Which Version is This?: Improving the Desktop Experience within a Copy-Aware Computing Ecosystem

Amy K. Karlson, Greg Smith, Bongshin Lee
Microsoft Research
One Microsoft Way, Redmond, WA 98052
{karlson, gregsmi, bongshin}@microsoft.com

ABSTRACT

Computers today make it easy for people to scatter copies and versions of digital items across their file systems, but do little to help people manage the resulting mess. In this paper, we introduce the concept of a copy-aware computing ecosystem, inspired by a vision of computing when systems track and surface copy relationships between files. Based on two deployments of a copy-aware software prototype and in-depth interviews with individuals in collaborative relationships, we present our findings on the origins of copies and the barriers to eliminating them, but offer a promising solution based on the set of files that together represent a user's conceptual view of a document - the *versionset.* We show that the versionset is viable to infer, and we draw upon user activity logs and feedback on personalized views of versionsets to distill guidelines for the factors that define a versionset. We conclude by enumerating the many PIM user experiences that could be transformed as a result.

Author Keywords
File management, PIM, copy-aware computing, versioning.

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms
Human Factors, Design.

INTRODUCTION

> *"Computers are machines for copying data. A good computer is one that copies well, quickly and cheaply."*
>
> *—Cory Doctorow, co-editor of BoingBoing [13]*

While Doctorow's statement was made in the context of digital rights management and the proliferation of copyrighted materials, the insight is no less important for personal datasets. Backups, email attachments, synchronization services, collaboration spaces, clipboard operations, "Save As" dialogs, and many other mechanisms in common use today all result in a full or partial copy of a stream of digital content. The depth to which copying behaviors are built into today's computing systems means that a document that starts as a single file can very easily become two, or ten, or a hundred files with very little explicit effort—sometimes no effort—on the user's part. Indeed, people are filling their hard drives as quickly as storage capacity is increasing [1], and there is evidence that much of this data represents copies of other files, such as those originating from shared sources [1], intermediate versions of documents [27], and duplicates across multiple personal devices [22].

The role that these rampant copies play in supporting or undermining personal information management (PIM) strategies has received relatively little research attention. It is obvious that copies—in particular, potentially divergent copies—can lead to significant PIM confusion on a regular basis: "*Which copy of this document is the one that I submitted for review? Which one did I send to Alice? Is this the only copy I have of this picture?"* And basic human factors research informs us that the presence of additional files in a folder will increase the time it takes a person to scan and select the desired one [21]. Whether the content across various copies remains identical or diverges widely, there are a great number of scenarios in which the user considers them "versions" of the same document, and is negatively impacted when her documents' digital manifestations do not reflect this fundamental connection.

In this paper, we will use the term **versionset** to represent this set of digital items that users conceptualize as a single entity. Through our investigation of the content copy operations that lead to versionsets, we uncovered ample evidence of the problems they pose to effective PIM. But we will also show that these operations are integral to a number of different collaborative and organizational processes, and will never be engineered away. Indeed, over the course of our investigation, we became convinced of the *opportunities* that versionsets present in a **copy-aware computing ecosystem**—an ecosystem that can capture and make use of the semantics behind the copy operations its users so commonly perform. We set out to determine whether the versionset was computationally viable to infer in real time, and whether it could indeed serve as a point of leverage in improving the organizational user experience.

The contributions of this paper are threefold. First, we present a *vision* of the user experience enabled when copy-awareness is built deeply into the interface, incorporating

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.
Copyright 2011 ACM 978-1-4503-0267-8/11/05....$10.00.

several examples we have implemented in an early prototype form. Second, we *motivate* the need and desirability for such a system with insights gleaned from two phases of copy-aware software prototype deployment, combined with in-depth interviews and data collection. We offer a characterization of copies as "crucial clutter"—difficult to manage, but ultimately unavoidable, and even desirable in many circumstances. Finally, we distill *design guidelines* for copy-aware systems to infer the higher-level versionset abstraction from low-level inter-file relationships, and to use this abstraction in improving the user's PIM experience across a variety of strategies and tools.

BACKGROUND

Since the early days of the personal computer, there has been no shortage of research to classify and understand the organizational strategies involved in personal information management [3,16,20,24]. As observed by Dourish and others, information workers naturally use abstract concepts like "document" and "project" as organizational constructs in expressing their workflows. But the actual digital items (files, emails, web pages, etc.) in a user's content collection are computational artifacts with strict behaviors defined largely by their technical implementations [14].

The first possible solution to the problem of allowing digital "atoms" on a personal computer to be organized into "molecules" of related content arrived with the first personal computers themselves: the hierarchical file system. Users could freely define folders and subfolders as organizational structures into which to place their files. Although the size of the average user's aggregate content base and the number of constituent personal, shared, or online content repositories has increased dramatically over the last two decades, the primacy of this basic hierarchical storage and retrieval metaphor has yet to be seriously threatened [3,7,18].

Yet no one is under any illusions that the rigid hierarchical metaphor is the perfect solution. At any given moment, the proper mapping between the intuitive concepts of a user's workflow and the digital artifacts of their computing devices is difficult to define; and the inability of today's systems to address and support this complex, fluid, and contextual relationship is a source of ongoing user frustration [9,20, 24]. A wealth of research and commercial effort has gone toward better supporting PIM by allowing the aggregation of a user's digital items and activity into more flexible groupings across a variety of computing domains, and here we outline several important categories of approach.

Item Grouping

Keyword search as a PIM tool has received increasing scrutiny in recent years, but has not achieved the same pervasive utility on the desktop that it has on the web. Despite advances in speed and relevance, user preference for "orienteering" approaches in finding things and regaining context has served to reinforce usage of the venerable hierarchical grouping metaphor [5,28]. Some researchers have recognized the potential of file-to-file relationships to help users with PIM search tasks, and built systems to track and leverage such data [17,25]. Many have exploited groupings of items by clustering items with related content, or expanding the set of search results, e.g., by returning an item in search results that did not contain the actual keyword in question but was related in content to files that did [15].

