Cardinal: novel software for studying file management behavior

Jesse David Dinneen
School of Information Studies
McGill University
Montreal, QC, Canada
jesse.dinneen@mail.mcgill.ca

Fabian Odoni
Institute for Information Research
University of Applied Sciences (HTW Chur)
Chur, Switzerland
fabian.odoni@htwchur.ch

Ilja Frissen
School of Information Studies
McGill University
Montreal, QC, Canada
ilja.frissen@mcgill.ca

Charles-Antoine Julien
School of Information Studies
McGill University
Montreal, QC, Canada
charles.julien@mcgill.ca

ABSTRACT
In this paper we describe the design and trial use of Cardinal, novel software that overcomes the limitations of existing research tools used in personal information management (PIM) studies focusing on file management (FM) behavior. Cardinal facilitates large-scale collection of FM behavior data along an extensive list of file system properties and additional relevant dimensions (e.g., demographic, software and hardware, etc). It enables anonymous, remote, and asynchronous participation across the 3 major operating systems, uses a simple interface, and provides value to participants by presenting a summary of their file and folder collections. In a 15-day trial implementation, Cardinal examined over 2.3 million files across 46 unsupervised participants. To test its adaptability we extended it to also collect psychological questionnaire responses and technological data from each participant. Participation sessions took an average of just over 10 minutes to complete, and participants reported positive impressions of their interactions. Following the pilot, we revised Cardinal to further decrease participation time and improve the user interface. Our tests suggest that Cardinal is a viable tool for FM research, and so we have made its source freely available to the PIM community.

Keywords
Personal information management, human-computer interaction

ASIST 2016,October 14-18, 2016, Copenhagen, Denmark.

INTRODUCTION
Every day, computer users interact with files and folders, including creating, downloading, naming, moving, saving, copying, reviewing, navigating, searching, and deleting them. This is a deeply personal and psychological activity (Lansdale, 1988) that can be supported by systems and services, but such support requires understanding the behavior that users exhibit and the factors that influence them. Despite many studies of Personal Information Management (PIM) reporting on how people perform file management (FM), a confident characterization of FM behavior has not emerged. This is primarily due to limitations in the available data collection methods. Here we introduce Cardinal, software that addresses these limitations by automating the mass collection of data about PIM behavior while also providing value to participants. In what follows we describe the existing FM data collection methods, detail the design of Cardinal, report on a trial implementation and its results, and conclude by noting the remaining improvements that may benefit FM research.

PROBLEM AREA
Broadly, PIM is an area of study concerned with how and why individuals manage information items, and how the results of these investigations might be used to improve services and systems designed to support such management. Understanding FM behavior and its factors aids the design of PIM systems and services, for example by revealing user preference and behavior in certain contexts. In time, such improvements are implemented in widely used software and improve the FM experience; desktop search and file tagging are examples of this process, having been developed and tested in academic and industrial research before being implemented into major operating systems (Kljun, Mariani, & Dix, 2015).

Many PIM studies have examined FM behavior, including how people name (Carroll, 1982) and organize (Hardof-Jaffe, Hershkovitz, Abu-Kishk, Bergman, & Nachmias, 2009b) files, the challenges of information fragmentation across multiple devices (Capra, 2009), and the various challenges to sharing and retrieving files (Bergman, Whittaker, & Falk, 2014). Together, such studies have provided only broad characterization of users' FM behavior; for example, recent studies have extended characterizations of user's paper-based organization strategies into the digital domain, advancing the characterizations from neat or messy and using files or piles (Malone, 1983) to include mixed approaches (Trullemans & Signer, 2014) and strategies such as filing the majority of files on creation, filing somewhat extensively but leaving many items unfiled, or filing occasionally but leaving most files unfiled (Boardman & Sasse, 2004). The inability to move far beyond these basic findings is due in part to the methods of collecting data that studies have implemented, of which there are three: (1) ask participants about their FM behavior, for example in a questionnaire or interview (2) observe the behavior directly, for example in a 'guided tour' of the desktop or during an experimental task, and (3) infer the behavior from properties of the file system, for example by running software on participants' computers. Each method entails limitations.

