Position Paper

Why Personal Information Management (PIM) Technologies Are Not Widespread

And What to do About It

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

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

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

ABSTRACT

Users of computer systems create and store valuable personal information in files, email folders, and bookmark collections. For decades, the main principle of interacting with files, emails, and bookmarks has remained unchanged: hierarchical directory trees with standard (Windows Explorer style) browsers.

Users often have problems both in classifying new items and maintaining a classification hierarchy as such. With files, emails, and bookmarks, users often end up maintaining three parallel classification hierarchies, one in each tool. Over the past thirty years, a number of alternative personal information management (PIM) tools have emerged, but the typical user is still faced with hierarchical directory structures.

This position paper addresses some of the reasons why modern PIM tools are not widespread and proposes a set of eight requirements for future PIM tools.

Keywords

information retrieval systems, file storage, hierarchical structures, dynamical structures, tagging, faceted search.

1. INTRODUCTION

Much progress has been made in the development of new personal information management (PIM) tools and ideas over the past three decades. However, many of these have failed to make it beyond the research laboratory and onto the PCs of typical computer users. A number of factors may lie behind that.

1.1 How Users Organise

Several studies have investigated user behaviour when organising information in paper-based offices. Malone [15] studied user behaviour regarding information in a physical office environment. He identified *files* (explicitly titled and logically arranged collections of information, for example in folders or binders) and *piles* (untitled piles of information arranged by physical location) as the main schemes employed by users to organise their information.

Lansdale [13] makes the point that users make use of piles to compensate for the difficulty of classifying (filing) things: "To avoid the process of classification, ..., he puts objects in a particular place. With this he forgoes the opportunity to retrieve the document by any simple classification-based search." [13, page 56]. As both Malone and Lansdale argue, and will be seen later, it is extremely difficult to create and maintain a neat, intuitive hierarchical classification scheme (or taxonomy) for documents.

Regarding computer-based information management strategies, Barreau interviewed seven managers [1]. She identified four common sub-activities in their management of information: (1) *acquisition* of items, (2) *classification* of items, (3) *maintenance* of the collection, (4) *retrieval* of items. She also found that users developed highly personalised strategies for organising their information and documents and that, broadly, three types of information could be identified: *ephemeral* (temporary), *working*, and *archived* (dormant).

In a similar study done around the same time, Nardi et al [19] interviewed 15 Macintosh users about their information management behaviour. The combined analysis of both studies [3] showed that users (1) liked to arrange resources by location (for example by grouping icons on the desktop), (2) avoided elaborate filing schemes, and (3) archived relatively little information. Macintosh users tended to use subdirectories to organise information, whereas DOS users did not.

Barreau re-interviewed four of the seven managers ten years later [2]. The four continued to leave most of their documents in a catch-all directory (such as My Documents) and still rarely grouped or classified documents into folders or directories (this was the case for all 7 managers in the first study).

Whittaker and Sidner [26, page 280] reported three basic behaviours in the personal management of email: *no filers* (no use of folders), *frequent filers* (folder users who file messages daily), and *spring cleaners* (folder users who file messages only periodically).

In an interview study of 10 users, Boardman [6] looked at user behaviour in organising and maintaining three separate hierarchies for files, email, and web bookmarks. Five of the ten users attempted

to maintain parallel hierarchies, with varying degrees of success. In a later study, Boardman and Sasse [7] again looked at cross-tool organisational strategies relating to file, email, and web bookmark data. Organising strategies varied significantly between the three types of data: files were the most extensively organised, with deeper hierarchies and fewer unfiled items compared to email and bookmarks.

In a more recent study, Bergman et al [4] surveyed several hundred users in a series of studies of personal computer users and asked them (among other things) to estimate the percentage of their file retrievals performed via search (desktop search), navigation (folders), shortcuts (desktop links), recent documents lists, and other mechanisms. Users strongly preferred naviation through a folder hierarchy (56–68% of retrievals) to search (only 4–15% of retrievals). Users often searched only when they could not remember the location of a file in the folder hierarchy.

1.2 To Classify or Not To Classify

The preceding studies indicate a clear reluctance on the part of users to invest time in advance to file (classify) documents, even if they would then be easier to retrieve later. Why is this? First, it is extremely hard to create category names which are unambiguous. Second, it is hard to find category names which divide up the parent category into mutually exclusive sub-divisions, so categories invariably overlap to some degree. Third, the child categories should completely partition the parent category, so that the user does not feel like a category is missing [23, page 3]. Fourth, information in the real world often falls into several categories. Taking an example from Morville and Rosenfeld [18, page 55], a tomato may be considered to be a vegetable, a fruit, or a berry, depending on the context. Fifth, the classification scheme may become unbalanced, with too many items in one category, and too few in another [24].