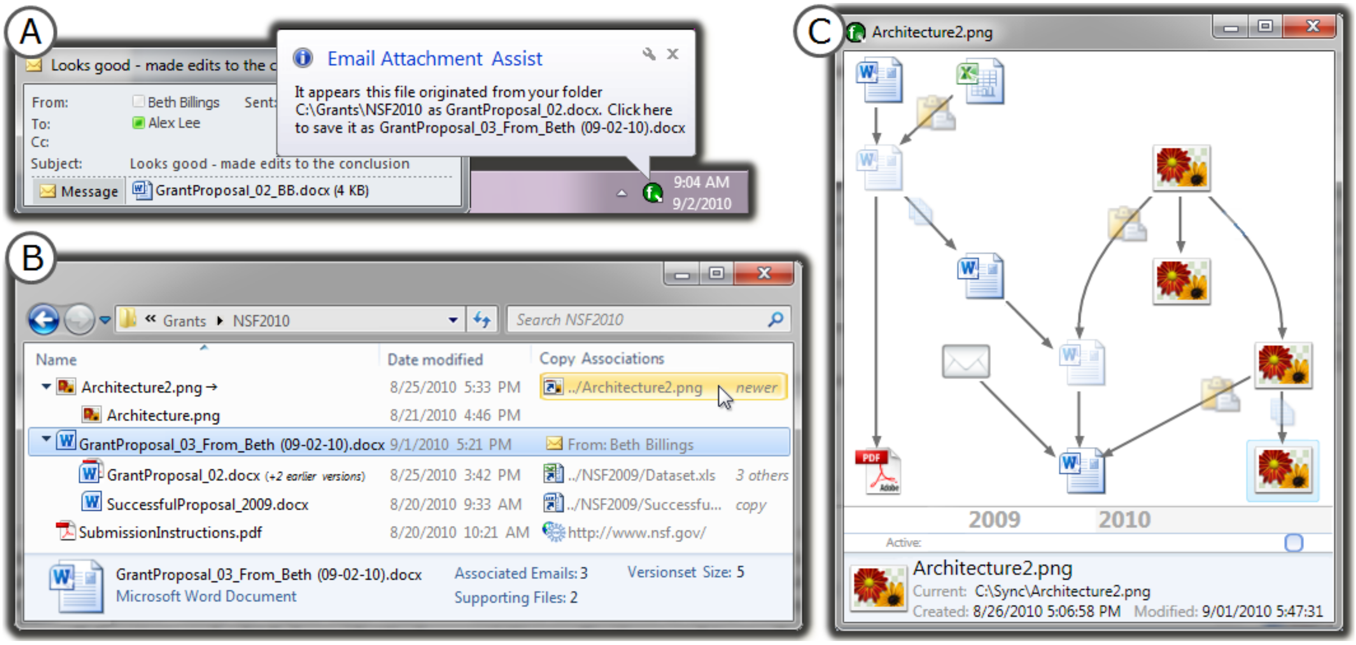
Another, less keyword-centric approach to overcoming users' issues with strict hierarchical groupings involves elevating the role of a digital item's "attributes" to a prime organizational principle. In these systems, items are automatically grouped on attributes such as access time, file size, and content type [12,14,20]. In contrast to the standard hierarchy, where each item has a single path leading to it, such systems allow users the flexibility to intersect desired attribute values in any order when locating a digital resource. Several of these systems have extended the attribute types beyond standard file metadata to include associations with other items [11] or information about the user's activity with respect to the items [15].

The idea of leveraging a user's interaction activity in organization and retrieval has been explored extensively under the general rubric of "activity-based computing." Drawing on activity theory, a number of research systems have been proposed and built to give the abstract user concept of "project" an explicit manifestation and organizational role. In several of these, a user explicitly signals the beginning of an activity, and the system associates an activity tag with subsequently-accessed digital resources to group them or otherwise optimize their future accessibility [2,29]. To obviate the often-prohibitive overhead of manually specifying activity resumption, some systems attempt to infer an activity from the usage of the digital items themselves [23,26], although such efforts are prone to categorization errors.

Interestingly, less well-explored than groupings of heterogeneous items based on attributes or projects is the idea of creating or inferring a group of digital items that might collectively represent a single "document": the versionset. The most obvious reason for this is that the user concept of a document is already inherently much closer to the digital definition of a document—indeed, the digital manifestation was directly inspired by the user concept. But as pointed out in the Introduction, the widespread availability and use of copy operations on today's systems renders the simple one-to-one mapping between "document" and digital item untenable. Recognizing this, there are in fact several research efforts and commercial systems designed with explicit or implicit support for the versionset concept.

Versions

As part of an exploration into information flow between an individual's desktop files, Jensen *et al.* recently called attention to the frequency of inter-file "provenance" relationships (which include versioning relationships) created by today's information workers [17]. Interview data suggested the potential for provenance events to aid in context recall, but the authors point to difficulties posed by the sheer volume of relevant events in designing useful interfaces around such information. The Old'nGray design [4] proposes to



Looks good - made edits to the c

From: Beth Billings Sent:
To: Alex Lee
Cc:
Subject: Looks good - made edits to the conclusion

Message GrantProposal_02_BB.docx (4 KB)

Email Attachment Assist

It appears this file originated from your folder C:\Grants\NSF2010 as GrantProposal_02.docx. Click here to save it as GrantProposal_03_From_Beth (09-02-10).docx

Name	Date modified	Copy Associations	
Architecture2.png →	8/25/2010 5:33 PM	../Architecture2.png	newer
Architecture.png	8/21/2010 4:46 PM		
GrantProposal_03_From_Beth (09-02-10).docx	9/1/2010 5:21 PM	From: Beth Billings	
GrantProposal_02.docx (+2 earlier versions)	8/25/2010 3:42 PM	../NSF2009/Dataset.xls	3 others
SuccessfulProposal_2009.docx	8/20/2010 9:33 AM	../NSF2009/Successfu...	copy
SubmissionInstructions.pdf	8/20/2010 10:21 AM	http://www.nsf.gov/	

GrantProposal_03_From_Beth (09-02-10).docx Associated Emails: 3 Versionset Size: 5
Microsoft Word Document Supporting Files: 2