The first approach, asking participants, is simple and direct, as data about PIM-relevant perceptions and behavior can be reported by participants, for example when elicited in an interview (Xie, Sonnenwald, & Fulton, 2015). This works well for identifying broad PIM practices and challenges that users remember, like transferring files between computers (Capra, 2009). It is limited, however, as it cannot capture data about activities or aspects of behavior of which users may not be cognizant, like the number of empty folders they keep, and participants' perceptions of their own PIM behavior can be inaccurate (Bergman, Gradovitch, Bar-Ilan, & Beyth-Marom, 2013)

The second approach, observing participant behavior, entails recording participant behavior, for example using video to capture the behavior exhibited during typical work tasks (Bruce, Jones, & Dumais, 2004), guided tours of the participants' desktops (Barreau, 1995), or structured experiment tasks (Bergman, Whittaker, Sanderson, Nachmias, & Ramamoorthy, 2012; Benn *et al.*, 2015). This allows for exploring particular aspects of user behavior in depth, like organizing downloaded files (Jones, Bruce, & Dumais, 2001) and retrieving shared files (Bergman *et al.*, 2014). The limitations of this approach are its temporality and impracticality: as the behavior is always observed during some particular time, the researchers necessarily do not see what participants are doing when not observed and meeting with participants for guided tours or reviewing recordings of experiments are both very time and labor intensive.

The third approach, utilized in much of the FM research literature, is to infer and understand users' behavior by examining the file system, its contents, and its properties, typically by running custom-made software on participants' computers. User behavior determines properties of the file system (e.g., the shape of the folder tree structure, the particular file system contents, the size of the collection), and the file system therefore serves as a record of such behavior. For example, recording folder names allows for discerning if particular conventions are used when a user names folders. Properties of particular files and folders can also be analyzed together to ascertain subtler facts about user behavior, such as the average depth at which a user stores document files or the number of files stored in folders that contain no sub-folders. Studies using this approach have, for example, examined the number and kinds of files people store (Gonçalves & Jorge, 2003), how files are organized across folders (Khoo *et al.*, 2007), and the effect of personality style on desktop tidiness (Massey, TenBrook, Tatum, & Whittaker, 2014).

Examining the file system has clear promise and interest in PIM research, but has yet to be fully realized as implementations have entailed their own limitations. First, despite having at least 40 file system properties available (discussed below), across 37 previous studies we find a mean of 4.4 properties; such studies often describe fewer than 5 properties for their participants' collections (e.g., Boardman & Sasse, 2004; Henderson, 2005), never more than 13 properties at once (e.g., Gonçalves & Jorge, 2003). Findings are therefore typically narrow, preventing researchers from drawing broad conclusions about the typical user's file management behavior. A second problem is that there has been little consistency across the properties examined, thus producing incommensurable findings; for example, while one study reports the number of files left in root folders (Henderson & Srinivasan, 2011), another reports the depths of folders not containing sub-folders (Zhang & Hu, 2014). This prevents comparing results to and analyzing data across studies (e.g., meta-analysis). These problems together make it unclear which properties are related or comprise the principal components of FM behavior, thus begetting further inconsistency of data collection and incommensurability of findings.

The third problem with implemented approaches to examining the file system is that software that makes distribution, administration, and recruitment difficult has caused small population samples despite the automation provided by software. This may be because the software is complex to use, thus requiring researcher guidance, or because it is difficult to find users willing to expose and share their digital possessions and desktops. Where large sample sizes have been achieved, they have been from incommensurable contexts, such as students using a proprietary, online environment during a class assignment (Hardof-Jaffe, Hershkovitz, Abu-Kishk, Bergman, & Nachmias, 2009a) and employees' behavior at a single

software corporation (Douceur& Bolosky, 1999; Agrawal, Bolosky, Douceur, & Lorch, 2007).

Finally, software that examines the file system has rarely supported multiple operating systems, causing researchers to instead rely on limited tools packaged with the OS (Evans & Kuenning, 2002) or to focus on a single OS (Khoo *et al.*, 2007). As a result, suggestions that software factors such as the OS and file manager used have an effect on FM behavior (Barreau, 1995; Massey *et al.*, 2014) have gone virtually unexplored. The overall consequence of the three approaches' limitations is that the results of FM behavior studies thus far have data that is too varied, samples that are too shallow or narrow, and have left important questions unanswered. This must be addressed to produce the kind of data necessary to advance PIM research, for example by producing nuanced models, frameworks, and theories and creating accurate datasets to use when evaluating PIM tools (Chernov *et al.*, 2008). What is needed, then, is software that facilitates the rapid and relatively easy collection of many file system properties, including those used in previous studies, across a large, heterogeneous population sample.

CARDINAL – DESIGN AND USE

We created software, called Cardinal, to overcome the limitations of the existing FM behavior data collection methods. Cardinal is cross-platform (e.g., runs in Windows, Mac OS X, and GNU/Linux), and will run in multiple versions of each OS on computers with both 32- and 64-bit processors. It does not require that users install it, but rather that they download a single small (<30MB) file, for example from a research project's Web site, which can then be run remotely, without researcher supervision, or with supervision, for example in lab settings. Both manually retrieving the data from participants and asking participants to manually send their data are avoided: upon the user's request the resulting data is encrypted, compressed, and sent to a predetermined destination. Cardinal supports sending data to the researchers' computer via secure file transfer protocol (FTP) and to Dropbox via the provided API. Data is stored in the common JSON format so that it can be imported in bulk into statistical software for analysis; an example of the raw data is in the appendix.