Consider the case of filing documents in a hierarchical file system. A file `bobs-ideas-on-XY.txt` contains ideas from Bob about a project XY. Should it be placed in a sub-folder for colleagues `people/bob/` or a sub-folder for projects `projects/XY/`? A decision has to be made. Putting copies into both places will lead to inconsistencies, as soon as any modifications are made.

The file systems of common operating systems such as Windows, OS X, and Linux already provide mechanisms (called shortcuts, aliases, and symbolic links, respectively) to make the same file visible in multiple places in the file hierarchy. Windows shortcuts are, in fact, only special text files rather than a feature of the file system itself. If an application does not know how to process shortcuts, it cannot access the linked information. Microsoft Windows does not actually make use of the available file system level link technologies (NTFS Junction Points, NTFS hard links, and NTFS symbolic links). Moreover, such linking mechanisms seem to be rarely used by users in practice anyway [12]

1.3 Location-Based Spatial Layouts

Several studies have indicated that users like to arrange files by placing icons into groups on the desktop [3]. In effect, spatial location is being used as an aid to memory. Windows Explorer and OS X Finder also support this behaviour by allowing users to position items in its Icon View (and remembers their positions for next time) at any level in the file hierarchy, not just on the desktop. However, spatial layouts ultimately suffer from lack of space. There is simply a limit to the number of items which can be organised effectively in this way.

1.4 Tagging Systems

The basic idea behind tagging has been around for a while: add a few descriptive keywords (tags) to an item so that you can find the item again later by searching for one or more of the keywords. Tagging is cognitively much easier than categorising (classifying), because it only involves users making local conceptual observations [24]. However, tagging also suffers from people using different words or variants to describe the same characteristic [14]. Weinberger [25, page 95] describes the advantages of shared social tagging in communities such as Delicious, but concedes that there will always be ambiguity when tags are assigned by (millions of) different people. Indeed, much current research has focussed on social and collaborative tagging rather than on tagging by individuals in a personal setting. Dourish et al [9] describe the use of tags (called properties) to support the concept of Placeless Documents in a system called Presto.

1.5 Faceted Browsing

Faceted classification was invented by Ranganathan in 1933 [22]. Whereas in tagging users are free to select any words to be tags, in faceted classification a restricted set of words (isolates) are available for use in each of a set of facets to describe the item [25, page 80]. For example, `epicurious.com`, an online recipe web site, characterises its recipes along 8 facets: recipe category (5), dietary consideration (12), cuisine (27), meal/course (12), type of dish (12), season/occasion (18), preparation method (18), and main ingredient (31). The number in parentheses indicates the number of valid tags for that facet.

The retrieval process, faceted browsing, proceeds through progressive refinement. The user can select a value from a first facet (say cuisine = Irish) to receive 92 recipes with that characteristic. Then further selecting, say, main ingredient = beef further restricts the number of matching recipes to just 7. There are no dead-ends in faceted browsing: combinations having 0 matches are not offered to the user. Feldspar [8] is a system which works in a similar way to faceted browsing. Document attributes are progressively refined until the intended document is located.

1.6 Desktop Search

Desktop search engines such Google Desktop and Copernic Desktop Search are increasingly common among average users. A desktop search engine indexes the full text content and various metadata attributes of documents, email, and bookmarks stored in the local file system. Items can be retrieved by typing in appropriate search terms, just like in a web search engine. Desktop search engines are extremely useful to users, but supplement rather than replace tagging systems and folder hierarchies [4].

2. REQUIREMENTS FOR PIM TOOLS

Modern technology offers far more possibilities for users than the mental model of a physical desktop: information in the real world can only occupy one location at the same time. A physical folder has its one physical place. In the digital world, information can be located in many virtual places at the same time [25]. The file from the previous example `bobs-ideas-on-XY.txt` can be made findable both by browsing through projects and also by browsing through people.

The metaphor of the physical desktop, although handy for novice users migrating from a paper-based environment, should no longer be used as the dominant mental model in the virtual desktop environment. Once users are liberated from the limitations of the

classical desktop metaphor, they can experience a variety further benefits of the digital world.