Architecture2.png
Current: C:\Sync\Architecture2.png
Created: 8/26/2010 5:06:58 PM Modified: 9/01/2010 5:47:31

Figure 1. A copy-aware enhanced user experience. (A) Alex gets an email from his collaborator with the proposal revision attached; the system recognizes this and offers to save the attachment in the appropriate directory using his preferred naming scheme. (B) When Alex goes to edit an image file, he notices a small arrow, showing that a newer version is located elsewhere. Alex clicks on the Copy Associations metadata to investigate. (C) The graph view shows Alex the versionsets for the image and proposal, from which he sees that a newer version of the image is in his synced files folder (C:\Sync).

enhance the user's keyword search and hierarchical browse experience by automatically de-emphasizing older versions of documents in search results and explorer views to reduce clutter. This proposal, based on a "user-subjective approach," represents the most explicit acknowledgement to-date of the potential usefulness of the versionset in PIM literature, but again the authors stop short of defining what the versionset is or how to deduce it. Static analysis of the content duplication resulting from copy operations has also been directly leveraged in improving areas such as disk storage efficiency [8], quality of search result sets [10], and commonality-awareness among co-workers [27]. But none of these approaches propose to leverage the semantics of the copy operations themselves to enhance the user's organizational experience.

In areas such as software engineering and content sharing, several commercial systems exist entirely to implement support for document management and versioning (e.g., GIT, SharePoint, DropBox). These systems are usually centralized, and they dictate a single namespace within which all versioning activity must be reconciled. Unfortunately, from the point of view of a user struggling to organize and maintain a broad collection of content across a range of roles and devices, these systems simply serve to introduce yet another rigid hierarchy. Indeed, sometimes the act of creating a version (such as emailing a document to a home computer for further editing) is taken explicitly to work around an environmental limitation (such as lack of network access) that would render a version control or synchronization system useless. These systems have a number of desirable features, and their wide acceptance in certain environments is testament to the importance of managing document evolution carefully and rigorously. But we will demonstrate that a copy-aware computing ecosystem can provide similar benefits across a wider variety of content and use scenarios, without the overhead of centralized control or pre-meditated version-based structuring.

THE COPY-AWARE USER EXPERIENCE

If computing systems tracked and understood the flow of content across a person's devices, and used that understanding to reason about and reconcile related resources, what might the user experience look like? We use the following everyday scenario to introduce a vision of how deep system awareness of content copy relationships between files has the potential to improve the data management experience for today's information workers.

Scenario

Alex is a researcher who has been working with a small distributed group of collaborators on a project for several months. Alex, as the primary investigator, has been the one most active in pulling supporting resources together and maintaining a shared project folder. A few documents—such as the grant proposal, the system design specification, and milestone reports—periodically circulate through email among team members for edits. Alex tries to keep work at work, but he does subscribe to a commercial synchronization service that allows him to keep certain folders on his home and work machines in sync.

Organizational Assistance

Alex receives an email with the latest revision of the grant proposal as an attachment from Beth, one of his colleagues.

When Alex opens the document, his system asks him if he would like to save it in the project directory with the other versions, to identify which root version Beth was working from, and name the new file meaningfully with respect to this information (Figure 1a). Alex accepts the suggestion and heads home for the day.

Folder View

Returning to the grant proposal project the next day, Alex navigates to his project folder to orient himself for resuming his work. Several features of his copy-aware folder view make his sense-making easier (Figure 1b). First, the many prior versions of his grant proposal are indented with respect to the latest working copy from Beth, with only a few key intermediate versions showing. Because the folder holds many other files, several earlier versions of the grant proposal insignificant to his current browsing context are elided completely, to make more room for other items. In informational columns in his folder view, he can see copy-related metadata at a glance—such as which files were emailed to other collaborators, and which have identical copies in other folders or on other machines. Now that he has the proposal back, Alex wants to embed the updated system architecture diagram he's been working on, so he turns his attention to the PNGs in his project folder. These files are also indented to show a versioning relationship, but here Alex stops—he sees that the latest diagram file in the folder is decorated to indicate there is a newer version outside of his current view! He needs to explore the file's relationship history in more detail to resolve this mystery.

Graph View

Invoking an alternate view of the diagram file in his project folder, Alex is presented with his interaction history coalesced into a graph of relationships (Figure 1c). Here he can see other picture files from which he copied content, and the exact sequence of branching that went on as he iterated on the diagram. Most importantly, he can see that last week he copied the latest PNG into another local folder that was synchronized to his home machine. He now remembers making changes to the diagram from home, but he had not remembered to copy the new version from the sync folder back into his project folder. Correcting this oversight, assured now that he has the correct version, he embeds the latest diagram into the grant proposal.

This brief vision of a copy-aware desktop system has served multiple roles over the course of our project—as an inspiration for developing the necessary infrastructure, as a probe for soliciting end-user experiences about current data management practices and pitfalls, and as a reference point for designing novel representations of users' own data to elicit feedback. Next, we describe the process by which we first exposed end-users to various elements of this vision.

INITIAL EXPLORATION

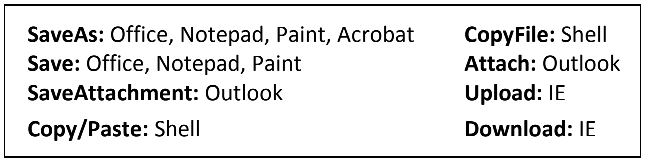
For today's users, evidence suggests that much of the hierarchical approach to organization, and the orienteering approach to retrieval, are grounded in providing *context*: How does a given item or set of items relate to the rest of a user's collection? [28] To explore the question of whether copying and versioning events are potentially under-utilized as contextualizing information, we built a prototype to track these events and expose them explicitly in real-time in the user interface. Our goals in this phase of the project were two-fold: 1) to explore the basic technical feasibility of robustly tracking cross-application and cross-machine copy creation information in a real-world operating system with existing applications; and 2) to use the newly-tracked data to provide useful context to users of the prototype at moments where they might otherwise experience frustration with (or succumb to pitfalls in) their organizational schemes.