To overcome the limitation of inconsistent data collection, we programmed Cardinal to collect 27 of the 28 file system properties collected by previous studies, and 11 of 12 additional properties, totaling 38 of 40 possible properties -- 25 more than collected by the next most widely-collecting tool previously used. The two excluded properties are discussed in this section, and a summary of all mentioned properties is presented in Table 1. Cardinal also collects properties about the technological factors discussed above (e.g., OS and FM software used), and further data may be collected by including additional fields or questionnaires.

Property category (previous + new)	**List of previously examined file system properties (28)**	**Expanded list of properties (12)**
Storage (11 + 6)	Hard drive capacity, use, and free space; total files, total folders; collection size (in bytes), collection size (files + folders); file extensions/types; file sizes; file age, shortcuts/symlinks	Available drives; folder age; hidden files, hidden folders; duplicate files (by hard link), duplicate folders (by hard link);
Organization (10 + 2)	Root folders; tree breadth, tree depth; folders in each folder (branching factor), files in each folder; file depths, folder depths; branch consistency or skewness; use of desktop for storage, use of default folders	Inaccessible folders; presence of user-excluded folders
Naming (5 + 2)	File or folder name*, length of name, numbers in names, punctuation or special characters in names, duplication of names	Letters in names, whitespace in names
Retrieval (2 + 2)	File access times, file modify times	Folder access times*, folder modify times

Table 1: a categorized summary of the 28 file system properties previously collected in FM research and 12 new properties; Cardinal collects 38 of these 40. *Names and folder access times are not collected.

Cardinal functions by iterating through the folder tree from user-specified starting points using Python's built-in *os.walk* function. To ensure that no sensitive or identifying data is collected, a list of folders that the participant wants excluded from data collection is consulted at each step and specified folders are noted but ignored instead of examined. For each location visited, Cardinal records the file and folder properties listed in Table 1 using the built-in *os.stat* function and other custom functions. For example, *os.stat* returns the size of files and the last time a file or folder was accessed or modified. Folder modify time is a previously unused property that is updated by the OS when the user adds or removes a file or subdirectory, or renames the folder; this may be used to better understand how users

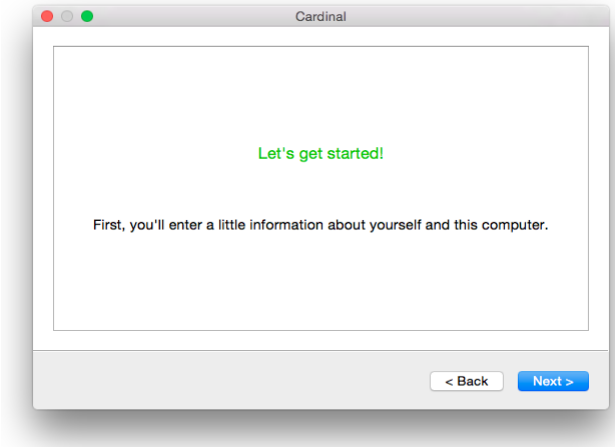

Figure 1. Cardinal's user interface: a 'sign-post' page greeting the user.

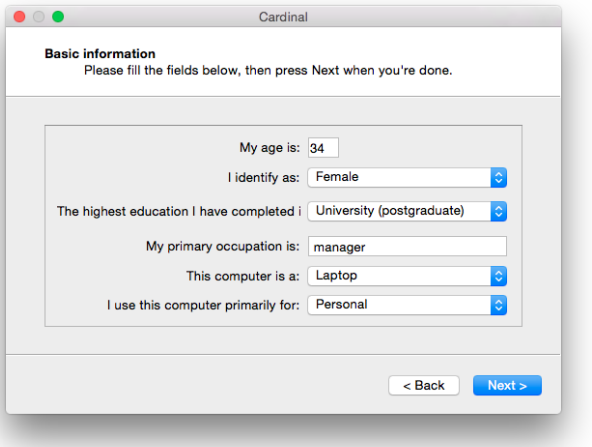

Figure 2. Cardinal's user interface: a page for the user to enter demographic data.

perform organizational or meta-level PIM activities (Jones, 2010). Folder access times are not recorded because this property is set to the current time by the OS at the moment Cardinal reads the contents of the folder.

