An Empirical Study of Personal Document Spaces

Daniel J. Gonçalves[1], Joaquim A. Jorge[1]

[1] Computer Science Department, Instituto Superior Técnico, Av. Rovisco Pais,
1049-001 Lisboa, Portugal
djvg@gia.ist.utl.pt, jorgej@acm.org

Abstract. The way people use computers has changed in recent years, from desktop single-machine settings to many computers and personal assistants in widely different contexts. Personal Document Spaces (PDSs) now tend to span several machines or *locii*. Moreover, the types and numbers of documents users manipulate have also grown. The advent of pervasive computing will reinforce this trend. In order to develop new approaches to help users manage their PDSs, we must have an idea of what documents they contain are and how these are organized across several *locii*. We performed an empirical study where the PDSs of eleven users were analyzed in depth, allowing us to extract a thorough characterization of those PDSs, both in terms of structure and contents. With these results in mind, we suggest several guidelines for the development of user interfaces.

1 Introduction

In recent years, computer hardware has become increasingly cheap. This made information gadgets accessible to large numbers of households. Nowadays, typical employees use computers not only at work, but also at home and, in some cases, laptops or PDAs. The advent of ubiquitous, pervasive computing will only increase the number of devices available to work on or access documents for any given user. Users edit and store their documents in an increasing number of locations. All the locations where the user has stored documents will be generically referred to as *locii*. The set of all documents accessible to a user in all *locii* constitutes his Personal Document Space (PDS).

Users' PDSs are becoming increasingly large and complex. Not only do they span a number of different *locii*, but the number and diversity of documents in store are increasing. PDS's are no longer organized as a single hierarchy of documents, but rather, as a polyarchy, for which traditional ways of document handling are not effective. New tools that allow users to more easily find a specific piece of information (regardless of location), or to visualize the PDS as a whole will soon become an imperative necessity. One of the major challenges of HCI in the upcoming years will revolve around these issues, as pervasive computing becomes a reality [10]. In fact, we have witnessed in recent years, an increasing concern on the issues the new interaction modes will bring about [1][2][16]. The increasing complexity of PDSs has also become of concern in recent years. Large numbers of documents coupled also become of concern in recent years. Large numbers of documents coupled with distributed placement worsen cognitive load problems while requiring new techniques for archiving and retrieving information [4][6][17].

To correctly address those problems, it is important to know beforehand what the actual characteristics of a PDS might be. For instance, not all information visualization techniques are suited for all kinds of structures. Broad and shallow hierarchies are better visualized by some techniques, while others are better suited to handle narrow and deep ones. Most techniques are also limited in the number of elements they can display. Short and long-term memory problems in remembering the location and contents of documents will become more serious as the complexity of PDSs increases. Knowledge of both the structure and contents of PDSs is of capital importance for the user-centered design of new techniques that provide answers to new needs posed by their size and complexity.

Some studies undertaken in the past tried to understand how users store their documents and organize personal information. Malone [14] established the groundwork for early research regarding the organization of documents in personal spaces (such as described in [21] and [7]). This seminal study identified specific modes of interaction and organization providing a solid foundation to new approaches to managing office documents.

Gifford et al's Semantic File Systems [9], where properties are associated with files allowing users to organize and retrieve them with the help of those properties (trying to effectively deal with growing file numbers) inspired some research on that field. The works of Baeza-Yates et al [3] and the Placeless Documents approach by Dourish et al [5] share that idea. Others, such as Freeman and Gelernter [7] provide different approaches for navigating in PDSs, presenting documents in chronological order. Finally, some works, such as Lamming et al's Satchel [13] directly tackle the problem of managing documents across several *locii*.

None of these otherwise excellent works has, however, been based on a thorough characterization of the PDSs they handle and strive to present in straightforward and meaningful ways. Such system's usefulness and scalability directly depends on their adequacy to the PDSs they must handle. Some, such as Lamming [13], discuss the need for different strategies to handle large numbers of files. The best way to verify if those numbers should actually be taken into account is a user-centered study. That will yield an assessment of users' real needs, and provide tools to better address problems brought about by the increasing heterogeneity and distribution of *locii*.