Prototype Implementation

Following prior work identifying file system, email, and web as three particularly important domains for PIM data [6,7,17,28], we identified the eight main applications across these domains that were in common use in our organization. We built a Microsoft Windows-based C# .Net Framework prototype that injects a C++ hook library into each of these applications (and Windows itself) at runtime to track a variety of otherwise un-tracked (and un-reconstructable) copy creation events. These events are recorded into a local SQL Compact Edition database in the form of a node/link graph: files, emails, and web URLs are the nodes, and various copy-creation relationships (Save As, SaveAttachment, etc.) are the links (Table 1). We built a visualization for this data, invoked by right-clicking a file in Windows Explorer and selecting "Show History..." from the resulting context menu. The visualization module queries the relationship database to perform an exhaustive graph traversal outward from the target file of all tracked content copy relationships. The results are presented in a graph view (Figure 1c).

During the course of prototype development, we solicited feedback at several events internal to our organization (all refereed to limit acceptance to potentially high-value or interesting technologies) as to the perceived utility of the prototype and the desired feature set. We used the earlier events to get individual feedback via one-on-one demonstrations of the system, and implemented the tracking and features deemed of highest interest to initial audiences. Responses to the system were very positive, quickly accruing 44 installations by users across the organization (who were not given any incentive or compensation for downloading and installing the software).

Ongoing bug reports from this set of users allowed us to iteratively refine our implementation over the ensuing months. After four months we followed up with 9 of the 16 people who were still running the original prototype to understand what was and was not working well for them with



SaveAs: Office, Notepad, Paint, Acrobat	**CopyFile:** Shell
Save: Office, Notepad, Paint	**Attach:** Outlook
SaveAttachment: Outlook	**Upload:** IE
Copy/Paste: Shell	**Download:** IE

Table 1. The file content copy events captured by our system.

respect to the tool itself, as well as in their own data management practices. We conducted semi-structured interviews at the participants' desks to elicit personal stories drawn from their own graphs and PIM structures. We brought with us a tool to identify interesting graphs to explore, and an early version of the file explorer (Figure 1c).

Users Need Integrated Assistance

Interestingly, almost none of the participants reported opening the prototype graph visualization after the initial installation. One possible explanation was that performing an explicit action to explore a graph was an act users would only think to take during "emergency" situations: P1, "*It's definitely a rare scenario that there are things I need to track down, but when it happens, this would save me a lot of time*." Rare as such occasions may be, our system was unused in the opportunities that did occur, suggesting that despite its potential value, the assist was too hidden or too burdensome to invoke.

Yet during our joint interactive exploration, every interviewee discovered relationships that he or she found interesting and useful in the data we explored together, reinforcing findings by Jensen *et al.* [17]. Belying the hypothesis that this information would only be useful in rare cases, the explanations that our interviewees gave for why a particular piece of metadata was interesting often related to an everyday data management task, rather than a special occasion inquiry. A common theme that emerged was the need for assistance in determining the "correct" version of a file. For example, seven participants wanted more confidence that a file being accessed was the most recent version (e.g., that there wasn't a more recent version elsewhere). Four participants showed us folders with several similarly-named files and wanted easy ways to distinguish between them. Three others thought the copy metadata would help them understand which files were *un*important versions that could be deleted or moved. Thus, our lesson was not that users needed better reminders to open the graph view, but that they needed better integration of valuable file relationship information into their existing views and workflows.

While our interviews seemed to confirm that version and copy information could be useful if properly integrated into the existing user experience, we also needed to address the question of whether we were tackling the problem from the right perspective. That is, instead of surfacing information about the copies and versions that exist, perhaps we should be trying to eliminate the creation of copies in the first place? To answer the question of whether eliminating file copies was possible or desirable, we needed deeper insight into how copies manifest within a user's computing ecosystem, and what specific PIM problems they pose.

STUDY: THE HOW AND WHY OF COPIES

We designed a study that would provide us multiple perspectives from which to build a rich picture of the role, positive and negative, that copies play in users' file management practices. We wanted to: a) characterize the *origin* of copies, which would tell us whether the role of technology intervention should be to eliminate copies or to support users in managing them; and b) get users' perspectives on possible solutions to the PIM challenges that copies pose, to gauge the feasibility of using the versionset concept to provide such solutions. These goals dictated a field study involving real user data over a period of time.

Participants

We recruited 16 information workers (6 female) at our institution for the study. Since some of the copies we witnessed in our preliminary evaluation involved sharing files with others, we resolved to include in our study several sets of people who were in collaborative relationships with each other. Among our 16 people, 8 (4 female) were students visiting from academic institutions for a 12-week internship (I1-I8), and 8 (2 female) were full-time employees chosen from each intern's set of project collaborators to serve as mentors (M1-M8). This subject pool had several useful properties for our study. First, it ensured that a half of our participants (the interns) had information management habits originating from outside of our institution. Second, it allowed us to study a set of activities constituting a complete project lifecycle among the interns, yet still investigate organizational environments with multiple simultaneous, potentially long-lived projects among the mentors. Finally, it allowed us to explore the particular implications for copy-aware systems posed by collaborative relationships among information workers.

Approach

We employed multiple data collection methods to study copies simultaneously from three different perspectives:

The file system: We wrote a disk snapshot tool to gather information about the static state of the file system. This tool captured the file path, creation time, last access time, last modified time, and MD5 content hash value of each file on each local hard drive of the target system.

User behavior: We deployed an updated version of our copy-aware software prototype to log and examine user actions that lead to copies. This software logged the set of copy creation events shown in Table 1.

Social and organizational factors: We conducted semi-structured one-on-one interviews with each participant about their organizational and collaborative practices to catalogue the forces that cause users to create copies.

Procedure

Within one week of the start of each intern's internship we ran the disk snapshot tool on both the mentor and intern's primary work computers, and simultaneously installed our copy-aware activity logger. Two weeks prior to the end of each 12-week internship, we instructed each member of the intern/mentor pair to run the disk snapshot tool a second time. At the end of each internship, we instructed each intern/mentor pair to run a final disk snapshot. The snapshot software allowed participants to anonymize file names and URLs within the dataset if desired.