Since we designed Cardinal to not store file and folder names, semantic measures are calculated and stored as each file and folder is examined, including the previously used properties of name length, use of numbers, use of special characters, and detection of duplication of names, but also records the use of letters and whitespace.

Other new properties collected include identifying files and folders that are hidden or duplicated across multiple hard links. Previous studies have examined how users manage duplicate files and folders (Hicks *et al.*, 2008; Henderson *et al.*, 2009) as identified by duplicate names. Files and folders can be duplicated in a number of ways; for example, by making a copy, maintaining two files with the same content and name, or by creating a hard link. Files are themselves hard links to some data on a disk, though additional hard links to that data can be made such that two files really provide access to the same content, or in other words, these two files really are the same file but the use manages its existence across multiple locations. Cardinal identifies when files and folders have been duplicated in this way by checking the *nlink* property in *os.stat;* a value greater than 1 entails duplication via multiple hard links to a file.

Hidden items have not been examined in prior PIM research, but may exist in the user's collection as a result of the user unintentionally downloading or explicitly hiding them, and require special attention to manage since FM display settings must be toggled to view them. To protect user privacy, Cardinal does not record properties about hidden files nor enter hidden folders, but it does note their existence and locations. To identify hidden items in Windows OS-provided file attributes are checked, while in Mac and Linux the file name is checked for a leading dot ('.') per the POSIX convention for marking hidden files.

Files and sub-folders are assigned to folders by ID so that further properties can be derived later, like tree topology (e.g., depth and breadth); in essence, a mirror of the hierarchical arrangement of files and folders is made. This means researchers can later make post-hoc measures of the mirror that would be impossible to derive from a flat list of files and folders. For example, rather than being limited to mean file size, derived from a list of file sizes, the distribution of file sizes or types across folder depths can be derived by examining the files where they are located across the folder tree.

Once the provided executable is downloaded and run, a simple interface (seen in Figures 1-4) walks the user through the following steps:

1. Greets the participant, outlines the process, and presents a consent form (Figure 1).
2. Asks for basic demographic information (age, occupation, education, gender) and the form (laptop, desktop, tablet, other) and use (work/school, personal, both) of the computer (Figure 2).
3. Asks for the names of installed software relevant to file management and suggests any likely values based on the OS detected (e.g., Finder for Mac, File Explorer for Windows).
4. Asks the participant to select folders that they personally manage, suggesting the user's home directory as one location.
5. Allows the participant to select folders that they wish to have excluded from data collection.
6. Allows the participant to initiate the examination of the selected folders, collecting file system data while ignoring file contents and file and folder names.

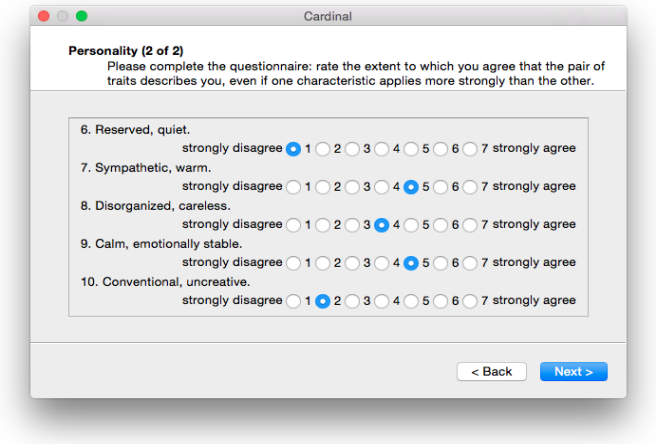

Figure 3. Cardinal's user interface: a page presenting an included questionnaire (example items from Gosling *et al.*, 2003).

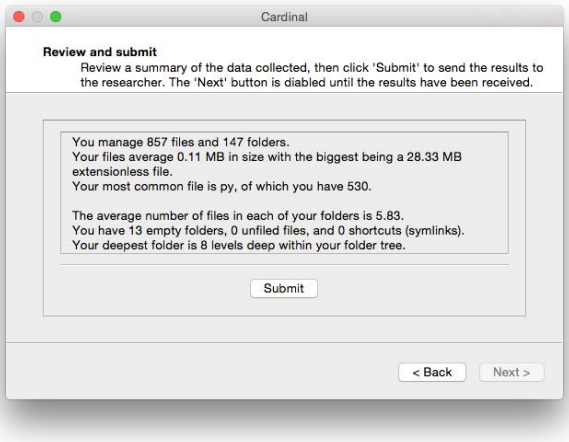

Figure 4. Cardinal's user interface: the results summary page. Further results are viewable by scrolling or enlarging the window.

7. Presents any included questionnaires to the participant (Figure 3).
8. Presents a summary of their collection and results of any questionnaires, and asks the participant to initiate submitting the collected data to the researchers (Figure 4).
9. Thanks the participant and exits the application.

