Towards a unification & integration of PIM support

William Jones, David Karger, Ofer Bergman, Mike Franklin, Wanda Pratt, Marcia Bates

Information fragmentation is a pervasive problem which is felt in several stages of personal information management (PIM). As the example in the introduction to this special issue on PIM illustrates, even a seemingly simple decision, such as whether to say "yes" to an invitation, often depends upon a number of different kinds of information – information from a calendar, from a paper flyer, from web sites, from a previous email conversation, etc. Information can be fragmented by physical location. This is nothing new. Now information is often fragmented by the very tools that have been designed to help us manage our information. Our information may be scattered across various computers and gadgets. Some information, for example, may be on a laptop computer we use at home, other information may be on a desktop computer we use at work and one or more PDA or smart phones. Even on a single computer, our information is scattered across the computer desktop, "My Document", file folders, email folders, collections of bookmarks, etc. New applications introduce still more forms of organization with little or no integration to previous forms. People can rightly complain that they have too many organizations.

Information fragmentation creates problems not only in the maintenance of several organizations but also in everyday PIM actions such as keeping and finding. We may sometimes need to look in several places, physical and virtual, in order to gather together the information we need for a particular task. We may also be less certain where and how to keep newly encountered information (see breakout session # 3). Or do we "have it" already? If we keep the information again anyway ("just in case") we may then face some serious problems with consistency and updating later on.

Even when information is not directly copied, today's applications often force us to repeat the same data in several places. For example, a name such as "Jill Johnson" might appear in an address book, and also as the creator of a photograph in a digital photo album. Changes to one version of the data (a new married name, for example) often do not propagate to other versions of the data. Also, we may experience the frustration of having some operations – name resolution, for example – available in one place (when sending email) but not in another (when working with photographs). Finally, there is no easy way to "link" together the various bits and pieces of data relating to "Jill Johnson". In some cases, we may need to perform a difficult search in order to access another representation of information we are already looking at!

If the computer has been an unintended agent of information fragmentation, it can also be used to help us "put the pieces together" again. This article provides a sampling of some of the ways in which our personal information might be better integrated. The article concludes with a look at three research prototypes that illustrate varieties of approach to the fundamental challenge of personal information integration.

As research in the previous section makes clear, information fragmentation creates problems for keeping, finding and the various m-level activities. The obvious antidote to fragmentation is integration (or unification). This section considers some approaches to integration. Each approach is illustrated by a brief discussion of an ongoing research project and possibly a prototype tool. This chapter is not a review of PIM tools. Before approaches are discussed, it is useful to consider of the kinds of integration that apply to personal information:

Variations and levels of integration

Integration across physical location. Perhaps the most basic kind of integration is the integration of information from many physically distinct sources. It is a significant burden to move physically from location to location to get the information we need especially when these are separated by some distance. Computing technology, in several ways, has already done a great deal to integrate information across physical location. Data transfer protocols such as the File Transfer Protocol (FTP) long existed to bring information from where it is stored to where we, as users, need it. More recently, tools such as network file systems (NFS) and the Web free us from even having to think about the physical location of the desired information.

As the capacity and portability of storage devices continues to increase we can now bring with us a substantial proportion of the information we use regularly – on a laptop, for example, or even on much smaller device such as a “smart phone”. We can access still more information via a wireless connection. Moreover, we now have access to many new kinds of information in digital form – text, of course, but also pictures, music and even full-motion video. Computing technologies already combine to enable a high degree of integration with respect to physical location.

For information in digital form, then, we have reason to hope that physical location will be come less and less a factor in information fragmentation. For digital information, several additional integrations remain to be accomplished:

Information in the means of access and organization. As noted in the previous section, desktop search utilities already provide one integrative means of information access. However, studies continue to show that people have a strong preference for browsing or “orienteering” styles of access to their information [2, 33, 37]. People use search only after these preferred methods of access fail. There are more basis reasons to organize information – we understand the information better. People often have, and complain about having, several distant organizations of information – usually folder hierarchies Integration means providing at least an option to bring these organizations together. Some people may still find it useful to have distinct organizations for email, e-documents and web references. But this would be a choice freely made, not a separation imposed by supporting applications.

Integration by the grouping and inter-relating of items. It is often useful to group and inter-relate items – to each other and to the tasks for which they are needed. We might, for example, want to inter-relate all the information for a particular person in our lives. Frequently, we group together information relating to a particular task we wish to complete. For example, we might group together information concerning hotels in a city in order to select a hotel for our stay in that city.

Traditional folders provide one means for grouping information together. We also bring together information in a more ad hoc way through the windows we open on our computer display as we try to complete a task. A collection of a kind may be implicit in the trail of information items we have viewed. However, the trail does not persist and open windows add to our clutter and obfuscate as well as group our information.

Research has explored the more general and flexible notion of a *collection* [10, 11, 25]. Items can be manually assigned to a collection (e.g. files placed into a folder) but items (or at least suggested items) for a collection can also be generated based upon a match between items and a "definition" (e.g. a query) for the collection. Limited support for the automated creation of collections is available now via features such as Microsoft Outlook's "Search Folders". Variations on this are now expected in new releases of both the Macintosh and Microsoft Windows operation systems [16].