At the end of the internship, we separately interviewed the intern and the mentor about their data management practices. Interviews included questions along four dimensions: team structure and dynamics, including communication and sharing practices; file and data management strategies; archiving practices; and feedback on a design proposal for a copy-aware folder view. In this last phase of the interview, we pre-selected a folder from the participant's own file structure that had a large amount of file activity according to the pre-submitted log data. Using knowledge gathered from the participant's logged copy operations and file accesses, we hand-crafted a hypothetical copy-aware view of the folder using different visual elements to convey versionsets, duplicates, associated emails, and files (including URLs) related by copy/paste relationships. Because this representation was based on real data, not all participants saw all design elements.

RESULTS

As with many field studies, we encountered a number of issues—participant vacations, prototype bugs, machine reliability problems, etc.—that kept us from recording a full twelve weeks of data for each participant. Over the course of the study, we collected data representing 3,336 hours of logged user activity time across 820 unique person-days. Figure 2 shows the aggregate number of copy operations we captured by domain and direction of content movement. We wrote a visualization and analysis tool for exploring the logged data, and used an affinity diagramming approach to categorize the interview data. To address the question of whether copies could and should be eliminated, we turned to our interviews (N=14) to understand the origin of copies from the user's perspective. (One intern/mentor pair was unavailable for the final interview.) Our choice of highly technical, information-intensive study participants obviously dictates that care be taken in generalizing our results. But in the analysis that follows, hopefully the applicability to, say, a home computer user managing and sharing a collection of digital photographs, will be clear.

Crucial Copies

We identified three main categories of copy creation into which the logged copy operations fell among our study population: content preservation, sharing across a user's multiple content hierarchies, and sharing content with others. In each category, we found content copies to be crucial to achieving users' goals.

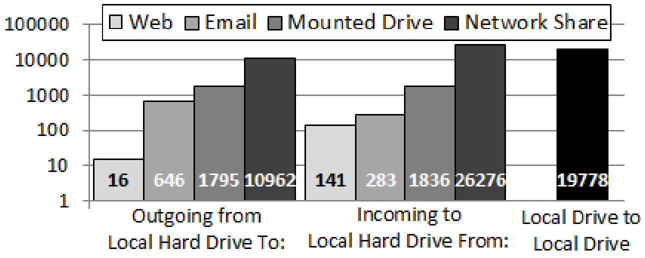


Figure 2. Number of copies logged from, to, and within participants' systems, by domain.

Copies for Content Preservation

To say that people are concerned about losing important digital content would be an understatement. Reinforcing findings by Jones *et al.* [18], participants unanimously kept volumes of data of questionable utility because of fear that a deleted item might someday become valuable. Almost as common was the habit of making deliberate copies of known-useful data for safety—nearly every participant had a nightmare story to relate about the catastrophic loss of useful data.

Participants used a variety of methods to ensure data preservation, including formal backups to servers (M5), manual backups to external drives or computers (M1, M7) data synchronization to other work or personal computers (M1, M4, M5), emailing oneself documents (M1, M3), server-backed source control (M1, M3, M6), and copying files and directories within the same computer to preserve a snapshot of work in progress (I4, I5).

All but three participants reported saving versions of documents to checkpoint those with contributions or comments from collaborators, and typically before sending documents out to collaborators for edit or review. Participants also created versions of documents before or after substantial changes to a document's content or formatting.

Despite their aversion to deletion, participants typically reported that they were "really never going to return to any of those old versions" (M1). Past versions often co-existed in folders alongside more recent versions and thus unavoidably contributed to folder clutter. But participants generally professed to be unfazed by the effect of this clutter because they used consistent manual naming schemes (e.g., incrementing numbers), albeit at the cost of cognitive overhead. Most participants relied on the system-maintained last modification timestamp to identify version relationships—although in one exceptional case, a participant used a script to auto generate version names with a date-time stamp because file modification dates had proven to be fallible.

Copies to Share Data across Devices

The information workers in our study all managed their work across multiple computers and expended considerable effort in making their content accessible from multiple locations—a goal they considered to be crucial to their work. All but one participant performed work activities from both a desktop and laptop. Each of the four mentors who synchronized data manually admitted occasional human error in keeping files up to date and keeping track of the canonical version of a file (M2, M3, M6, M7); both mentors who used email to transfer files admitted they sometimes forgot to email a file or save one they sent from another system. The two mentors who used a formal sync service were more confident in their strategies, but also keenly aware of its fragility. For example, M1 always keeps in mind that his sync software is peer-to-peer: "*It has yet to cause me problems but I do need to be cognizant of the state of the universe and which machine is 'on'*" and M5 falls back to email as insurance when data propagation is slow.

For five of the seven interns, the main challenge was keeping data synchronized between their development and deployment environments. Deployment machines included high performance clusters, web servers, and mobile devices. In all five cases, interns copied files manually between two or more systems and all encountered difficulties keeping changes synchronized between machines.

Copies to Share Data with Others

The need to collaborate and share data with others was a necessary part of all our participants' workflows, and yet was without question the largest contributor to unreconciled, ambiguous, and confusing data copies on their systems. Each intern-mentor pair chose a unique combination of tools and techniques for communicating, coordinating and sharing data during the project. Email attachments were the one sharing method used by every pair. Five pairs used a shared project folder, which was either synchronized automatically (three pairs) or hosted by the mentor (one pair) or intern (one pair). While four of the projects used a source control system, only one pair used it to actively share documents and code with one another during the internship.

All but one project pair used multiple methods to share data, which itself caused confusion about which files and versions were stored and shared in which locations. Although it is tempting to consider whether the solution to sharing-based copies lies in designing a single "perfect" sharing system, we developed two main insights from participants' reported behaviors that suggest little potential for converging on a single solution:

The Ownership Problem: Reinforcing prior PIM literature [19], we found users to be highly attached to their own local organizational practices. When people collaborate, they must either impose their organizational preferences on others, defer to someone else's preferences, or refuse to play—and we saw instances of all three in our study. The group that had the least confusion about collaborative versioning used an approach dictated by the mentor: "I3: *I found M3's process to be very stable. He's been using it a long time and once I learned his process it was easy to use…It just meant I didn't work outside work [because of the need for internal access] but it didn't turn into more artifacts*."

