Defragmenting Information using the Syncables Framework

Manas Tungare, Pardha S. Pyla, Miten Sampat, Manuel A. Pérez-Quiñones
Dept. of Computer Science and Center for Human-Computer Interaction
Virginia Tech, Blacksburg VA USA.
{manas, ppyla, msampat, perez}@vt.edu

ABSTRACT

Information fragmentation is the problem of having a user's data tied to different formats, distributed across multiple locations, manipulated by different applications, and residing in a generally disconnected manner. In this paper, we take a closer look at the problem and describe the design of a framework that addresses information fragmentation of personal information across multiple platforms. This framework recognizes that there is data beyond files, provides a way to address data items consistently across applications and devices, and enables migrating task information seamlessly across diverse platforms without additional effort on part of the users.

1. INTRODUCTION AND MOTIVATION

Information fragmentation is the condition of having a user's data tied to different formats, distributed across multiple locations, manipulated by different applications, and residing in a generally disconnected manner. In current Personal Information Management (PIM) systems, information formats determine storage locations, means of access, addressing of individual pieces of information, and facilities to store or search the collections. Bellotti *et al.* [5] refer to the same problem as compartmentalization of information.

For example, Bergman *et al.* [6] describe the case of information fragmentation for a student who has her project-related data in three formats under three different hierarchies: documents, emails, and bookmarks. Since users associate information objects with their projects and tasks, rather than document formats [6], this represents a potential disconnect between information systems and users' needs.

Also, in this age of mobile computing, it is extremely common for users to perform their tasks using multiple devices ranging from the desktop computer to the ubiquitous cell phone. However, migrating a task from one device to another requires stopping work, saving the data currently in use to a file, transferring it to the second device, opening and loading an assortment of applications to complement or replace the applications being used on the first device, and then restarting work on the original task. Context information is often lost, and users must perform additional steps to be able to resume their task where they left off. This situation adds one more dimension to the information fragmentation problem, since document collections themselves are now fragmented across multiple devices and platforms. It is this dimension that we address on in this paper.

SIGIR PIM Workshop '2006 Seattle, Washington.

2. RELATED WORK

The problem of information fragmentation has been widely studied in the literature. Bellotti *et al.* [5] explored the design of a PIM system iteratively by going back to the users for feedback. Boardman *et al.* [8] studied the organizational hierarchies created by users for bookmarks, files and emails, and noted a significant amount of overlap between the file and email folder hierarchies.

Jones *et al.* [16] found that although some users used web-browser features such as bookmarks or history lists to preserve website addresses for later access, a significant number sent email to themselves with the URL and a brief note about their personal interest in that webpage. In addition to sending bookmarks over email, Whittaker *et al.* [26] observed that users often used email systems for purposes such as personal task management, task requests from collaborators, personal archiving, and asynchronous communication. This is a case of information fragmentation, since artifacts are being kept in a format different from their 'native' format. One of the main reasons behind this practice is that email is accessible from any device with an Internet connection, while other types of information may not be [16].

When moving an item between collections, or splitting up an item into multiple collections, useful metadata is often lost. For example, whenever a file received as an email attachment is saved to a local disk, information such as the sender, subject and links to other messages in that conversation thread are not preserved [15].

Systems such as PaperSpace [22] are good first steps in addressing the *'keeping'* issues of information fragmentation. PaperSpace attempts to bridge the digital and physical worlds by taking the best of the two worlds. Similarly, Indratmo *et al.* [15] propose a context-rich environment that has the potential to help users have a better understanding of where their data exists. The ProjectFolders system [6] is a good attempt at reducing the information fragmentation by providing a single hierarchy for different formats of data.

Systems designed with the awareness of this problem can be roughly classified into two types: those that attempt to provide a unified search interface, and those that attempt to provide a cross-tool browsing interface. Thus, systems such as Google Desktop [14] and Mac OS X Spotlight [2] fall into the first category, while projects such as Haystack [1, 18], LifeStreams [13], Presto [10], WorkspaceMirror [8], WinCuts [23], and Stuff I've Seen [12] fall into the second category.