It may also be useful to assign properties to a collection as a whole. For example, if a collection of information relates to a task ("Find a hotel"), then it may be useful to assign task-like properties like "remind by" and "due by" [22] which might then appear as appointments in an electronic calendar or trigger a reminder (via pop-up or email message) later on.

Integrative views of information. A fourth kind of integration of digital information takes us inside a collection or grouping of information (however defined or created). We seek to "view" the items within a collection. We look for recurring patterns among and important connections between information the items in view. For paper documents, the desktop and other flat surfaces of an office traditionally serve as a view space. We may move paper documents from filing cabinets to the desktop in order to "see" the information better. Computers provide several alternatives for comparable viewings of digital information including the computer desktop, a folder listing of files (or email messages or web references) and the window displays of opened documents, email messages, web sites. Our view of items can act as a powerful extension to our limited internal working memory for information [29].

Unfortunately, as we attempt to arrange information on a computer display, we experience problems. For example, applications involved in rendering the items of a collection may each consume a large part of the display. Documents, email messages, web pages, etc. may each "live" in a large window with attendant menus, toolbars, jumping-off points and default presentations.

A computer display may be filled with windows, often obscuring each other and each competing for our attention. We can experience similar problems with the computer desktop and, of course, and with the top of a physical desk. The information we lay out in order to see and understand can turn into a jumble that actually impedes our ability to work effectively. Worse, a carefully arranged configuration of windows is typically not preserved between restarts of the computer.

Current computing support for the creation of more workable, integrative views of information is quite limited. There has been little progress in file managers, for example, beyond the standard icon, list (possibly with properties) and thumbnail views.

Integrative facilities of data manipulation. In a fifth kind of integration, we move from "read" to "write" access. For example, we may want to give explicit, external representation to the patterns we notice, the connections we make and main points we note for information that we are viewing. Or we may want to transfer information from one application to another. A basic facility of data manipulation that we use repeatedly in a typical day is the copy/cut-and-paste facility (and the drag-and-drop facility). The cut-and-paste facility provides an intuitive way of moving data from one application to another – although in some cases, the transfer is still text-only.

Other facilities for manipulating data are still provided in a very fragmented, piecemeal fashion. For example, in ways that are analogous to those we use when marking up a paper document, it is possible to highlight and annotate selected text for a document in Microsoft Word. Similar, but not identical, operations can be made on "PDF" documents displayed in Adobe Acrobat. However, it is not possible to perform comparable operations on the selected text of an email message displayed in Microsoft Outlook nor is it possible to highlight the selected text of a Web page as displayed in most web browsers. Even the basic ability to impose an ordering on information items is unsupported (e.g., for email messages) or accomplished only by a clever use of leading characters [23].

Integration with the current context. Associations to various aspects of the current context are also a potential basis of unification. The time of our last interaction with a document (email message, web page) is recorded currently. But many other aspects of the interactive context are not [27]. We may recall that, when we last viewed an information item at home, not at work, and the weather was warm and sunny outside. But these recollections provide little help in our interactions with the computer. As we create a new email message or e-document or as we browse to a web site, we may have a particular task in mind, but there is very little support for communicating this task to the computer. Worse, newly created documents are often placed, by default, in a place like "My Documents". In general, the context we "share" with the computer in our interactions with information items is very limited.

Other integrations are also relevant to PIM. Users may have considerable time and energy invested in existing folder hierarchies and other organizations. Moreover, these organizations and supporting applications are used in many ways that are not well understood [23]. Consequently, a new tool has a better chance of success if it is able to build upon these organizations and extend the functionality of existing applications rather than forcing a leap to an entirely new way of doing things.

This discussion is intended neither to be a definitive nor exhaustive treatment of the ways in which we might like to see a greater integration of our information. However, the discussion should provide a sense for the many facets of a general term like "integration" as applied to PIM. These facets (and others that may occur to the reader) can be used as a basis for comparing approaches to the integration of personal information that are discussed in the remainder of this section

Existing Unification Examples

Given the value of data unification, it is unsurprising that some unification mechanisms already exist. In this section we outline some of the most common unifying tools in use at present. This lets us demonstrate how some of the unification benefits discussed above are achieved, and also lays the groundwork for our discussion below of the different ways unification can be accomplished.

Text

A significant portion of the data managed by many applications is text. It may come formatted in many different ways: within HTML documents on the web, in bulleted lists in Microsoft Powerpoint, formatted in a word processing document, or typeset in Adobe PDF. But strip away the formatting, and one is left with a sequence of characters. At present, most applications are able to offer text to and accept text from other applications—typically with the assistance of a clipboard application with which they all communicate. The quotes offered earlier in this article could 2 therefore be “cut” from Adobe reader and “pasted” into an Emacs buffer. The typical linear flow of text means that even interfaces for those operations are somewhat standardized—mouse drags from the beginning to the end of the text to be cut, and click within the text to indicate the “insertion point” of text to be pasted. The common model of text also means that most applications can contribute some (stripped down) representation of their data to a *text search engine.* Such an approach gives the end user a well-understood framework for searching by content over all of their data, independent of which application owns any given piece. Significant buzz has developed around the recent set of “desktop search engines” such as Google desktop and MSN desktop, that offer this capability. This support is not universal—had the PDF source document instead been postscript, which does not provide a text model, it would have been necessary to manually retype the quotes instead of cutting and pasting them. Another unfortunate omission is error messages: it is often impossible, when a dialog box pops up with an error message, to cut and paste its text into an email to technical support.