Although some participants implied they would be willing to adopt a collaborator's version naming strategy (I1, M6), we also observed significant friction in groups trying to adopt a single sharing strategy. In one case, a synchronized folder was arranged between the intern and mentor, but was not taken up by new members of the team: M5: "*we also had a bunch of other people working on the project with us and not all of them were invested in getting up to date on the shared folder, so more stuff went back and forth through email than it might have needed to if only I5 and I were working on the project.*" Divergent naming schemes kept M4 from adopting the intern's sharing scheme: "*I completely ignored requests to put data [in the intern's directory], I emailed it to him. [...] I'm a neat freak when it comes to my file organization, so the idea of having two organizing schemes being applied to the same folder at the same time is disturbing to me. So I wouldn't do it. I'd email it to him and say: put it where you want it to be.*"

Distaste for the unruly state of a shared space was echoed by M5, referring to several collaboration spaces as "spaghetti." Yet there is also friction in taking responsibility to clean up a shared space: "*there's a tension between keeping the messy structure [in a shared project folder], because we know it. But there's also some concern that it would break the replication*" (M5). The perceived permanence of one's decisions when sharing data was also the source of some anxiety, as illustrated by M4's discomfort when sharing data out on a public directory: "*I'm traumatized by the fact that it means the filename is crystallized. Once you've shared the link, you'll break it if you change it.*"

The Tools Problem: Participants deemed certain tools to be more appropriate for some data types and work styles. Source control was useful for code but less appropriate for document collaborations because of the need for wide accessibility, which could not be fulfilled when outside the corporate network. Other synchronization methods were seen as too heavyweight or unreliable for time-constrained collaboration. Email was commonly used to circumvent these constraints, but had its own drawbacks. M4 recalled the synchronization problems he recently faced: "*There were so many people writing things all the time that it definitely happened that someone didn't follow the scheme, or, something that happens a lot is that something will get stuck in an outbox because it exceeds a file quota. Someone thinks it has been received or it gets junk-foldered. I think all of these things happened on that project. It was brutal.*" M2 had trouble using email to keep track of multiple documents at the same time: "*We were writing a bunch of papers so it did get a little confusing keeping track of it because, again, we were using email, and, you know, the classic 'use the email thread about the other paper' and so there was a bunch of that that happened.*"

Getting Feedback on Solutions

Our second study question was whether the versionset concept was useful and tractable as a leverage point in solving general copy-related PIM problems. To get useful feedback, we identified a specific PIM user experience for which to propose versionset-derived assistance. From our earlier investigation that folder clutter can be version-related, we hypothesized that visually demarcating a versionset in the standard Explorer folder view might serve to de-clutter the interface during folder browsing. This design was meant to probe the benefit, oft-cited in information management literature [14], of turning a cognitive task (in this case, identifying versions of a file) into a perceptual one.

For this exercise, we defined a versionset as the files related by logged SaveAs, CopyFile and Copy/Paste relationships. We understood this was overly simplistic, but the interview was designed to elicit a wider discussion of the factors involved in the versionset definition. We presented each participant with a paper prototype view of one of their own folders—a familiar organizational context, but with copy-aware enhancements. To get an idea of how the versionset definition might need to change according to context, we also solicited separate feedback from each mentor on their intern's prototype folder view—an unfamiliar organizational context, with the same copy-aware enhancements.

Versionsets were depicted in the view as collapsible sets of related files. In the collapsed view, a versionset was depicted as a single line showing the "canonical" current version; the expanded view showed all files in the versionset, using indentation to depict strict ancestry (SaveAs) relationships, and shortcut decorators on file icons to indicate "related" files as determined by CopyFile and Copy/Paste relationships. We also provided a metadata column showing the people that a file was emailed to or received from. The views were similar to Figure 1b, except that related documents were listed under, rather than next to, the versionset.

Context: My Organization

When looking at their own folders, participants were unanimously enthusiastic about the benefits of the view, especially the notion of the collapsible versionset, and its potential for reducing clutter. Although participants varied in their opinions about what aspects of the visual representation resonated the most, and under which scenarios the visual elements would be most useful, every element of the design was deemed "very useful" by at least several of the participants. Equally important was the finding that the precise versionset definition was an important factor in participants' satisfaction with the visuals. Participants were quick to point out examples of files in the indentation relationship that "didn't belong" in the collapsible set, such as milestone branches that were important in their own right ("submission branch" vs. "camera-ready branch"). It is clear that the correct determination of the versionset is a key factor in user satisfaction with copy-aware interfaces.

One surprise was the use that participants made of the email column, which we expected to be simply informational. However, participants were quite sensitive to the gaps and anomalies in the data that could signal an error in their data organization. As M4 explains: "*There are versions on my hard drive that I don't remember if I've sent them to the publisher or not, so it's happened a couple of times that I've sent two different versions, and the publisher says 'why are you sending us two different versions—which one is the right one?' So this would help. It would eliminate all of that. I could just say 'oh, well I haven't sent this yet.*'"

Context: Someone Else's Organization

When viewing their intern's folder view, mentors deemed the copy relationships equally or more useful than when viewing their own folder. Positive reactions to the resources related by Copy/Paste were explained as "*getting a sense of a file without having to open it*" (M2) and "*automatic annotation and referencing*" (M4). Mentors also appreciated how the indented view made it easy to pick out the "canonical" version of a file, as M1 points out, "*In [the copy-aware] view it is immediately evident which [file] is the most relevant. When I looked in the flat directory I would have been misled and grabbed the wrong one.*"

Overall, visually collapsing a versionset to a single canonical representative file appeared to represent a promising enhancement to the folder view in the existing PIM user experience. When browsing for a particular target version of a particular document, it was clear that the non-target versions of a file represented distracting clutter, and unsurprisingly, collapsing them away would be helpful. Our simple versionset definition for the user-feedback exercise was clearly not entirely sufficient for this task, but the study gave us the data we needed to produce a richer set of guidelines. The feedback taught us the factors that would allow us to create proper definitions and use cases for the versionset in a wider variety of user contexts.