To encourage participation, we aimed to make Cardinal simple and easy to use. For example, it appears 'native' on each OS to reduce unfamiliarity, requires little time of participants (specific measurements are presented in the next section), and is laid out sequentially, with back and next buttons and instructions on each panel. During development we employed a simple iterative design process by soliciting free-form feedback from five colleagues through email. Though all five users were able to make basic use of the software, two rushed through the pages without reading the instructions and then expressed feeling confused about what they were meant to do, so we inserted 'sign-post' pages containing summaries of what general task comes next.

All five users expressed concerns about privacy that arise from exposing their file collections, so we configured Cardinal to respect participant privacy: participation is anonymous as the identities of the participants are never known to the researchers, sensitive folders can be excluded from data collection, and identifying file and folder properties are respected as described above. Two users still noted feeling unsure about what the software had seen, so we added an instant summary of the results of their participation (e.g., their most common file type, the length of their longest folder name, and the number of empty folders), which they said alleviated their concerns enough to use the software. We also added a link to a Web page[1] displaying averages of the FM data collected thus far so that they could compare their own results. Though providing such a page remains an optional aspect of using Cardinal in future studies, it may encourage participation by making the data collection more transparent and meaningful to potential participants. Cardinal is also open-source software, thus some degree of trustworthiness is implied by the code being visible to a community of developers and open for interested participants to review for themselves.

As the software facilitates rapid distribution, recruitment can be tailored to reach the intended population, and any participants not meeting demographic criteria can be filtered out afterwards. For example, with the software hosted on a Website, traditional recruitment methods (e.g., fliers, emails, social media) may point to the page and participants can 'help themselves' to the software. Direct compensation is made impossible with anonymous participation, but participant identification could be added by including a free text field (e.g. for inputting email addresses), and internal motivation may come from the participants' desire to learn about their own FM behavior, which is summarized and reported to them at the time of data collection. Improved distribution and recruitment may make the software attractive also to computer science researchers, like those working on file system design and file-size distribution (Douceur & Bolosky, 1999), where small and niche population samples have been a research limitation as much as in PIM research.

To aid reusability, Cardinal was made using open-source tools[2], and we have shared its source[3] under a liberal license (GPL 3). Next we describe its use in a trial implementation.

[1] http://dinneen.research.mcgill.ca

[2] Python 3, the Qt graphics framework, and PyQt bindings

TRIAL IMPLEMENTATION AND SUBSEQUENT IMPROVEMENT

We implemented a pilot study to demonstrate a use case for Cardinal and test its efficacy as a data collection tool. We emailed 48 people (12 faculty and 36 PhD students) in our department leading them to a Web page explaining what participation entailed and containing links to download the software. Within 13 days, we received 21 responses (44%). In two following days we invited 82 master's students to participate, and received 25 responses (30%), resulting in 46 of 130 possible participants (35%). Collection was successful on both laptops and desktops running the 3 supported OSes (26 Windows, 19 Mac, and 1 Linux). In total, Cardinal collected data about 2.3 million files and 290 thousand folders, and recorded questionnaire responses and technological data (OS and FM software used) for each participant.

Time stamps were recorded each time a new page of the interface was accessed. Excluding two outliers discussed below, the mean time taken to complete a session was 10.6 minutes (SD = 7; min. 2.5; max. 33.4), of which an average of 7 minutes (66%) were spent reading the consent form, entering demographic data, and answering two questionnaires bundled within the software. The remaining time was passed collecting data about the file and folder collection and preparing a summary of the data. The former took an average of 1.86 (SD= 2.7) minutes, accounting for 17.5% of the time to complete a session, while the latter took an average of 1.69 minutes (15.9% of the completion time).

Participants' use of Cardinal was largely unproblematic: responding to the invitation email, two participants reported that using Cardinal was "a breeze" and "painless", and six participants reported finding their summarized results to be of interest, noting for example that they did not know they had so many empty folders or large files. Two issues in using Cardinal were identified during the trial. First, one participant was unsure if they should plug in external hard drives to be examined. This should therefore be clarified in the participation instructions of each study implementing Cardinal. Second, four participants stated that the software appeared unresponsive while collecting and summarizing data about large numbers of files. This was solved by putting the relevant processes on a separate processor thread so that the interface stays responsive while they are running.