The File System

The file system has been another, highly successful data unifier. At the most basic level, the file system offers yet another standard data format: the file. All applications can make use of the notion of a sequence of bits that can be read from, copied, or written to (although of course, writing bits into a file whose meaning is not known will serve only to corrupt that file). An email application can “attach” a file for delivery to another computer without any understanding of which application created the file or what its contents mean. The file system also offers a unified interface for the end user. Using the now-standard model of hierarchical directories or folders, a user can organize files according to whatever principle they find useful. For example, a user may gather into a single directory all the files necessary for accomplishing a particular task, regardless of which applications manage those files. Working inside that directory gives the user immediate access to all of those files. The file system lets users name individual files and list the names of files in a directory, an important aid to organizing and searching. Most applications let users specify files to open or save. On the downside, the file system assumes so little about the meaning of the files it holds that any significant manipulation of any file requires launching an appropriate application. This will take a user away from the directory view of all files relevant to a given task, and back to an application view that shows only some of the information they want to work with. Additionally, files are typically large. An address book, for example, will usually be stored as a single file, rather than as one file per entry. This precludes using the file system for fine-grained organization. A user will not be able to put, into a directory aimed at writing a particular paper, the address-book entries of all his co-authors. He can certainly place a link to the address book, but from within that directory he would need to launch the address book and then go through an address-book specific process to find the co-authors in the address book.

The Desktop and Window Manager

Just as the file system offers a lowest-common-denominator unifier for representing all applications' data, the desktop and window manager ofter a lowest-common-denominator unifier for viewing all files and applications together. Nowadays all applications run inside one or several windows, and delegate to the window manager the control over those windows being stacked, tiled, moved, resized, opened, and closed. Thanks to the window manager, a user can simultaneously view and manipulate all the information objects they care about, in multiple application windows laid out side by side on the display. On the downside, it often seems that each application wants the entire display to itself. To get at the data managed by an application, users typically need to launch the entire application, which lives in a large window with its attendant menus, toolbars, jumping-off points, and default presentations. Because the window manager treats the application opaquely as a rectangle full of pixels, it cannot select out the one piece of the display that the user actually cares about. A common consequences is *window clutter*—a desktop filled with tens of windows, all obscuring each other, each bearing a small fragment of information that a user cares about. To get at it, users must continuously locate and rearrange windows to find the fragments they need. One often sees significant effort invested in laying out windows 3 just right, so that the desired information in one window obscures only unimportant information in another window. But this requires a significant investment of effort, is unlikely to remain the "right" layout as the user continues to manipulate the information, will be lost when the applications are exited, and is often simply impossible due to the topological constraints imposed by the location of the key information in each window. Even if users can get the window layout exactly right, this only gives them a convenient *view* of the information. Since each window is managed by a different application, there is still no guarantee that the user will be able to effectively manipulate information across the application boundaries. The fact that one can display a person in an address book and, at the same time, display a photograph of that person in a photo album offers no guarantee that one will be able to use either application to associate the person with their photo. In addition to managing windows, the desktop has also become an important tool for data organization. Files are represented on the desktop by named icons, and users choose a physical layout of those icons that keeps related icons near each other or in regions of the screen that have specific meanings for the user.

The World Wide Web.

Much of the impact of the World Wide Web can be attributed to its success in unifying a tremendous range of information. The web rides on the concept of the universal (we might say unified) resource locator, or URL: every web page has one, and any web page can use one to link to any other web page. As the web grew, many different entities—institutions, people, books, recipes, and so on—became represented as HTML documents on the web, linked to each other with no restrictions as to type. The uniform representation as HTML meant that a single tool, the web browser, could present all the different information types. Also, directories and search engines could be built to navigate the entire web, years before anything similar was attempted for the end-user desktop. With this appealing unification already present, we can ask why it has not been adopted as the primary environment for personal information management. Hypothetically, a user could create a separate web page for each email message, each directory, each music file, each calendar appointment, each individual in their address book, and so on. Editing these pages, the user could indicate arbitrary relationships between their information objects. Feeding these web pages to a tool like Google would give users powerful search capabilities; combining them with the orienteering opportunities offered by the user-created links would surely enhance users' ability to locate information. In response, we observe that HTML is quite an impoverished data representation, offering little more than a rich formatting of text and images. Users' rich, structured, personal information spaces can certainly be "boiled down" to HTML for presentation, just as they can be boiled down to pixels, but such a representation loses much of the semantics of the underlying data. While an application could hypothetically store its richly structured information in HTML, it would have to select a canonical schema for representing it, and would not be able to cope with a user or another application making modifications to other valid HTML that does not obey the canonical schema. At the interface level, while the web browser is certainly a good tool for *browsing* data, it is often too limited a tool for *manipulating* data. Web forms are useful only for relatively primitive data entry, not the sophisticated manipulations offered by applications' carefully specialized interfaces and operations.