INFERRING THE VERSIONSET

We analyzed the log data and the interviews to get a deeper understanding of what goes into the proper determination of versionset from a user-subjective perspective. Given the effects of context, we did not expect to be able to define a single set of characteristics that would suffice across all scenarios. But we attempted to comprehensively categorize all the factors surfaced to us in the interviews or visible in the logging data that appeared to have relevance in determining the versionset's proper membership.

Hierarchy co-location: Most participants went to great lengths to keep file versions in the same folder to the extent possible, and several reported confusion caused specifically by the issue of versions not being co-located.

File types: Most content copy operations we observed preserved the type of the content file, and thus it is no surprise that versionsets often consisted of a single file type. But there were notable exceptions, such as a .pdf file that represented a final version of a set of .docx files. Such exceptions often served to distinguish a particular file *within* the versionset, because a different operation led to its creation. Participants differed in their estimation of the relative importance different file types played within the versionset.

Naming patterns: In many examples we saw, versions consisted of a base name and a variety of semantic suffixes: numbers for establishing chronological order (e.g., "_09"), initials or person names for establishing ownership transfer (e.g., "_from_John"), and descriptors for representing milestones (e.g., "_final", "_uploaded"). Eleven participants made explicit reference to a "naming scheme", either theirs or someone else's, when discussing versions. It was clear that naming conventions were used both to visually establish versioning relationships and to call out semantic differences within the versionset. Many interviewees had strong attachment to their own personal conventions [19].

Timestamps: Versions often exhibited serial creation and modification dates, and users often depended on these attributes in determining the "correct" version within a versionset to use for a particular purpose. In other situations, a long gap between two clusters of timestamps among a set of files seemingly related by copy operations could be an indicator of a break in the versionset chain—for instance, when an old document with several prior revisions was branched to a new one simply to preserve formatting in kicking off a new document creation and revision process.

Access patterns: Users described accesses to the versionset that tended to be temporally clustered, because multiple versions in a versionset were associated with a particular user 'activity.' That is, multiple files in the same versionset were often open simultaneously, or sequentially, when a user was working on a task involving that document.

Content copy operations: Interviewees confirmed that SaveAs, file copy (CopyFile, Attach), and Copy/Paste operations were strong indicators of version relationships between files. SaveAs was often used to mark significant content changes or preserve earlier content that might otherwise be lost. File copy operations often bridged hierarchies (e.g., between an email attachment and a file system, or between a home and work system) for the purpose of content preservation, transmission to others, or improved accessibility. Depending on the context, the significance of a Copy/Paste operation as an indicator of a version ranged greatly: on one extreme, an entire document might be pasted into a new blank document and re-named in an act functionally equivalent to a SaveAs of a new version; on the other, a small portion of a completely unrelated document could be pasted simply to borrow formatting details; most commonly, Copy/Paste operations successfully identified *related* files, but not proper versionset members. The size and content of the pasted data serve as clues to differentiate the significance of such operations.

Content overlap: Even when specific copy operation history was unavailable, static metrics of content similarity were an indicator of version relationships. Large content overlaps between two files often suggested a common source file. This promises to be a particularly key feature to leverage in a real-world copy-aware system, because copy operations in legacy components or external namespaces that can't be recorded directly may sometimes be recoverable from static content analysis.

These categorizations make it clear that the versionset is not a simple set to define digitally, but each category reveals factors that have useful positive or negative correlations with version relationships in certain contexts.

USING THE VERSIONSET

Our investigation suggests that a wide variety of existing and proposed tools and systems used for personal information management have the potential to be enhanced by the versionset concept as part of a copy-aware ecosystem.

Applications or systems that allow digital items to cross organizational boundaries—such as email clients or synchronization services—are fertile ground for offering **integrated organizational assistance** using the versionset concept. Knowledge of the version correspondences across namespaces would allow these systems to suggest the correct course of action to the user just at the moment of potential confusion, as in Figure 1a. In-depth relationship views such as the graph view proposed in Figure 1c could be triggered by integrated assistance UI or other item-specific mechanisms. These views would have the ability to support complex sense-making with respect to a particular digital item, providing a rich history of user activity viewed through the lens of versionset-relevant operations.

During our investigation, we gained concrete insight on the use of versionsets for enhancing hierarchical navigation views with respect to two different information management contexts: exploring familiar data and exploring unfamiliar data. The difference in context had implications for the proper definition of the versionset, but in both cases the judicious de-emphasis of subordinate content, and the careful inclusion of related content normally not available in-view, showed promise (Figure 1b). Similar principles used in **keyword search** would allow the results view to either collapse items that were tightly related by version to allow more room for other relevant targets, or to include items related by version that might not otherwise match the query.

Attribute-based systems such as Presto [14] could benefit from the versionset by extending the application of properties meant to apply to the user concept of "document" to all the digital manifestations of the document. **Activity-based computing** systems attempting to group a user's digital items into activities could benefit from a particular definition of versionset by automatically extending project associations to all files within the versionset. For example, having associated a particular URL as a related resource to a particular document, they could automatically extend that association to later versions of the same document. In essence, any system that proposes new groupings for a user's organizational experience can benefit by using the versionset as a foundational building block.

CONCLUSION

At any given moment, the file is often the main unit of attention for the understanding, creating, and editing activities that make up the bulk of what an information worker considers to be her information work—whether that file is a web page, an email, or a spreadsheet. Yet the conceptual units of work natural for users to organize their overall workflow—e.g., project, task, or document—commonly span multiple file artifacts. Users today face a proliferation of these artifacts across their many devices but receive scant assistance from their systems in making sense of it.

In this paper we introduce the versionset—a context-sensitive aggregation of digital items defined as the set of files that, at a given moment, represent a single document to the user. We propose that a copy-aware computing ecosystem can leverage its understanding of the mechanisms and semantics of the ways copies and versions of content are created to define and exploit the versionset across a wide range of different PIM user experiences.