Given that participation was done remotely and in a potentially wide array of software environments we expected that Cardinal may encounter some errors or fail to run in at least a few cases. Indeed, three participants had Mac OS versions that were too old to run the software at the time of the pilot. To remedy this, we compiled Cardinal in an older Mac OS X version, and it now runs on versions 10.8 and above, supporting 90% of the Mac OS X market.[4]

The outlying participation times for Cardinal were 1.25 and 12.75 hours. The participant with the longer time emailed us to explain that they left Cardinal running overnight to complete the results summarization, and analysis of the time stamps revealed that this took nearly all 12 hours of the completion time. This was the longest summary time by approximately 11.5 hours. The lesser outlying completion time was primarily due to 33.9 minutes of file system data collection. This was nearly twice as long as the second longest collection time.

These outliers are extreme and surprising given that neither collection was the largest one seen in our pilot study. Similar cases may arise in future data collection, so we attempted to decrease the time required to perform both the data collection and results summary phases. To speed up the data collection, *os.walk* was augmented with a function called *os.scandir*, which iterates through directories faster. We also revised our approach to generating a summary of the participant's results by deriving several measurements more efficiently.

To understand the impact of these changes, we analyzed a test collection consisting of 222,321 files and folders (5% larger than the largest participant collection) using both approaches. Where the old approach, using *os.walk*, took 11.5 minutes to collect data about the test collection and 56.3 minutes to summarize the results, the new approach, using *os.scandir*, took only 1.45 minutes to collect the same data (an 87.4% decrease in time) and the new summarization approach took just 1 second (less than 0.03% of the original time). This implies an improved data collection time of 4.3 minutes (down from 33.9) for the most outlying collection time, and an improved summary time of 12.9 seconds (down from 12 hours) for the most outlying summary time. Considering these improvements together, we can expect the mean time for future participants to complete participation to be approximately 7 minutes, rather than the 10.6 average from the pilot.

LIMITATIONS

A file management-based approach is only one among several for understanding personal information management. Others may examine physical representations of information, or examine digital organization but focus closely on cognitive- and context-related aspects. Nonetheless, the approach outlined here complements these, and has been used in many studies (at least thirty, in our count) to understand users' regular experience of managing, sorting, and navigating their personal information stored in files. Further, file system property data can be used together with such approaches, and as such

[3] https://github.com/jddinneen/cardinal

[4] https://www.netmarketshare.com/operating-system-market-share.aspx

we have included provisions in Cardinal for integrating standard cognitive and context-related instruments. For example, our pilot study included questionnaires related to personality style and spatial cognition, and it would be simple to include other instruments, questionnaires, or free text fields for user-reported data.

Another concern is that inferring FM behavior from the file system, rather than observing actions as they happen, may capture a limited selection of a user's FM behavior. For example, Cardinal can count the number of folders at the time of scan, but does not indicate if a user created and deleted folders beforehand, nor does it inform us about actions like renaming, moving, or sharing files. In other words, the data produced by Cardinal is a snapshot of a user's file system as it has been produced by their behavior leading up to any singular point in time. It is desirable to improve upon this limitation, as the importance of longitudinal data will grow as the prevalence of long-term personal information management increases (Jones *et al.*, 2016). This may be partially overcome, however, by repeated executions of the software by the same participant; the data would then together be longitudinal and could be analyzed as such.

Finally, since the default setting in Cardinal is to respect participant privacy by not recording file and folder names, the semantic analysis that will follow is limited to the specific properties measured during data collection: name length, number of letters, numbers, whitespaces, and special characters, and name duplication. This necessarily means that it will be difficult or impossible to identify naming conventions or understand the use of a folder based on its name. This is the price of participant privacy; though Cardinal may be modified to overcome this, it will likely make recruitment more difficult.

CONCLUSION

We have developed Cardinal to overcome the limitations of methods used in PIM and FM behavior research, specifically: narrow data collection caused by a small number of inconsistent measures, and small sample sizes caused by technological inflexibility, impractical administration requirements, and difficult recruitment. In a trial implementation of just 15 days, Cardinal collected FM behavior data along 38 file system properties and additional demographic and psychological data from 46 participants, and did so remotely, asynchronously, and across three OSes. This indicates it is a viable tool for FM research and an improvement over the previous data collection methods, and should scale well to facilitate longer collection periods over larger and more heterogeneous samples. Cardinal can therefore facilitate an understanding of FM behavior and provide insight into which aspects of FM are more important to support in the design of future PIM systems and services, in turn saving time and effort during the frequent task of managing files.

We are now using Cardinal in its first study by letting anyone interested in participating download it from our website; through running it they provide us with questionnaire responses and rich file system data so that we can explore the relationship between FM behavior and technological and psychological factors. This is just one possible use - studies seeking more control could, for example, use the file system data in conjunction with a structured task, such as extracting file access times and depths in the folder tree to guide prompted retrievals. We are happy to share Cardinal with other researchers and hope it will save time and effort in future PIM studies.