Towards greater integration and unification

Integration through email

The uses of email now extend well beyond its original use to send text messages between people separated from each other by time and distance. For example, email is now used for task management, personal archiving and contact management [4, 13, 30, 39]. Many of us might say that we practically "live" in email in a typical work day. (On the other hand, many of us may also go "offline" in order to do concentrated work without the constant interruption of email). One approach to current problems of PIM – in particular, the fragmentation of information by application – is to declare email the winner and to build additional PIM functionality into an expanded email application..

This approach is exemplified by *Taskmaster* [4]a prototype that deliberately builds task management features into an email client application. Taskmaster introduces support for "thrasks" as a way to automatically connect together task-related email messages based upon an analysis of message content. The thrask is intended to be an improvement on "threads". Email discussion within a thread can diverge widely from an original task while, at the same time, as other task-related email messages are sent outside the context of a thread. A thrask can also include links (e.g., web references) and documents that relate to the task. In this way several forms of information are brought together.

Following an "equality of content" principle, Taskmaster also displays attachments (links and documents) at the same level as the email messages associated with their delivery.

Attachments are no longer buried within the email message. This makes it easier for the user to see and access all information related to a task, regardless of its form. Email messages and associated content can be sorted and grouped by thrask but otherwise remain in the inbox until moved by the user. Users can also fine-tune by changing the thrask associated with an email message. The design intent is that Taskmaster adds new task-related functionality without taking away the functionality the user is already familiar with. Taskmaster provides several means of view thrask related email messages and also supports the assignment of task-relevant properties. One potential limitation of the "integration through email" approach was alluded to above – people may want to spend *less*, not more, time in email. Also, adding additional facilities relating to task management and other PIM-related activities may increase the complexity of an email application that is already quite complex for many users. In addition, users are likely to have other reasons for continuing to use files in the file system – better backup, for example, or better, finer-grained control over access rights and security.

Integration through search

Desktop search facilities that can search across different forms of information – especially files, email and web pages that a person has visited – have a tremendous potential to support a more integrative access to information. Some of this potential has already been realized in available facilities such as Google Desktop.[1]

Fast, integrative cross-form searches are also supported in the Spotlight features of the Macintosh operating system X (Mac OS X)[2]. Spotlight also includes support for persistent searches and the related notion that "smart folders" can be populated and constantly updated to include the results returned for an associated query. Similar features are also planned for inclusion in the next major release of Microsoft windows (code-named "Longhorn")[3].

Microsoft's *Stuff I've Seen* (SIS) project is exploring additional integrations that build upon a basic ability to search quickly through the content and associated properties for the information items of a PSI. The UI for SIS supports sorting of returned results on several properties including a "useful date" (with a definition that varies slightly depending on the information form). Time intervals can be further bracketed in a *Memory Landmarks* add-on through the inclusion representations for memory events, both public and personal. An *Implicit Query* (*IQ*) add-on to SIS is a further step in integration. As a user is viewing an email message, content and properties associated with the message are used to form a query. Matching results are shown in a side panel. The panel may sometimes list useful information items that are in the user's PSI but have been completely forgotten.

These and other search features make it clear that search is about more than typing a few words into a text box and waiting for a list of results. We return to a question posed earlier:

---

[1] For a more complete review of desktop search engines available now for use see Answers.com (http://www.answers.com/ and then search, of course, for "desktop search").

[2] See http://www.apple.com/macosx/features/spotlight/.

[3] Spanbauer, S. (2005, August 2005). Longhorn Preview: The newest versions of the next Windows add graphics sizzle and more search features but lack visible productivity enhancements. *PC World*.

Will the constellation of features enabled by fast, indexed search of content *and all associated properties* for information items in a PSI eventually eliminate the need for many PIM activities? In particular, does the need to actively keep information and to maintain and organize this information largely go away? Can people leave their information "flat" so that the need for conventional folders disappears?

There are two very different reasons for believing that the answer is "no":

1. A search can return many versions of the needed information. People create multiple versions of a document, for example, in order to represent important variations or to "freeze" a document at key points in its composition or, simply, because they need to use this document on different projects, in different contexts (and it's easier to copy than to reference). People may also save external items into their PSI several different times because they can't recall whether they have done so before or, again, because they want to access this item in different "places". Or people may receive several different variations of information in email. Airlines, for example, sometimes email several different variations of an on-line, e-ticket confirmation.

   When multiple versions are returned, considerable time may be spent deciding which version is correct or deciding which collection of items provides the necessary information. The problem of multiple versions becomes worse when people modify or correct a document or they save a new version of an item without tracking down and removing all the old versions. A CEO of a major financial services company told the author that he had recently spent over an hour trying to decide which of several versions of a PowerPoint presentation was the right one to use and modify for an upcoming meeting with a customer.

2. The second reason to believe keeping and organizing will remain essential PIM activities is more speculative but also more basic: The act of keeping an item, the act of organizing a collection of items, may be essential to our understanding of this information and our memory for it later on. If filing is cognitively difficult it is also cognitively engaging. Filing, as an act of classification, may cause people to consider aspects of an item they might otherwise not notice. People may forget to search for information later if they don't make some initial effort to understand the information. Folders, properties and other constructs can be seen as an aid in understanding information. Even if a tool like Implicit Query is wonderfully successful at retrieving relevant information anyway, people may fail to recognize this information or its relevance to a current need.