Computer environments today are not like the computer environments of even a decade ago. Hardware has become much more powerful, software has become more capable (and complicated), and much more data is being processed from far more sources. There needs to be a shift of metaphor to meet today's computer environments rather than those of the last century.

Based on the previous discussion, eight fundamental requirements are proposed for future PIM tools, with their main focus on the retrieval process in a local file system. These requirements are not a final nor a complete set of requirements. Additional requirements will be developed as PIM research continues to loosen the limitations of the metaphors previously introduced from the physical world. Some requirements are obvious, but are not implemented well enough in current systems. Some requirements can not yet be found in current systems.

2.1 Be Compatible with Current User Habits

Users are comfortable with their application environment and want to keep it that way. Any new software solution has to integrate into the current environment as smoothly as possible. Any tool which covers only a subset of applications [10] will fail to satisfy a broader user population, because they do not want to be limited to a subset of applications.

Special file browsers were developed to provide more power to the user for the retrieval process [8, 16]. Unfortunately, most of these solutions require a different and sometimes confusing user interface which the typical user might reject. Lack of integration with preinstalled applications is a crucial issue. The user should not be locked-in to a special interface for browsing and searching.

The file system level seems to be a good level to achieve compatibility, since all existing applications share this interface level. Gifford et al [11] and Bloehdorn et al [5] propose promising approaches, although they require special (new) file systems and sometimes special file servers. However, a typical user is unwilling to install a special file system or file server software, particularly if it is not guaranteed to be compatible with their familiar operating system tools. In the long term, future operating systems will have to provide some kind of information retrieval features even in the lower layers of the file system. In the mean time, informations retrieval software solutions have to compensate for the missing support within current file systems.

Many PIM solutions are based on databases [10]. Here again, users are seldom willing to run a specialised file storage database on their computers. They may be unable to make backups with their familiar tools (backup often means simply copying to external storage media), backing up a database is a very different procedure. Users may have to learn a new interface, may be locked-in to the new interface, and the new interface is often poorly integrated into existing applications.

2.2 Minimal Interference

Any new software solution requires some kind of additional user interface. It is essential to keep the learning effort as small as possible. Any interaction step which the users have to make should be absolutely necessary to the process. Optional features should be hidden behind an optional (advanced) interface. In contrast to popular belief, snazzy graphic displays do not automatically result in usable and efficient information retrieval interfaces [13, 16].

2.3 Support Multiple Contexts

A user searching for information always has some kind of mental context. This mental context depends on the current situation and is typically different from the context the user was in when performing the storage process [13, 21]. Good PIM software supports different mental contexts with multiple browsing paths [16].

Considering the file example from the introduction, the user should be able to find the file from Bob about project XY using the people context and/or using the project context. Users want to be able to file information under different categories such as task-related, topic-related, time-related, provenance-related, and form-related [2].

That means that information should be able to be found in multiple places rather than only in one specific location. Tagging seems to be a promising approach [21, 24], although two recent studies comparing tagging with classification reported inconclusive results [14, 20].

2.4 Support Browsing

Studies show that over the years users still prefer browsing over teleporting [4, 2, 8]. When browsing a classification hierarchy, users can see the choices available at each level and choose the most promising.

A great deal of effort was invested into developing improved search engine technology. Although these advances were important and resulted in more capable desktop search engine products, users still prefer browsing in a directory hierarchy to searching with a desktop search engine [4]. Thus it would make sense to invest some future energy and effort into radically improving current hierarchy browsing mechanisms.

2.5 No Unnecessary Limitations

Since large numbers of computer files define our everyday lives, any PIM software solution should scale well to a large number of files and should not affect the efficiency of the browsing process. Even with a large number of files, users must be able to locate their data as quickly and easily as possible.

Some special retrieval tools handle only a small set of file types. Such systems — although very popular in the form of music or photo management software — are not a general solution to the underlying shortcomings of current file browsing tools. Some features provided in specialised management tools would be of great help for other file types.

For example, OS X Finder has the feature of smart folders. A smart folder is a stored search query which dynamically shows any results matching the search criteria. When the user "opens" the smart folder, the content is updated instantly. With this feature, it is very easy, for example, to create a smart folder showing all text files modified within the last two days without having to repeatedly define a search query every time. This is very similar to what iTunes offers for music collections, but other software do not yet offer this feature, say, for image files. Future file browsing solutions should provide enhanced methods for all kinds of file types.