To this end, we conducted a study where the PDSs of eleven users were extensively analyzed. The background of those users ranged from college faculty and students (seven users) to IT-related workers. The results of that analysis provide interesting insights of the surveyed PDSs' contents. Some patterns with direct implications for the development of PDS-handling applications were extracted, in terms of PDA organization and of document type and distribution.

In the following section, we'll start by describing how the study was conducted. Next, we analyze the results thus obtained. We then discuss the main results, extracting guidelines relevant for user interface design. Finally, we'll present the work's main conclusions and point to possible future work on this area.

2 The Analysis

We developed a computer program, coded in Python, to analyze PDSs. It can be run in every *locus* in a PDS (all the machines a user stores documents in, for instance) and then aggregate the results for the PDS as a whole. This program needs, as input for each *locus*, a list of directories located somewhere in the user's disk. It then traverses all those directories and their sub-directories, collecting all kinds of information. In particular, it gathers information on the number of files and sub-directories on each directory, establishing the size of PDSs and the distribution of their contents. Also collected are directory tree measures, such as its branching factor, to provide an estimate of PDS topological structure. Numbers and sizes of files by class provide an insight on the nature of contents. The collection of statistics on the dates and times of creation, access and modification of all files allow the discovery of PDS parts not used for a given period of time. File sizes and file distribution by class complete the description of PDS contents. Finally, an analysis of elements that make up the names of files allows us to extract naming conventions and patterns.

The program stores this information in intermediate files, for each *locus* in a PDS, where relevant statistics are presented. Users have to move the intermediate file produced on a *locus* to the next, to produce global statistics on the entire PDS. After the last *locus* is analyzed, the final report is produced automatically, in a human-readable format. We chose this format to allow users to inspect the file before returning it to us. We hoped this would ease their minds regarding privacy concerns. At all times subjects had absolute control over their data, and the option of not sending in the results. These were to be sent by email rather than automatically by the script, to allow such control. Also out of privacy concerns, no information on a single file was ever recorded on the report. Only aggregate data on each directory was collected.

The program was made available on the World-Wide-Web (currently at http://www.gia.ist.utl.pt/~djvg/phd/resources.php), together with instructions of use, and a description of report file format. The program was provided pre-compiled for several architectures (namely, Windows and several flavors of UNIX/LINUX), and required no special installation process (unpacking the archive sufficed to run it). Thus, we prevented alienating users without administrator or super-user privileges that would be unable to perform a full installation. Also, target architectures were not chosen at random. Rather, a previous study [10] showed they were by far the most used architectures (99.5% of *locii*) among test group users.

3 Procedure

Users were instructed to feed into the program only those directories that containing actual documents, and not directories that contain operating system, applications or system-generated data (such as '/usr' and '/var' directories on UNIX systems or the 'Program Files' folder on Windows machines). We were interested in the user's Personal Documents Spaces and not in the entire contents of their machines.

A call for participation was posted among the faculty and students of Instituto Superior Técnico's computer-science department, and to several users with unrelated jobs that had participated in previous studies [10]. After a two-week period on the end of July/beginning of August of 2002, we had received eleven reports whose analysis we'll present in the next section.

In the previous (questionnaire based) study, the total number of participants was of 88. We directly contacted over 120 for the study here described (including the aforementioned 88). The relatively low 10% participation rate (and taking into account most users had eagerly participated in the previous study) shows that, despite our best efforts to ensure privacy and program ease of use, no manner of persuasion was enough to allow people to relinquish the privacy of their machines. Some users were personally contacted and stated an outright refusal to participate. This was the greatest barrier to our study. Hence the two weeks it took to gather a number of reports deemed sufficient to allow a thorough examination of the results (each report was carefully studied) and at the same time to extract patterns and statistically relevant results. Whenever thought necessary, oral interviews with individual participants were conducted to clarify some aspects of their PDSs.

3.1 User Profile

Of the eleven participants, three had jobs where they use computers on a regular basis as a work tool (their areas are, mainly, database design and project administration). Another participant was a senior manager for a small software development company. The remaining seven participants were either computer science faculty or graduate students in the field (four and three, respectively). We purposefully tried to collect data from users with different backgrounds, to get a more general view of interaction habits.