   In a better world, we might hope realize the advantages associated with the current use of folders and other means of external representation without suffering their disadvantages. The penalty associated with misfiling currently, for example, is too severe. We may, for all practical purposes, lose this information until it is too late to use it. If folders become more "transparent" or more like tags, we might be more inclined to reference than to copy and more inclined to tag an item in several ways to represent different anticipated uses. We might still be able to search or sort through items as part of a larger set.

   In this regard, improving desktop search facilities may have a paradoxical effect. With search, the cost of misfiling goes down. Even if an item is misfiled, it can still be found again later using search if necessary. Moreover, regardless of folder location, search can be used to construct a useful set of results that can then be quickly sorted by time and other useful properties. Is it possible that people may sometimes be more inclined to file?

Integration through projects

It might be argued that information management and task or project management are two sides of the same coin. It certainly makes sense to try to organize information according to anticipated use and people are observed to do this [26]. ROOMS [19] represents an early attempt to integrate information items and other resources (tools, applications) with respect to a user activity. For example, a user could set up a "room" for a programming project in which each window provided a view into a project- related resource. A task-based approach to integration, TaskMaster, has already been discussed in the context of extensions to an email application.

Another approach in tool support advances the notion of a "project" as a basis for the integration of personal information. When a distinction is drawn between tasks and projects it is typically with respect to length and complexity. In HCI studies of task management [3, 8], for example a task is typically something we might put on a "to-do" list. "Check email", "send mom flowers for Mother's Day", "return Mary's phone call", or "make plane reservations". With respect to everyday planning, tasks are atomic. A task such as "make plane reservations" can certainly be decomposed into smaller actions – "get travel agent's phone number", "pick up phone", "check schedule", etc. – but there is little utility in doing so. In these studies, therefore, the focus is on a management *between* tasks including handling interruptions, switching tasks and resuming an interrupted task.

A project, by contrast, can last for several days to several years and is made up of any number of tasks and sub-projects. Again, the informal "to-do" measure is useful: While it makes sense to put tasks like "Call the real estate broker" or "Call our financial planner" on a to-do list, it makes little sense to place a containing project like "Buy a new house" or "Plan for our child's college education" into the same list (except perhaps as an exhortation to "get moving!").

In the UMEA prototype [24] uses the notion of a current project to bring together various forms of information – electronic documents, email-messages, web references – and associated resources (applications, tools). A design goal of UMEA is to minimize the user costs in setting up a project by automatically labeling items as these are accessed. Unfortunately, UMEA depends upon the user to signal a change in current project. Since users frequently forget to do this, projects are frequently associated with the wrong project. Users can go back later and edit project/item associations to correct for mislabeling but, understandably, users in an evaluation are not likely to do this. Kapetlinin sketches possible ways in which the system might detect a change in project but, to the author's knowledge, nothing along these lines has yet been implemented that can do this with any degree of accuracy. Another limitation of UMEA is that the project is essentially just a label and has no internal structure

Another approach in integration through projects is to have labeling of items occur as an incidental part of an activity that people might do in any case. People plan projects. Some of this plan finds external expression in, for example, to-do lists or outlines. The Universal Labeler (UL) prototype [22], described earlier, encourages users to develop a project plan using a Project Planner module. The Planner provides a rich-text overview for any selected folder hierarchy which looks much like the outline view of Microsoft Word. A hierarchy of folders appears as a hierarchy of headings and subheadings. The view enables users to work with a folder hierarchy just as they would work with an outline. As headings are added, moved, renamed or deleted, corresponding changes are made to the folder hierarchy. The goal is that the Planner is simply another view into the file folder hierarchy and is, in fact, integrated into the file manager. But, as part of UL's general support for shortcuts, folders for a project plan can be used to reference project-related email messages and web pages as well as files. The Planner also provides document-like features not available in a standard file manager:

- **Support for a "drag & link"** action of excerpting. It is possible to select text of interest, drag (or copy) and then drop (or paste) into a project plan. A link to the source of the drag (or copy) is automatically created. Often we are mainly interested in only a small part – a name, number or phrase – of the email message, web page, or e-document that we are reading. But we might like to return to the rest of the information item later on.
- **Support for "create & link"**. A link to the newly created information item (e-document, email message) is automatically created at the insertion point in a project plan.
- **Ordering of elements.** Users can order headings, subheadings and links of a project plan however they like. People depend on ordering as a way to establish priorities and to direct their attention to "first things first".
- **Notes** (annotations). Users can include notes just as they would in a document. Notes can, for example, be used to provide clarification for an associated heading.

Behind the scenes, the Planner is able to support its more document-like outline view by distributing XML fragments as hidden files, one per file folder, which contain information concerning notes, links and ordering for the folder. The Planner assembles fragments on demand to present a coherent project plan view including notes, excerpts, links and an ordering of subfolders (and sub-subfolders). The architecture allows for other views as well. Efforts are currently underway, for example, to support a "mind map" view [38].

A general goal of the UL is to make it easy for users to give external expression (e.g., in a to-do list, a conventional outline, a table, or a "mind map") to their understanding of a project they are working on. This representation is then used as a means of organizing the information needed to complete the project. Also, in support of project management, properties like importance, due date or "remind by" date can be associated to elements in the representation to further leverage that work that went into its creation.

