within folder hierarchy (black) and tagstore (gray) of all test persons.

Test person number six had to be disregarded because of constant talking, making separation of task time and comment time impossible. Except for technical issues, every item was found by the TPs. The test framework *tagstore* had some minor bugs but all interruptions related to software problems were logged and removed from the actual task times.

[2] Median age 28; high education level; seven students and 11 employees; average computer experience of 16 years; average computer usage of 45 hours per week; MS Windows-dominant; 50% not currently using tagging.

[3] tagstore version 226, with one tag-line, no controlled vocabulary; Microsoft Windows 7, Windows Explorer.

[4] Ten graphics, ten photographs, ten text documents (PDF).

	$\bar{x}$	σ
Filing (secs) F	451.4	147.5
Filing (secs) T	542.5	239.7
Re-find (secs) F	25.0	6.5
Re-find (secs) T	26.0	7.6
Re-find (clicks) F	19.0	0.7
Re-find (clicks) T	13.9	0.3

Table 2: Mean and standard deviation of the results for *all TPs*: folder hierarchies (F) and tagging (T).

	$\bar{x}$	σ
Filing (secs) F	385.8	86.3
Filing (secs) T	337.4	56.0
Re-find (secs) F	23.8	6.3
Re-find (secs) T	24.7	7.7
Re-find (clicks) F	19.4	0.9
Re-find (clicks) T	14.5	0.4

Table 3: Mean and standard deviation of the results for the *fast performers*: folder hierarchies (F) and tagging (T).

The feedback questionnaire results showed an overwhelming preference: 78% of the TPs said that they prefered *tagstore* over their current hierarchy of folders. This is quite remarkable since at that time, *tagstore* still had numerous minor bugs and 50% of the TPs were not currently using tagging systems. TPs thought that both storing and re-finding items with *tagstore* was faster and that *tagstore* supports users better than traditional folders.   The objective measures in Figure 6, Figure 7, and Table 2 tell a different story: the average filing time in the folder hierarchy was faster than the time used for filing in *tagstore*. For the re-finding tasks, there was no notable difference between folders and *tagstore*, but the average number of mouse clicks performed for re-finding items was much higher for folders than for *tagstore*. These are work in progress results, thorough statistical analysis and significance tests are still outstanding. The large standard deviation for filing illustrates the performance differences between TPs. One reason might be related to differences between TPs for performing file management in general. Slower TPs might dilute a clearer result. After ordering the overall task times of the TPs, there is a distinct sub-group comprised of the eight fastest TPs, equally divided into group 1 and 2. This sub-group is denoted as the "fast performers".

Within the fast performers, filing in folder structure is much more similar to the filing measures of *tagstore* as shown in Table 3, Figure 4, and Figure 5. For the re-finding tasks, there remains no notable difference.

Comparing the fast performers and the other TPs showed certain differences: fast performers used more file moving actions when filing into folder hierarchies, used fewer tags within *tagstore*, were distracted much less[5], did not re-inspect the test items that often, and used much less file move actions to file into *tagstore*. It is worth noting that the percentage of TPs already using tagging is much lower among the fast performers: $3/8$ (38%) compared to $6/9$ (67%).

Conclusions and Outlook

Studies of filing and re-finding in hierarchies compared to within tagging systems are hard to conduct in a balanced way. Previous comparative studies had severe implementation issues (see Table 1), which do not allow reasonable comparison. Using *tagstore* allows for a fairer experimental setup [16].

The formal experiment described here shows that, in general, filing seems to be faster in folder hierarchies whereas re-finding does not show any notable differences. For a sub-group of fast performers, tagging seems to be better: filing in folders is approximately as fast as filing by tagging with *tagstore*. For re-finding in general, *tagstore* requires less mouse clicks. Subjective feedback from the test users was very positive for *tagstore* compared to the traditional method. Having adequate usability features such as tag completion, default tags, and a tag recommender system make it possible to provide a better subjective user experience while not necessarily requiring more time. Navigational structures like TagTrees provide associative browsing to enhance the user experience. The authors believe that in a future experiment (with more users, more test items, and a longer pause between filing and re-finding tasks), the advantages of *tagstore* might become even more apparent.

[5] Distraction times were subtracted from the numbers everywhere.

References

[1] C. Alvarado, J. Teevan, M. S. Ackerman, and D. Karger. Surviving the information explosion: How people find their electronic information. AI Memo AIM-2003-006, MIT AI Laboratory, Department of Computer Science, 2003.

[2] O. Bergman, R. Beyth-Marom, R. Nachmias, N. Gradovitch, and S. Whittaker. Improved search engines and navigation preference in personal information management. *Transactions on Information Systems*, 26(4):1–24, Sept. 2008.

[3] A. Civan, W. Jones, P. Klasnja, and H. Bruce. Better to organize personal information by folders or by tags?: The devil is in the details. *Proceedings of the American Society for Information Science and Technology*, 45(1):1–13, 2008.

[4] P. Dourish, W. K. Edwards, A. LaMarca, and M. Salisbury. Presto: An experimental architecture for fluid interactive document spaces. *Transactions on Information Systems*, 6(2):133–161, June 1999.

[5] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The Vocabulary Problem in Human-System Communication. *Communications of the ACM*, 30(11):964–971, Nov. 1987.

[6] D. K. Gifford, P. Jouvelot, M. A. Sheldon, and J. James W. O'Toole. Semantic file systems. In *Proc. 13th ACM Symposium on Operating Systems Principles (SOSP 1991)*, pages 16–25. ACM, Oct. 1991.

[7] D. J. Goncalves and J. A. Jorge. An empirical study of personal document spaces. In *Proc. 10th International Workshop on Design, Specification and Verification of Interactive Systems (DSV-IS 2003)*, volume 2844 of *LNCS*, pages 46–60. Springer, June 2003.

[8] S. Ma and S. Wiedenbeck. File management with hierarchical folders and tags. In *Proc. 27th SIGCHI Conference on Human Factors in Computing Systems (CHI 2009) Extended Abstracts*, pages 3745–3750. ACM, Apr. 2009.

[9] R. Pak, S. Pautz, and R. Iden. Information organization and retrieval: A comparison of taxonomical and tagging systems. *Cognitive Technology*, 12(1):31–44, 2007.

[10] K. Rodden and M. Leggett. Best of both worlds: Improving gmail labels with the affordances of folders. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI 2010)*, Apr. 2010.

[11] D. Tunkelang. *Faceted Search*. Morgan Claypool, June 2009. ISBN 1598299999 .

[12] K. Voit. Open science: Raw data, derived data, scripts, analysis and remarks on a formal experiment conducted using tagstore. Online Code Repository, Jan. 2011. https://github.com/novoid/2011-01-tagstore-formal-experiment.

[13] K. Voit. tagstore, 2012. http://www.tagstore.org/.

[14] K. Voit, K. Andrews, and W. Slany. Why personal information management (PIM) technologies are not widespread. ASIS&T 2009 Workshop on Personal Information Management (PIM 2009), Nov. 2009.

[15] K. Voit, K. Andrews, and W. Slany. TagTree: Storing and re-finding files using tags. In *Proc. 7th Conference of the Austrian Computer Society Workgroup: Human-Computer Interaction (Usab 2011)*, volume 7058 of *LNCS*, pages 471–481. Springer, Nov. 2011.

[16] K. Voit, K. Andrews, and W. Slany. Creating a comparative environment for pim evaluation. In *PIM12 CSCW 2012 Workshop*, Seattle, WA, USA, Feb. 2012.