2.6 Transparency

One major aspect of good PIM solutions is transparency to the user. User have built up knowledge of their software environment: a set of experiences, expectations, and standard processes concerning file storage and retrieval. For example, an existing backup process should not be affected by a new PIM system. Users should know where their files are located and what happens to them.

Approaches which require the installation of unfamiliar underlying software introduce complexity and opacity to the system. Users do not trust database systems for metadata or file storage. Ordinary users do not know about database management, database structures, and binary large objects. They do not know how to get their files out of a database system again. Users lose confidence in the software environment, if they are confronted with software that they do not understand.

2.7 Provide for Expiry Dates

Studies show that, with progressively cheaper storage, users tend to keep files over a longer period of time or do not delete them at all [2]. This compounds the information overload problem. There are increasing calls for "forgetting" to be recognised as an important feature in the digital age [17].

During the storage process, users often have an idea of how long the file might be of interest, but this information is forgotten once the file has been stored. Giving the user a method to explicitly define expiry dates — even if they are in the far future — can diminish data overload over time. Providing an expiry date offers the user to explicitly define information as ephemeral, which is an important need as user studies [3] suggest.

In addition, users might be allowed to hook into the process of handling expired data. A user could, for example, automatically move files which are no longer of interest from the "working area" into explicit "archive areas" to remove clutter from current work. Due to the enormous amount of data, users can no longer afford the time to screen all data for archiving.

However, all of this requires an expiry date be attached to the user's information, which in most cases only the user can define.

2.8 Add Metadata While Storing

When a file is stored the user should be given the option to manually add metadata and contextual information to the file. Manual and semi-manual tagging can offer an effective solution for a better retrieval method. Other metadata can (and should) be added automatically, such as a timestamp for time-related retrieval.

Automatically extracted metadata (alone) is often of less use for the purpose of retrieval through browsing. Desktop search engines handle the entire content as metadata and so provide this additional means of access.

Allowing users to explicitly add metadata supports the creation of a user-defined vocabulary (intentionally or subconsciously), which can strongly support subsequent browsing.

3. CONCLUSIONS

The previously proposed set of requirements are intended to spark discussion and serve as a basis for the development of future PIM tools.

Such tools should not seek to radically change user behaviour in one stroke, but rather to bring to pass a gentle evolution. Special interfaces and special software layers requiring additional user interaction are not being accepted by ordinary users.

Modern desktop search engines are a great help to some users, but most users prefer browsing to their files within a hierarchical directory. Thus the browsing process needs to be revisited by PIM researchers and interface developers.

4. REFERENCES

[1] D. Barreau. *Context as a Factor in Personal Information Management Systems*. *Journal of the American Society for Information Science*, 46(5):327–339, June 1995. ISSN 0002-8231. doi:10.1002/(SICI)1097-4571(199506)46:5<327::AID-ASI4>3.0.CO;2-C.

[2] D. Barreau. *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, January 2008. ISSN 1532-2882. doi:10.1002/asi.20752.

[3] D. Barreau and B. A. Nardi. *Finding and Reminding: File Organization from the Desktop*. *SIGCHI Bulletin*, 27(3):39–43, July 1995. ISSN 0736-6906. doi:10.1145/221296.221307. http://www.sigchi.org/bulletin/1995.3/barreau.html.

[4] 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, September 2008. ISSN 1046-8188. doi:10.1145/1402256.1402259.

[5] S. Bloehdorn, O. Görlitz, S. Schenk, and M. Völkel. *TagFS – Tag Semantics for Hierarchical File Systems*. In *Proc. 6th International Conference on Knowledge Management (I-KNOW 06)*, pages 304–312. September 2006. http://triple-i.tugraz.at/blog/wp-content/uploads/2008/11/37_tagfs.pdf.

[6] R. Boardman and M. A. Sasse. *Multiple Hierarchies in User Workspace*. In *Proc. 19th SIGCHI Conference on Human Factors in Computing Systems (CHI 2001) Extended Abstracts*, pages 403–404. ACM, March 2001. doi:10.1145/634067.634304. http://www.iis.ee.ic.ac.uk/~rick/research/pubs/workspace-chi2001.pdf.

[7] R. Boardman and M. A. Sasse. *"Stuff Goes into the Computer and Doesn't Come Out": A Cross-Tool Study of Personal Information Management*. In *Proc. 22nd SIGCHI Conference on Human Factors in Computing Systems (CHI 2004)*, pages 583–590. ACM, April 2004. doi:10.1145/985692.985766. http://www.iis.ee.ic.ac.uk/~rick/research/pubs/boardman-chi04.pdf.