Of course not all of our information directly relates to a project we are working on (or plan to work on). We keep information for a variety of reasons. Some items are more properly regarded as reference material that may have repeated use, whether or not in projects. For example, we may keep articles, music, recipies, quotes, and images with no specific project in mind. It may be more effective to organize reference items such as these by properties, perhaps using a faceted classification scheme.

Integration through properties

Dourish and Edwards argue that the folder hierarchy is limited, antiquated and should be abandoned outright in favor of a property-based system of filing and retrieval as in their

PRESTO/Placeless Documents prototype [10-12]. Such proposals are not new. Ranganathan's colon or faceted classification scheme [35] is essentially an organization of information by a set of properties where items value assignment for one property can vary independently of its value assignment for another property. Recipes, for example, might be organized by properties such as "preparation time", "season", "region or style", etc.

However, an organization of information by properties depends upon an understanding of the information so organized. Meaningful, distinguishing, useful properties for special collections like recipes may be readily apparent but much less so for newly acquired information. Information for a project may more readily organize into the hierarchy of a problem decomposition.

Integration through time. One property of clear relevance across most items is time (as in "time of encounter" or "last accessed"). Several projects and prototypes are motivated by the integrative power of time as a means to organize information.

The MEMOIRS system [28] organizes information items in a sequence of events (which can also include meetings, deadlines, etc.) Perhaps best known of the time-based approaches to information integration is LifeStreams [14, 15]. In LifeStreams, documents and other information items and memorable events in a person's life are all placed in a single time-ordered "stream".

LifeStreams also permits users to place items into the future portion of the stream a point where a need for these items is anticipated. But it is with respect to the future that the LifeStreams timeline metaphor begins to wear. There are some future events are that are "fixed" (to the best of our ability to fix anything in the future) – meetings for example. It makes sense to place a presentation or report that is needed for a meeting at a point in the stream's future to coincide with the meeting. However, in many other cases we have no clear notion of when we will need an item in the future or when we will have an opportunity to use it. In these cases it may make more sense to organize items according to a need (goal, task, project) for which it may have use. Needs, in turn, are often organized into a hierarchy.

Integration through a common underlying representation

The digital information items discussed in this chapter – in particular the file – are high-level. The operations we can perform at the file level are useful, but limited. We can create, move, re-name, and delete files. The data with in a file is typically in a "native format" and readable only by a single application – the word processor, spreadsheet, presentation software, etc., used to create the file. In this circumstance, opportunities to share, consolidate and normalize data (e.g., to avoid problems with updating) are extremely limited. The user can initiate a transfer of data in a file to another file ("owned" by another software application) via mechanisms like "copy & paste" and "drag & drop". But this transfer is often little more than an interchange of formatted text. Information concerning the structure and semantics of the data stays behind in the source application. Moreover, the data is copied, not referenced, which can lead to many problems of updating later on.

As a result, data concerning a person we know – say Jill Johnson – may appear in many, many places in our PSI. This is another variation of fragmentation. Because of this fragmentation, even simple operations, like correcting for a spelling mistake in Jill's name or updating for a change in her email address, become nearly impossible to complete. We may update some of copies but not all. Also, we may experience the frustration of having some operations – name resolution, for example – available in one place (when sending email) but not in another (when working with photographs). Underlying these problems is a problem that there is no concept or "object" for a "person named 'Jill Johnson'" in the PSI and no means by which data associated with this person can be referenced, not copied, for multiple uses (as managed through various software applications).

The situation may improve with increasing support for standards associated with the Semantic Web [5] including XML (eXtensible Markup Language), RDF (Resource Description Framework) and the URI (Uniform Resource Identifier). RDF and XML, for example, can be used to include more semantics with a data interchange. URIs might be used to address data, in place, so that it doesn't need to be copied in the first place (thus avoiding problems with updating information about Jill Johnson, for example). Support for these standards may make it possible, in some future day of PIM, to work with information and data packaged around concepts such as "Jill Johnson" rather than with files. Data for Jill would be mostly referenced; not copied. We could readily add more information about Jill or to make a comment like "she's a true friend". And we could group information about Jill together, as needed, with other information. We could, for example, create a list of email addresses and phone numbers for "true friends" we would like to invite to our birthday celebration.

These and other possibilities are explored in the Haystack project [1, 9, 20, 34]. Haystack represents an effort to provide a unified data environment in which it is possible to group, annotate and reference or link information at smaller and more meaningful units than the file. In the Haystack data model, a typical file will be disassembled into many individual information objects represented in RDF. Objects can be stored in a database or in XML files. When an object is rendered for display in the user interface, a connection is kept to the object's underlying representation. Consequently, the user can click on "anything" in view and navigate to get more information about the associated object (e.g., to get Jill Johnson's birth date, for example) and also to make additions or corrections to this information.

Haystack creates a potential to explore, group and work with information many ways that are not possible when information is "hidden" behind files. However, many issues must be addressed before the Haystack vision is realized in commercial systems. For example, the use of RDF, whether via XML files or a database, is slow. Beyond performance improvements, a great deal must be done if application developers are eventually to abandon the control they currently have with data in native format in favor of a system where data comes, instead, from an external source as RDF.

Integration through a digital recording of "everything"

If a sequence of information events are recorded – the viewing of a web page, for example – there is the possibility later to facilitate not only retrieval of the web page again but also to retrieve other items there were in close temporal proximity to this item. We might hope, for example, be able to access “the email message I was looking at right before I looked at this web page”.