4 Results

In what follows, all users have been numbered to allow us to individually reference them while maintaining complete confidentiality. The statistical results presented here reflect just part of the measurements and techniques applied to the data (including among others the standard error of the mean to estimate adequacy of data to the universe of study).

4.1 Number of *locii*

On average, the number of *locii* of each PDS is 1.45 (std.dev.=0.52). This was a rather surprising result, as a previous study [10] placed this value at 2. The comparison of both datasets and the oral interviews showed that, indeed, some users do have other *locii* on their PDS, but were unable to run the program on the relevant architectures (notably, PDAs and a flavor of UNIX not provided for). Mainly, we found that some of the "*locii*" mentioned on the previous study aren't part of PDSs after all. They relate to machines or working areas where users log on from time to time, and where some documents are stored (mainly work-related), but that, in practice, are seldom used.

In short, accounting for the *locii* where the script could not be run, we found that about 30% of PDSs only have one *locus*, 60% have two and the remaining 10% have three or more. Some untested *locii* were PDA-based, reinforcing our assumption that ubiquitous computing is becoming a reality. However, present day PDAs still lack capabilities to be expected on full-fledged computers. Documents as such are not generally stored and manipulated in PDAs. Rather, PDAs are used primarily for their personal information management capabilities (date and address book, etc.). This data, supported by previous results [10], and the fact that in or sample only 2 users had PDAs, allows us to conclude there wouldn't be significant differences to the results had they been included. Technology has yet to mature before relevant data can be collected.

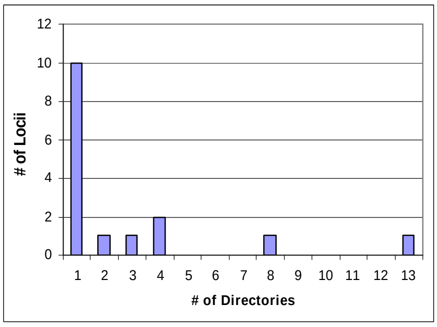

Figure 1. Number of high-level dirs. by *locii*

4.2 Number of High-Level Directories

The number of high-level directories in each *locus* (the directories provided to the program by users) was found to be 2.75 (std.dev.=3.23). However, most *locii* had only one high-level directory, as shown on Figure 1. In fact, with the exception of two *locii*, we find that the results are consistent, showing that documents tend to be concentrated on a reduced number of directories for each *locus*.

4.3 File Numbers

The number of files on the several PDSs varied a lot. Although the average value is 7940, the standard deviation of 8739 confirms this. We identified three categories of users: *file-rich* users had between 10,000 and 25,000 files on their PDSs; *file-*

average users have between 1,000 and 10,000 files, and *file-poor* users present values inferior to 1,000 (in practice, around 500 or below). It seems user occupation plays a determinant role in the category it belongs to. In our sample, all file-rich users were teachers, and all file-poor users worked outside academia.

Comparing the number of files and high-level directories showed no connection between the two. A higher number of high-level directories do not imply a larger number of files.

4.4 Number of Sub-Directories

The number of sub-directories also displayed significant variation. However, there is a relation between this number and the number of files on the PDSs. We found that, on average, there are thirteen files on each directory (std.dev.=6.2). This shows users tend to manage the complexity of their PDSs separating and classifying documents whenever possible. Even assuming some directories contain no files and exist solely to group related sub-directories, values remain on the order of a couple of dozen files per directory. While directories generated or managed automatically by applications tend to have large numbers of files, this is clearly not the preferred user way of organizing them.

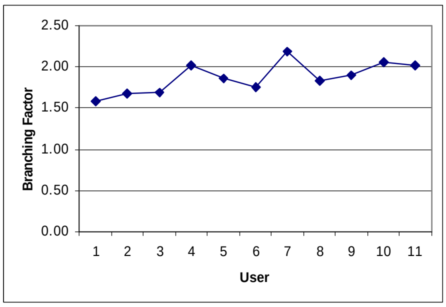

Figure 2. PDS Branching Factor

4.5 Branching Factor

To understand the structure of directory trees of PDSs, we computed their branching factor, defined as the average number of sub-directories of a given tree level. This is the number b such that $N=b+b^2+...+b^d$ (N being the total number of sub-directories and d the depth of the tree). "Virtual" tree levels were considered for the top-most level of each *locus* and for the PDS as a whole. In practice, the 0-th level's branching corresponds to the number of *locii*.