[8] D. H. Chau, B. Myers, and A. Faulring. *What to do When Search Fails: Finding Information by Association*. In *Proc. 26th SIGCHI Conference on Human Factors in Computing Systems (CHI 2008)*, pages 999–1008. ACM, April 2008. doi:10.1145/1357054.1357208. http://www.cs.cmu.edu/~dchau/feldspar/feldspar-chi08.pdf.

[9] P. Dourish, W. K. Edwards, A. LaMarca, and M. Salisbury. *Using Properties for Uniform Interaction in the Presto Document System*. In *Proc. 12th Annual ACM Symposium on User Interface Software and Technology (UIST'99)*, pages 55–64. ACM, November 1999.

doi:10.1145/320719.322583. http://www2.parc.com/csl/projects/placeless/papers/uist99-presto.pdf.

[10] J. Gemmell, G. Bell, and R. Lueder. *MyLifeBits: a personal database for everything*. *Communications of the ACM*, 49(1):88–95, January 2006. ISSN 0001-0782. doi:10.1145/1107458.1107460. http://research.microsoft.com/pubs/64157/tr-2006-23.pdf.

[11] 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, October 1991. doi:10.1145/121132.121138. http://cgs.csail.mit.edu/history/publications/Papers/sfs.ps.

[12] D. J. Gonçalves 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)*, pages 46–60. Springer LNCS 2844, June 2003. doi:10.1007/b13960. http://virtual.inesc.pt/dsvis03/papers/05.pdf.

[13] M. W. Lansdale. *The Psychology of Personal Information Management*. *Applied Ergonomics*, 19(1):55–66, March 1988. ISSN 0003-6870. doi:10.1016/0003-6870(88)90199-8. http://simson.net/ref/1988/Lansdale88.pdf.

[14] 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, April 2009. doi:10.1145/1520340.1520565.

[15] T. W. Malone. *How do People Organize Their Desks?: Implications for the Design of Office Information Systems*. *Transactions on Information Systems*, 1(1):99–112, January 1983. ISSN 1046-8188. doi:10.1145/357423.357430.

[16] G. Marsden and D. E. Cairns. *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)*, pages 122–129. South African Institute for Computer Scientists and Information Technologists (SAICSIT), September 2003. 1581137745. http://pubs.cs.uct.ac.za/archive/00000190/01/saicsit2003-dec.pdf.

[17] V. Mayer-Schönberger. *Delete: The Virtue of Forgetting in the Digital Age*. Princeton University Press, October 2009. 978-0691138619.

[18] P. Morville and L. Rosenfeld. *Information Architecture for the World Wide Web*. O'Reilly, Third edition, November 2006. 0596527349.

[19] B. A. Nardi, K. Anderson, and T. Erickson. *Filing and Finding Computer Files*. Technical Report 118, Apple Computer, Department of Computer Science, 1994.

[20] 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. http://business.clemson.edu/Catlab/pubs/pak-pautz-iden-2007.pdf.

[21] D. Quan, K. Bakshi, D. Huynh, and D. R. Karger. *User Interfaces for Supporting Multiple Categorization*. In *Proc. 9th IFIP TC13 International Conference on Human-Computer Interaction (INTERACT '03)*, pages 228–235. IOS Press, September 2003. 1586033638. http://www.idemployee.id.tue.nl/g.w.m.rauterberg/conferences/INTERACT2003/INTERACT2003-p228.pdf.

[22] S. R. Ranganathan. *Colon Classification*. The Madras Library Association, 1933.

[23] E. M. Rasiel and P. N. Friga. *The McKinsey Mind*. McGraw-Hill, September 2001. 0071374299.

[24] R. Sinha. *A Cognitive Analysis of Tagging*. Rashmi's Blog, September 2005. http://rashmisinha.com/2005/09/27/a-cognitive-analysis-of-tagging/.

[25] D. Weinberger. *Everything Is Miscellaneous: The Power of the New Digital Disorder*. Times Books, May 2007. 0805080430. http://www.everythingismiscellaneous.com/.

[26] S. Whittaker and C. Sidner. *Email Overload: Exploring Personal Information Management of Email*. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI 1996)*, pages 276–283. ACM, April 1996. doi:10.1145/238386.238530. http://dis.shef.ac.uk/stevewhittaker/emlch96.pdf.