If enough events in our daily life are recorded, we might get significantly closer to a situation where virtually anything recall about a desired item – the contexts of our interaction with the item as well as its content – might provide an access route back to the item. For example, we might query our computer to “go back to the web site that Mary showed me last week”.

In his article “As We May Think,” Vannevar Bush described a vision of a personal storage system, a memex, which could include snapshots of the a person’s world taken from a walnut-sized head-mounted camera and a voice recorder [6]. Bush’s vision has been realized and extended in wearable devices that can record continuous video and sound [7, 32, 36]

A bigger question is what to with all this data once recorded? *MyLifeBits* [17, 18] is an exploratory project aimed at finding out. The project tries to “digitize the life” of computer pioneer Gordon Bell.

The study of “record everything” approaches, also called “digital memories” is becoming a very active area of research [31]. For example, workshops on Continuous Archival and Retrieval of Personal Experience (CARPE) have been sponsored by the Association for Computing Machinery (ACM) in both 2004 and 2005.

A continuous recording of our life’s experiences has many potential uses. For example, might use to refresh our internal memories concerning a meeting. It might be useful in some cases to support our version of events later on. Or we might like to review our digital recording in an effort to learn from our mistakes. Sometimes, we might review just for fun. But clearly, digital memories raise serious concerns of privacy and security which can only be partially addressed by technology alone.

Integration through organizing techniques and strategies

Approaches to integration are predominantly tool-based as inspired by developments in technology. But a degree of integration can also be accomplished through techniques and strategies of integration that make use of existing tool support. It has already been noted, for example, that people are sometimes observed to focus on a single form of information and the development of organizing structures for this form. Other forms of information are “squeezed” into this organization. Everything is printed out, for example. Or everything is sent as an email. Or everything becomes a file.

Some people are observed to create, instead, a single organizing schema which is then applied to different forms of information. The observation has prompted Jones [21] to speculate on the possible value of a *Personal Unifying Taxonomy (PUT).* A person’s PUT would be developed after a review, as guided by a trained interviewer, of organizations for email, e-documents, paper documents, web references and other forms of information. Top-level elements in a PUT would represent areas with enduring significance in a person’s life (high-level goals, important roles). A PUT would also represent recurring themes in the folders and other constructs of different organizations.

However, a great deal of work remains to be done to determine a process and principles of PUT development and to determine whether a PUT can be maintained over time to realize benefits that compensate for its costs of creation and maintenance. In the development of a process and principles of PUT development, we might hope to borrow from the field of library and information science. The larger point is that, in our fascination with the potential of new tools and technology, we should not overlook the potential to improve through changes in our techniques, strategies and habits of PIM.

References

1. Adar, E., Karger, D. and Stein, L.A., Haystack: per-user information environment. in *8th Conference on Information and Knowledge Management (CIKM 1999)*, (Kansas City, MO., 1999), ACM, 413-422.
2. Barreau, D.K. and Nardi, B.A. Finding and reminding: file organization from the desktop. *SIGCHI Bulletin*, *27* (3). 7.
3. Bellotti, V., Dalal, B, Good, N, Flynn, P, Bobrow, D. G. and and Ducheneaut, N., What a To-Do: Studies of Task Management Towards the Design of a Personal Task List Manager. in *Conference on Human Factors in Computing Systems*, (Vienna, Austria, 2004), ACM Press, 735-742.
4. Bellotti, V., Ducheneaut, N., Howard, M. and Smith, I., Taking email to task: The design and evaluation of a task management centered email tool. in *Conference on Human Factors in Computing Systems (CHI 2003)*, (Ft. Lauderdale, Florida, 2003), ACM Press, 345-352.
5. Berners-Lee., T. Semantic Web roadmap: An attempt to give a high-level plan of the architecture of the Semantic Web.
6. Bush, V. As We May Think *The Atlantic Monthly*, 1945.
7. Clarkson, B.P. Life Patterns: structure from wearable sensors, Massachusetts Institute of Technology,, 2002.
8. Czerwinski, M., Horvitz, E and Wilhite, S, A diary study of task switching and interruptions. in *Proceedings of the 2004 conference on Human factors in computing systems*, (Vienna, Austria., 2004), ACM Press, Pages: 175 - 182.
9. David R. Karger, K.B., David Huynh, Dennis Quan, and Vineet Sinha, Haystack: A General Purpose Information Management Tool for End Users of Semistructured Data. in *CIDR*, (2005).
10. Dourish, P., Edwards, W.K., LaMarca, A., Lamping, J., Petersen, K., Salisbury, M., Terry, D.B. and Thornton, J. Extending document management systems with user-specific active properties. *ACM Transactions on Information Systems*, *18* (2). 140-170.
11. Dourish, P., Edwards, W.K., LaMarca, A. and Salisbury, M. Presto: an experimental architecture for fluid interactive document spaces. *ACM Transactions on Computer-Human Interaction*, *6* (2). 133-161.
12. Dourish, P., Edwards, W.K., LaMarca, A. and Salisbury, M., Using Properties for Uniform Interaction in the Presto Document System. in *UIST'99*, (1999).
13. Ducheneaut, N. and Bellotti, V. E-mail as habitat. *Interactions*, *8* (5). 30-38.
14. Fertig, S., Freeman, E. and Gelernter, D., Lifestreams: an alternative to the desktop metaphor. in *Conference on Human Factors in Computing Systems (CHI 1996)*, (Vancouver, B.C., 1996), ACM, 410-411.
15. Freeman, E. and Gelernter, D. Lifestreams: A Storage Model for Personal Data. *ACM SIGMOD Record (ACM Special Interest Group on Management of Data)*, *25* (1). 80-86.
16. Fried, I. Tiger, Longhorn search for desktop answers *C/net News.com*, 2004, June 30, 2004.