REFERENCES

Agrawal, N., Bolosky, W. J., Douceur, J. R., & Lorch, J. R. (2007). A five-year study of file-system metadata. *ACM Transactions on Storage (TOS)*, *3*(3), 9-24.

Barreau, D. (1995). Context as a factor in personal information management systems. *Journal of the American Society for Information Science*, *46*(5), 327-339.

Benn, Y., Bergman, O., Glazer, L., Arent, P., Wilkinson, I. D., Varley, R., & Whittaker, S. (2015). Navigating through digital folders uses the same brain structures as real world navigation. *Scientific reports*, *5*.

Bergman, O., Gradovitch, N., Bar-Ilan, J., & Beyth-Marom, R. (2013). Tagging personal information: A contrast between attitudes and behavior. *Proceedings of the American Society for Information Science and Technology (ASIS&T) 2013*, 1-8

Bergman, O., Whittaker, S., & Falk, N. (2014). Shared files: The retrieval perspective. *Journal of the Association for Information Science and Technology*, *65*(10), 1949-1963.

Bergman, O., Whittaker, S., Sanderson, M., Nachmias, R., & Ramamoorthy, A. (2012). How do we find personal files?: the effect of OS, presentation & depth on file navigation. *Proceedings of the 2012 ACM annual conference on human factors in computing systems*, 2977–2980.

Boardman, R., & Sasse, M. A. (2004). Stuff goes into the computer and doesn't come out: a cross-tool study of personal information management. *Proceedings of the 2004 SIGCHI Conference on Human Factors in Computing Systems*, 583-590.

Bruce, H., Jones, W., & Dumais, S. (2004). Information behaviour that keeps found things found. *Information Research*, *10*(1), paper 207.

Capra, R. (2009). A survey of personal information management practices. *Proceedings of the American Society for Information Science and Technology (ASIS&T) 2009, PIM Workshop*, 2-5.

Carroll, J. M. (1982). Creative names for personal files in an interactive computing environment. *International Journal of Man-Machine Studies,16*(4), 405-438.

Chernov, S., Demartini, G., Herder, E., Kopycki, M., & Nejdl, W. (2008). Evaluating personal information management using an activity logs enriched desktop dataset. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, PIM workshop papers.*

Douceur, J. R., & Bolosky, W. J. (1999). A large-scale study of file-system contents. *ACM SIGMETRICS Performance Evaluation Review, 27*(1), 59-70.

Evans, K. M., & Kuenning, G. H. (2002). A study of irregularities in file-size distributions. *Proceedings of the 2002 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*.

Gonçalves, D. J., & Jorge, J. A. (2003). An empirical study of personal document spaces. In G. Doherty & A. Blandford (Eds.), *Interactive systems. design, specification, and verification* (pp. 46-60). Berlin: Springer.

Gosling, S. D., Rentfrow, P. J., & Swann, W. B. (2003). A very brief measure of the Big-Five personality domains. *Journal of Research in personality, 37*(6), 504-528.

Hardof-Jaffe, S., Hershkovitz, A., Abu-Kishk, H., Bergman, O., & Nachmias, R. (2009a). How do students organize personal information spaces? *International Working Group on Educational Data Mining*, 250-258.

Hardof-Jaffe, S., Hershkovitz, A., Abu-Kishk, H., Bergman, O., & Nachmias, R. (2009b). Students'organization strategies of personal information space. *Journal of Digital Information, 10*(5), 1-17.

Henderson, S. (2005). Genre, task, topic and time: facets of personal digital document management. *Proceedings of the 6th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction: making CHI natural*, 75-82.

Henderson, S., & Srinivasan, A. (2009). An empirical analysis of personal digital document structures. In *Human Interface and the Management of Information. Designing Information Environments* (pp. 394-403). Springer Berlin Heidelberg.

Henderson, S., & Srinivasan, A. (2011). Filing, piling & structuring: strategies for personal document management. *Proceedings of 44th Hawaii International Conference on System Sciences (HICSS)*, 1-10.

Hicks, B.J., Dong, A., Palmer, R., & Mcalpine, H.C. (2008). Organizing and managing personal electronic files: a mechanical engineer's perspective. *ACM Transactions on Information Systems (TOIS), 26*(4), 23.

Jones, W. (2010). *Keeping Found Things Found:The Study and Practice of Personal Information Management.* Morgan Kaufmann.

Jones, W., Bellotti, V., Capra, R., Dinneen, J. D., Mark, G., Marshall, C., Moffatt, K., Teevan, J., & Van Kleek, M. (2016). For Richer, for Poorer, in Sickness or in Health...: the Long-Term Management of Personal Information. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems. ACM.*

Jones, W., Bruce, H., & Dumais, S. (2001). Keeping found things found on the web. *Proceedings of the tenth international conference on Information and knowledge management*, 119-126.