However, neither of these systems addresses the two issues of paramount importance: addressing items or pieces of information independently of their format, and making this information accessible across a users' multiple devices.

3. PHILOSOPHICAL UNDERPINNINGS FOR INFORMATION FRAGMENTATION

Fundamentally, human existence is socially mediated within the life cycle of creation, assimilation, and exchange of information. From the ancient cave drawings to the medieval abacuses to the modern computers, some of the paramount of human endeavors were towards the conception of artifacts to sustain one part or another of this information life-cycle. In the abstract, information exists as a continuum surrounding the various facets of everyday life: information about one's automobile includes particulars from bank payments to insurance policies, to details about maintenance and service. In other words, information rarely exists atomically; the tax returns include information about the bank loan; the annual budget, in turn, includes information about the tax return, and so on.

As the amount of this information grows, there arises the need to keep it in a structure that affords retrieval (or for a variety of other reasons [19, 25]). And any structure, by definition, is an implicit fragmentation mechanism, geared to splitting up information into manageable units for easy assimilation and exchange. However, humans have learned to naturally cope with this fragmentation in their physical world using their advanced cognitive, spatial [20], and association skills.

Two developments disturbed this coping equilibrium: the information explosion in the last few decades (mandating complex archival structures and strategies which implicitly fragmented the information) and the advent of digital technology (with metaphors and ideas which lacked a vision for the future, ultimately resulting in the information broken into arbitrary classifications).

4. CAUSES OF INFORMATION FRAGMENTATION

The fundamental cause of information fragmentation is still the fact that data exists on a continuum, that is data is implicitly tied to other data and is often difficult to abstract and structure. Current technological solutions impose an arbitrary and often unnatural form to it. A classic example of this fact is the file folder metaphor that labors under the assumption that one's data can be unambiguously classified into a hierarchy. This assumption forces the user to put a piece of information in a single location; when that piece of information falls on different points on one's continuum, an arbitrary decision should be made to place that information in the best location for that time, thereby fragmenting the information and resulting in inaccurate filing (with respect to real understanding on that information). In other words, this is a problem of 'keeping' (where should one store this?) in such a way that finding (where was this stored?) would be easy [17].

4.1 Heterogeneity of Collections

Even though humans think of information in terms of interconnected, context-rich, association of experiences, the same are not replicated in the digital realm. For example, the idea of a contact (say, a friend) in the physical (real) world is often stored or filed (in long-term memory) as an abstraction of the friend's name, place of residence, mental images, shared experiences, mutual friends, and a variety of other attributes. However, when this information is brought into the digital world, a significant amount of detail and context is lost in translation. For example, when we save a URL a friend sent as a bookmark, we lose the context of that information – it was sent by a friend when we were chatting on the 4^{th} of July – stripping the social cues that might be crucial in retrieving or assimilating that information at a later time. Further, when this information is filed in the digital world, the various attributes get fragmented because of the architecture of current solutions: address in the address book, shared experiences in photos or videos, communication in emails, and some other attributes in files or hypertext. Therefore these digital representations cause information fragmentation and result in an unnatural filing structure. Solving this problem is one of the holy grails of PIM research, starting from the vision of a Memex [9] to more recent efforts such as Haystack [1].

4.2 The Metadata Problem

An important problem results from the fact that different document types and collections involve different relevant metadata fields. For example, a file has a creation date and a last-accessed date, an email message has an associated sender, subject and conversation thread, while music files have fields such as genres, artist names, and albums. Material documents (i.e. non-digital artifacts) have many more dimensions of metadata associated with them: for example, users may remember a book by its cover, a document by its thickness, or a paper by its dog-eared corners. Such metadata does not translate well to the digital domain.

When moving an item between collections, or splitting up an item into multiple collections, useful metadata is often lost. For example, whenever a file received as an email attachment is saved to a local disk, information such as the sender, subject and links to other messages in that conversation thread are not preserved [15].