17. Gemmell, J., Bell, G., Lueder, R., Drucker, S. and Wong, C., Mylifebits: fulfilling the memex vision. in *2002 ACM workshops on Multimedia*, (Juan-les-Pins, France, 2002), ACM Press, 235-238.
18. Gemmell, J., Lueder, Roger, and Bell, Gordon, The MyLifeBits Lifetime Store. in *ACM SIGMM 2003 Workshop on Experiential Telepresence (ETP 2003)*, (Berkeley, CA, 2003).
19. Henderson, A.a.C., S. Rooms: the use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. *ACM Transactions on Graphics (TOG)*, *5* (3). 211-243.
20. Huynh, D., Karger, D. and Quan, D., Haystack: A Platform for Creating, Organizing and Visualizing Information Using RDF. in *Semantic Web Workshop*, (2002).
21. Jones, W. Finders, keepers? The present and future perfect in support of personal information management. *First Monday,* *http://www.firstmonday.dk/issues/issue9_3/jones/index.html*.
22. Jones, W., Munat, C & Bruce, H., The Universal Labeler: Plan the Project and Let Your Information Follow (accepted). in *68th Annual Meeting of the American Society for Information Science and Technology (ASIST 2005)*, (Charlotte, North Carolina, 2005), American Society for Information Science & Technology, TBD.
23. Jones, W., Phuwanartnurak, A. J., Gill, R & Bruce, H., Don't Take My Folders Away! Organizing Personal Information to Get Things Done. in *Conference on Human Factors in Computing Systems (CHI 2005)*, (Portland, OR., 2005), ACM Press.
24. Kaptelinin, V., Integrating tools and tasks: UMEA: translating interaction histories into project contexts. in *ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2003)*, (Ft. Lauderdale, Florida, 2003), 353 - 360.
25. Karger, D.R. and Quan, D. Collections: flexible, essential tools for information management in *CHI '04 extended abstracts on Human factors in computing systems* ACM Press, Vienna, Austria 2004 1159-1162
26. Kwasnik, B.H., How a personal document's intended use or purpose affects its classification in an office. in *12th Annual ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1989)*, (Cambridge, MA., 1989), ACM SIGIR, 207-210.
27. Lansdale, M. Remembering about documents : memory for appearance, format, and location. *Ergonomics*, *34* (8). 1161-1178.
28. Lansdale, M. and Edmonds, E. Using memory for events in the design of personal filing systems. *International Journal of Man-Machine Studies*, *36*. 97-126.
29. Larkin, J.S., H. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, *11*. 65-99.
30. Mackay, W.E., More than Just a Communication System: Diversity in the Use of Electronic Mail. in *Conference on Computer-Supported Cooperative Work*, (1988), ACM NY, 26-28.
31. Mary Czerwinski , D.G., Jim Gemmell, Tiziana Catarci, Cathy Marshall, Manuel Perez-Quinones & Meredith M Skeels Digital Memories & Ubiquitous Computing. *Communications of the Acm, Special Issue on Personal Information Management*.
32. Niedzviecki), S.M.w.H. *Cyborg: Digital Destiny and Human Possibility in the Age of the Wearable Computer*. Randomhouse (Doubleday), 2001.

33. O'Day, V. and Jeffries, R., Orienteering in an information landscape: How information seekers get from here to there. in *ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 1993)*, (1993), 438-445.
34. Quan, D., Huynh, D. and Karger, D.R., Haystack: A Platform for Authoring End User Semantic Web Applications. in *2nd International Semantic Web Conference (ISWC 2003)*, (2003).
35. Ranganathan, S.R. *The Colon classification.* Graduate School of Library Service, Rutgers, the State University., New Bruswick, N.J.,, 1965.
36. Steve Mann, A.S., and James Fung, Continuous Lifelong Capture of Personal Experience Using Eyetap. in *First ACM Workshop on Continuous Archival and Retrieval of Personal Experiences (CARPE '04)*, (New York, NY, USA, 2004), 1-21.
37. Teevan, J., Alvarado, C., Ackerman, M.S. and Karger, D.R., The Perfect Search Engine Is Not Enough: A Study of Orienteering Behavior in Directed Search. in *ACM Conference on Human Factors in Computing Systems (CHI '04)*, (Vienna, Austria, 2004).
38. Tony Buzan, B.B. *The Mind Map Book: How to Use Radiant Thinking to Maximize Your Brain's Untapped Potential*, 2004.
39. Whittaker, S. and Sidner, C., Email overload: exploring personal information management of email. in *Conference on Human Factors in Computing Systems (CHI 1996)*, (Vancouver, B.C., 1996), ACM SIGCHI, 276-283.