Through the deployment of copy-aware software prototypes, and interviews with real users about their data management challenges and practices, we gathered evidence that not only are copies unavoidable for a variety of personal, organizational, and contextual reasons, but they play a crucial role in achieving many specific user goals. At the same time, they cause confusion and inefficiency in users' data management strategies. Our observations allowed us to distill the factors that go into the proper determination of a versionset for a given user context. We also demonstrate that users perceive the versionset to be a valuable construct for managing clutter and reasoning about file versions. Finally, we enumerate the many PIM user experiences that could be transformed by leveraging the versionset. It is our hope that this approach points the way toward a PIM experience substantially less burdensome for tomorrow's users.

REFERENCES

1. Agrawal, N., Bolosky, W.J., Douceur, J.R. and Lorch, J.R. A five-year study of file-system metadata. *ACM Trans. Storage 3*, 3 (2007), 9:1-9:32.
2. Bardram, J., Bunde-Pedersen, J. and Soegaard, M. Support for activity-based computing in a personal computing operating system. *Proc. CHI '06*, ACM Press (2006), 211-220.
3. Barreau, D. and Nardi, B.A. 1995. Finding and reminding: file organization from the desktop. *ACM SIGCHI Bull. 27*, 3 (1995), 39-43.
4. Bergman, O., Beyth-Marom, R. and Nachmias, R. The user-subjective approach to personal information management. *JASIST 54*, 9 (2008). 872-878.
5. Bergman, O., Beyth-Marom, R., Nachmias, R., Gradovitch, N. and Whittaker, S. Improved search engines and navigation preference in personal information management. *ACM Trans. Inf. Syst. 26*, 4 (2008), 1-24.
6. Bergman, O., Tucker, S., Beyth-Marom, R., Cutrell, E. and Whittaker, S. It's not that important: demoting personal information of low subjective importance using GrayArea. *Proc. CHI '09*, ACM Press (2009), 269-278.
7. Boardman, R. and Sasse, M.A. "Stuff goes into the computer and doesn't come out": a cross-tool study of personal information management. *Proc. CHI '04*, ACM Press (2004), 583-590.
8. Bolosky, W.J., Corbin, S., Goebel, D. and John, R.D. Single instance storage in Windows 2000, *Proc. USENIX '00*, USENIX Association (2000), 2-2.
9. Bondarenko, O. and Janssen, R. Documents at hand: learning from paper to improve digital technologies. *Proc. CHI '05*, ACM Press (2005), 121-130.
10. Broder, A.Z., Glassman, S.C., Manasse, M.S. and Zweig, G. Syntactic clustering of the Web. *Compt. Networks ISDN 29*, 8-13 (1997), 1157-1166.
11. Chau, D., Myers, B. and Faulring, A. What to do when search fails: finding information by association. *Proc. CHI '08*, ACM Press (2008), 999-1008.
12. Cutrell, E., Robbins, D.C., Dumais, S.T. and Sarin, R. Fast, flexible filtering with Phlat-personal search and organization made easy. *Proc. CHI '06*, ACM Press (2006), 261-270.
13. Doctorow, C. Copy killers. The Guardian. July 31 2007. Accessed from http://www.guardian.co.uk/technology/2007/jul/31/comment.drm, September 23, 2010.
14. Dourish, P., Edwards, W.K., LaMarca, A. and Salisbury, M. Presto: an experimental architecture for fluid interactive document spaces. *ACM TOCHI 6*, 2 (1999), 133 -161.
15. Gyllstrom, K. and Soules, C. Seeing is retrieving: building information context from what the user sees. *Proc. IUI '08*, ACM Press (2008), 189-198.
16. Henderson, S. Personal document management strategies. *Proc. CHINZ '09*, ACM Press (2009), 69-76.
17. Jensen, C., Lonsdale, H., Wynn, E., Cao, J., Slater, M. and Dietterich, T.G. The life and times of files and information: a study of desktop provenance. *Proc. CHI '10*, ACM Press (2010), 767-776.
18. Jones. W. Finders, keepers? The present and future perfect in support of personal information management, *First Monday 9*, 3 (2004).
19. Jones, W., Phuwanartnurak, A.J., Gill, R. and Bruce, H. Don't take my folders away!: organizing personal information to get things done. *Ext. Abstracts CHI '05*, ACM Press (2005), 1505-1508.
20. Kaye, J., Vertesi, J., Avery, S., Dafoe, A., David, S., Onaga, L., Rosero, I. and Pinch, T. To have and to hold: exploring the personal archive. *Proc. CHI '06*, ACM Press (2006), 275-284.
21. Neisser, U. Visual search. *Scientific American 210*, 6 (1964). 94-102.
22. Oulasvirta, A. and Sumari, L. Mobile kits and laptop trays: managing multiple devices in mobile information work. *Proc. CHI '07*, ACM Press (2007), 1127-1136.
23. Rattenbury, T. and Canny, J. CAAD: an automatic task support system. *Proc. CHI '07*, ACM (2007), 687-696.
24. Ravasio, P., Schär, S.G. and Krueger, H. In pursuit of desktop evolution: User problems and practices with modern desktop systems. *ACM TOCHI 11*, 2 (2004), 156-180.
25. Shah, S., Soules, C.A., Ganger, G.R. and Noble, B.D. Using provenance to aid in personal file search. *Proc. USENIX '07*, (2007), 1-14.
26. Shen, J., Irvine, J., Bao, X., Goodman, M., Kolibaba, S., Tran, A., Carl, F., Kirschner, B., Stumpf, S. and Dietterich, T.G. Detecting and correcting user activity switches: algorithms and interfaces. *Proc. IUI '09*, ACM Press (2009), 117-126.
27. Tang, J. C., Drews, C., Smith, M., Wu, F., Sue, A. and Lau, T. Exploring patterns of social commonality among file directories at work. *Proc. CHI '07*, ACM (2007), 951-960.
28. 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. *Proc. CHI '04*, ACM Press (2004), 415-422.
29. Voida, S. and Mynatt, E.D. It feels better than filing: everyday work experiences in an activity-based computing system. *Proc. CHI '09*, ACM Press (2009), 221-230.