4.3 Multiplicity of Applications

The inconsistency among these metadata fields has led current designs to require independent applications for each such type; for example, email programs to read emails [11], browsers to read hypertext, etc. Current approaches to information systems design treat the difference in metadata formats as valid justification for creating independent applications; attempts to separate content organization from presentation, and thus to use different renderers atop the same browsing framework have met with limited success [3].

Most applications are designed in isolation to support a particular task (for example, an email application is designed to handle email only [24]). Because most applications do not take a global perspective toward the user's information as a continuum, they often use different locations to save their respective data. This puts the burden on the user for maintaining different hierarchies for the different task data [8]. From the scenario described above, Jane has to maintain three chemistry folders in different locations for the three applications she uses to keep her chemistry information, with all the inherent versioning, contextual, and other extraneous overheads in the process.

4.4 Diversity in Platforms

The problem of information fragmentation is compounded even more with the proliferation of different computing platforms such as laptops, PDAs, and cellphones. Not only does data exist in different formats and locations, requiring different applications, there are now different dimensions to this problem: versioning (*"where is the latest version of my document?"*), context preservation (*"in what state did I leave my information?"*), data flavors (*"can I keep my information on this platform and not lose its richness?"*), etc.

Information of the same type that is stored on different devices often ends up as part of different collections. For example, email

messages might be saved and accessed via an email client on a desktop computer, but the same messages could be viewed in a web-browser on a mobile device. Similarly, a user might create new calendar events on a mobile platform, but it is not the same as the desktop calendar events collection because of a pending synchronization operation. Thus, the availability of multiple devices and platforms further exacerbates the information fragmentation problem.

4.5 Inadequate Support for Different States of Data

Barreau *et al.* [4], in their seminal study on users' perception of files and file systems concluded that users do not always think of their data in terms of a hierarchical structure. Often, the data need not even be associated with a file on disk and could be temporary. They classify users' data as *ephemeral*, *working* or *archived* data [4]. Data such as current contents of the operating system's clipboard, position of the cursor (edit position) within a document, and knowledge about the currently-running applications and open documents are ephemeral, because this information is never saved to disk or given a persistent name. Similarly, contents of open documents and personal information such as to-do lists constitute working information that has not yet been assigned a specific position in the user's many hierarchies (although they will be filed away at some point in the future.)

Current tools and systems help manage only archived data, but there is inadequate support for addressing and manipulating ephemeral and working data.

5. OUR APPROACH

Our overarching research objective in this work is to create a framework that seamlessly migrates a task across different platforms. Migrating a task across platforms entails many dimensions such as the transfer of data and context, the transformation of data to suit the target platform, the provision of cues to recover context on the second platform, etc [21]. In the interest of space, a complete discussion of all the dimensions of seamless task migration is not possible here.

In this section, we discuss in detail the design goals and an overview of the implementation for one aspect of our system, the Syncables framework, and how it addresses some of the issues that contribute to information fragmentation when using multiple devices. The Syncables framework lets applications migrate their task information seamlessly across multiple platforms. Information need not be saved to a file first, and the framework provides a unified addressing scheme for each unit of data stored. Applications can thus directly address data created or manipulated in other applications, and create deep links between data items. These are similar in concept to the information trails proposed by Vannevar Bush in his seminal 1945 paper [9].

Other features of the Syncables framework include support for transcoding and filtering data for appropriateness to less-capable and/or mobile platforms, as well as version control, but these features are not discussed in detail in this paper.

5.1 Syncable Objects

An application developer that wishes to synchronize data must define its Syncable objects (or use one of the pre-defined ones). A few examples of Syncable objects would be files, calendar events, email messages, notes, to-do items, contact information, etc. A syncable object in our framework can be more than information that is typically stored on a computer. A syncable object can also represent application state information such as the current television channel, or radio station frequency, or a list of unheard voice-mail messages.

The three properties each syncable object must expose to the framework are, its fully-qualified name, an `OutputStream` to which data will be written during synchronization, and an `InputStream` from which data will be read. The framework does not try to interpret the resulting stream of bytes in any meaningful way (transcoders and filters do, but the framework itself does not assign any meaning to these bytes.)