Khoo, C., Luyt, B., Ee, C., Osman, J., Lim, H.-H., & Yong, S. (2007). How users organize electronic files on their workstations in the office environment: a preliminary study of personal information organization behaviour. *Information Research, 12*(2), paper 293.

Kljun, M., Mariani, J., & Dix, A. (2015). Transference of PIM research prototype concepts to the mainstream: successes or failures. *Interacting with Computers, 27*(2), 73-98.

Lansdale, M. W. (1988). The psychology of personal information management. *Applied Ergonomics, 19*(1), 55-66.

Malone, T. W. (1983). How do people organize their desks?: Implications for the design of office information systems. *ACM Transactions on Information Systems (TOIS), 1*(1), 99-112.

Massey, C., TenBrook, S., Tatum, C., & Whittaker, S. (2014). PIM and personality: what do our personal file systems say about us? *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, 3695-3704.

Trullemans, S., & Signer, B. (2014). From user needs to opportunities in personal information management: A case study on organisational strategies in cross-media information spaces. *Proceedings of 2014 IEEE/ACM joint conference on Digital libraries (JCDL)*, 87–96.

Xie, X., Sonnenwald, D.H., & Fulton, C. (2015). The role of memory in document re-finding. *Library Hi Tech, 33*(1), 83-102.

Zhang, H., & Hu, X. (2014). A quantitative comparison on file folder structures of two groups of information workers. *Proceedings of 2014 IEEE/ACM Joint Conference on Digital Libraries (JCDL)*, 485-486.

APPENDIX

The following is a portion of raw data collected by Cardinal. The data here describe a pilot participant's computer, files, and folder hierarchy. For brevity, only one hard drive, folder (node), and file are shown. Demographic data, a list of installed file managers, questionnaire responses, and time stamps are not included in this example. Comments (#) have been inserted at the ends of some lines for further explanation.


```
"computer_description": {
    "form": "Laptop",
    "use": "Personal AND Work/School",
    "operating_system": "darwin",   # Darwin is the Mac OS X platform name
    "version": "10.10.5"
},
"drives": [{
      "disk_code": "/dev/disk1",
      "size": 122880.0,    # Figures are in megabytes; this is a '128 GB' drive
      "used": 63488.0,   # This drive is filled roughly half way to capacity
      "free": 59392.0
        }],
"node_lists": [
  {   # Begin describing folders on the first hard drive encountered
    "1": {    # Begin describing the first folder encountered
      "node_id": "1",    # Each node is given an ID to identify it since names are not stored
      "depth": 0,    # This folder is the root folder at the top of the tree
      "hard_link_duplicate": false,    # This folder is not present in the tree twice via a hard-link
      "c_time": "2015-11-30 11:06:23",    # This folder was created in November of 2015…
      "m_time": "2015-11-30 11:06:23",    # …no files or folders have been added or removed since
      "default": true,    # Name matches a list of default folders for Mac OS
      "name_duplicate": false,    # No other folders have the same name
      "name_length": 11,    # The folder name is 11 characters long
      "letters": 9,    # The folder name contains 9 letters…
      "numbers": 2,    # …and 2 numbers
      "special_chars": 0,
      "white_spaces": 0,
      "hidden_children": 0,    # No hidden folders within this folder
      "unknown_children": 0,    # No inaccessible (e.g. system) folders within this folder
      "children": ["2",],    # IDs of sub-folders in this folder
      "hidden_files": 2,    # There are two hidden files within this folder
      "symlinks": 0,    # There are no symlinks or shortcuts in this folder
```

```
        "file_list": [     # A list of files present in this folder.
          {
            "file_id": 1,
            "extension": "pptx",     # This is a Powerpoint file
            "file_size": 70636,    # File size is in bytes; this file is ~70 KB
            "hard_link_duplicate": false,
            "name_duplicate": false,
            "full_name_length": 46,    # This includes the extension and separating dot (e.g., ".pptx")
            "letters": 35,
            "numbers": 0,
            "special_chars": 2,
            "white_spaces": 4,    # This file has four spaces in its name
            "c_time": "2015-09-19 19:18:01",    # This file was created in September 2015
            "m_time": "2015-09-19 19:18:01",    # and hasn't been modified since creation
            "a_time": "2015-12-13 14:26:53"    # but was last accessed in December, 2015
             }    # additional files would be listed here
        ]    # end file_list
    }    # end description of the first folder, additional folders would be listed next
  } # end the first node_list (hard drive), additional hard drives would be listed next
]    # end node_lists
```