5.2 A Consistent Hierarchical Naming Scheme

Identification of a syncable object is via a Uniform Resource Identifier (URI) [7]. Each such syncable is assigned a URI that consists of four main parts: , as illustrated in the example below:

`sync://<info-cluster-id>/<syncable-type>/`
`<path-defined-by-type>/<object-name>`

An Information Cluster ID defines the cluster in which data will be migrated; the Syncable type is a string that indicates the type of syncable object: it may be a `File` or a `Calendar` or `Note`, or a custom type for an application. The rest of the hierarchical name of the object is entirely defined by an application as part of its Syncable type definition.

An example of a calendar event is as below:
sync://1D220FFE-B291-58B1-FAA1-C96B7883225C
/Calendar/2006/05/01/Meeting-With-Steve

5.3 Transparency of File Directory

Applications are free to use any file directory location for their data. The Syncables framework does not make any assumptions about where data is stored. All that is required is that an application provide a means for reading and writing that data when presented with a URI request.

When dealing with multiple devices, the exact location of that data is device-independent and transparent to the user. Requests for that data do not specifically state what device it is to be fetched from. This is required because each application as well as operating system has its own conventions for naming and storage of archived files. Furthermore, there is a conspicuous lack of naming conventions for working data or ephemeral data.

5.4 Not Just Files

As Barreau *et al.* [4] stress, considering a user's data as consisting of just files is to take a very narrow look at the data. Our goal was to enable migration of ephemeral, working as well as archived data. As discussed in section 5.1, the Syncables framework allows synchronization of arbitrary streams of data. Hence, task information such as scroll position context, information about running applications and open documents can be made available across platforms to enable seamless task migration. In the absence of the Syncables framework, the user would have to save this data to disk first, then copy the file via any means (USB drive, network, email) to the destination device, and bring the data into his/her working space again.

6. FUTURE WORK

We are currently creating many Syncable object definitions, such as the ability to capture and relay information about running applications, documents, and contextual information about these applications. To address information fragmentation, we are exploring how to capture and transfer information that is traditionally not stored in user files. This includes things like preference settings (e.g. rules in an email program), state information (e.g. currently

selected paragraph in a word processor), and even information that has not been saved yet (e.g. an unsaved document).

We intend to evaluate the framework from two perspectives: users' perspective and developers' perspective. Since the usefulness of such a tool can only be gauged by repeated, frequent use in a mobile setting, we plan to conduct a longer-term field trial by inviting participants to try it for a significant amount of time. Information gathered from interviews and data collected from system logs can help us determine whether the framework represents a significant step towards bridging task disconnects.

7. CONCLUSION

Information fragmentation is a complex problem that needs a radical rethinking of a variety of issues in the PIM life cycle. Nevertheless, given the limitations and acceptance of current technology choices, we present in this paper a solution that addresses the information fragmentation produced by using multiple devices. The Syncables framework assigns each information item a unique name, independent of its source or format, and provides the plumbing for applications to migrate their task information across multiple platforms. We believe that the adoption of tools based on Syncables will mitigate the disruptive effects of information fragmentation in a multi-platform context.

8. REFERENCES

[1] E. Adar, D. Kargar, and L. A. Stein. Haystack: per-user information environments. In *CIKM '99: Proceedings of the eighth international conference on Information and knowledge management*, pages 413–422, New York, NY, USA, 1999. ACM Press.

[2] Apple, Inc. Mac OS X Spotlight. http://www.apple.com/macosx/features/spotlight/, 2004.

[3] Apple, Inc. OpenDoc. http://developer.apple.com/-documentation/macos8/Legacy/OpenDoc/opendoc.html, Last Accessed: May 2006.

[4] D. Barreau and B. A. Nardi. Finding and reminding: file organization from the desktop. *SIGCHI Bull.*, 27(3):39–43, 1995.

[5] V. Bellotti and I. Smith. Informing the design of an information management system with iterative fieldwork. In *DIS '00: Proceedings of the conference on Designing interactive systems*, pages 227–237, New York, NY, USA, 2000. ACM Press.

[6] O. Bergman, R. Beyth-Marom, and R. Nachmias. The project fragmentation problem in personal information management. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 271–274, New York, NY, USA, 2006. ACM Press.

[7] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifiers (URI): Generic Syntax. http://www.ietf.org/rfc/rfc2396.txt, August 1998.

[8] R. Boardman, R. Spence, and M. A. Sasse. Too many hierarchies?: The daily struggle for control of the workspace. In *Proc. HCI International 2003*, 2003.

[9] V. Bush. As we may think. *The Atlantic Monthly*, 1945.

[10] P. Dourish, W. K. Edwards, A. LaMarca, and M. Salisbury. Presto: An Experimental Architecture for Fluid Interactive Document Spaces. *ACM Trans. Comput.-Hum. Interact.*, 6(2):133–161, 1999.

[11] N. Ducheneaut and V. Bellotti. E-mail as habitat: an exploration of embedded personal information management. *interactions*, 8(5):30–38, 2001.

[12] S. Dumais, E. Cutrell, J. Cadiz, G. Jancke, R. Sarin, and D. C. Robbins. Stuff i've seen: a system for personal information retrieval and re-use. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 72–79, New York, NY, USA, 2003. ACM Press.

[13] S. Fertig, E. Freeman, and D. Gelernter. Lifestreams: an alternative to the desktop metaphor. In *CHI '96: Conference companion on Human factors in computing systems*, pages 410–411, New York, NY, USA, 1996. ACM Press.

[14] Google, Inc. Google Desktop. http://desktop.google.com/, 2004.

[15] I. Indratmo and J. Vassileva. No more isolated files: Managing files as social artifacts. In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, pages 899–904, New York, NY, USA, 2006. ACM Press.

[16] W. Jones, H. Bruce, and S. Dumais. Keeping found things found on the web. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 119–126, New York, NY, USA, 2001. ACM Press.

[17] W. P. Jones. Finders, keepers? the present and future perfect in support of personal information management. *First Monday*, 9(3), 2004.

[18] D. R. Karger and D. Quan. Haystack: a user interface for creating, browsing, and organizing arbitrary semistructured information. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, pages 777–778, New York, NY, USA, 2004. ACM Press.

[19] J. J. Kaye, J. Vertesi, S. Avery, A. Dafoe, S. David, L. Onaga, I. Rosero, and T. Pinch. To have and to hold: exploring the personal archive. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 275–284, New York, NY, USA, 2006. ACM Press.

[20] B. Nardi and D. Barreau. Finding and reminding revisited: appropriate metaphors for file organization at the desktop. *SIGCHI Bull.*, 29(1):76–78, 1997.

[21] P. S. Pyla, J. Holman, and M. A. Pérez-Quiñones. Designing for Seamless Task Migration in MPUIs: Bridging Task-Disconnects. Technical Report 04-24, Virginia Tech, 2004.

[22] J. Smith, J. Long, T. Lung, M. M. Anwar, and S. Subramanian. Paperspace: a system for managing digital and paper documents. In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, pages 1343–1348, New York, NY, USA, 2006. ACM Press.

[23] D. S. Tan, B. Meyers, and M. Czerwinski. Wincuts: manipulating arbitrary window regions for more effective use of screen space. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, pages 1525–1528, New York, NY, USA, 2004. ACM Press.

[24] S. Whittaker, V. Bellotti, and J. Gwizdka. Email in personal information management. *Commun. ACM*, 49(1):68–73, 2006.

[25] S. Whittaker and J. Hirschberg. The character, value, and management of personal paper archives. *ACM Trans. Comput.-Hum. Interact.*, 8(2):150–170, 2001.

[26] S. Whittaker and C. Sidner. Email overload: exploring personal information management of email. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 276–283, New York, NY, USA, 1996. ACM Press.