We found an average branching factor of 1.84 (std.dev.=0.187). Individual values were extremely regular, as depicted on Figure 2. We can therefore conclude that, in general, directory trees are narrow rather that wide. It is worthy of notice that this branching factor was found regardless of file numbers of files and sub-directories or user occupation.

4.6 Tree "Skewness"

While the average branching factor was, as seen, fairly low, it gives no indication on whether a tree is balanced, i.e., all its parts have similar branching factors. To evaluate this, a measure of unbalance was computed, defined as the standard deviation of the branching factor on each PDS sub-directory. Lower values correspond to more balanced trees. The average value was 3.61 (std.dev.=0.67). To better understand this value, we took a closer look at "skewness" deciles for each directory in each PDS. We found that (on a rather consistent way across all studied PDSs) up to 40% of directories only have one sub-directory, 20% have two sub-directories, 10% have three and 20% four to nine. Only the last 10% present superior values, up to around 20 (although, on one case, a value of 62 was registered). These numbers show that, not only are directory trees rather narrow, but also fairly well balanced, with the exception of 10% or so of directories.

4.7 Tree Depth

The depth average of PDS directory trees is 8.45 (std.dev.=2.9). This shows users strive for medium-depth trees.

4.8 File Classes

Files types were identified looking at their extensions. About 350 common extensions were included in the study. Extensions that might be associated with several applications were not considered ('.DAT', for instance, does not unequivocally identify the kind of file it is attached to). After identifying its type, each file was classified into one of 20 classes: Text, Image, Spreadsheet, Database, Presentation (such as Powerpoint files), Personal Information Management (PIM), Shockwave files (a class on its own given the latitude of things it can actually contain, and not included into 'executables' since they usually exist on Web-Pages), Web Script, Bookmarks, Video, Audio, Executable, System, Source Code, PDA-related, Archive, Backup, Symbolic Link, Files with No Extension and Files of Unknown Type. Those classes cover a wide range of commonly used application kinds.

On average, only 3.95% of files had no extension, even considering that some of the considered *locii* were UNIX-system based, where extensions are not required. Thus, extensions seem to be a valid hint of the type of a file.

Of all files with extensions, 87.7% were, on average, identified. This value, however, takes into account data for User 9, that has an abnormally high percentage of unknown files (42.76%). The next worst value was of 20%, and most users remained between 5% and 10%. A detailed analysis of the data showed those unknown files to be automatically generated files, directly related with the user's research. Excluding this user, the percentage of recognized files rises to 91.6%, a good value considering we used only extensions for which there was little or no ambiguity regarding the corresponding file types. Globally, we found that most unidentified extensions were either numeric or some general-use extension (such as '.DAT'), impossible to unequivocally associate to a given application domain.

Some file classes occur more often than others. As depicted on Figure 3, all users had Text, Image, Archive, and Executable files on their PDSs (and nearly all had Database, Spreadsheet or System files). Others classes, such as Symbolic Links, PDA-related, PIM or Shockwave files, are seldom used.

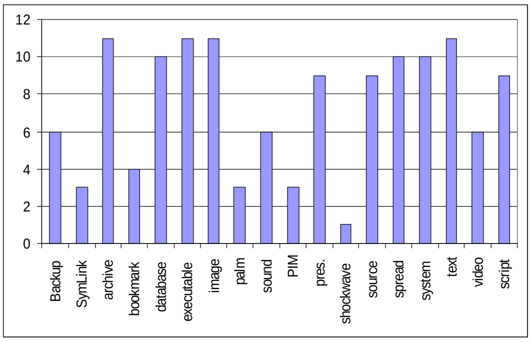

Figure 3. Number of PDS by File Class

The notable absence of Bookmark files is easily explained by the fact that bookmarks tend to be created and managed by specific applications, such as Web Browsers, and are thus stored by those applications on special-purpose directories, outside the PDSs. A similar explanation might account for the low numbers of PIM and PDA-related files, often stored on special directories of their own.

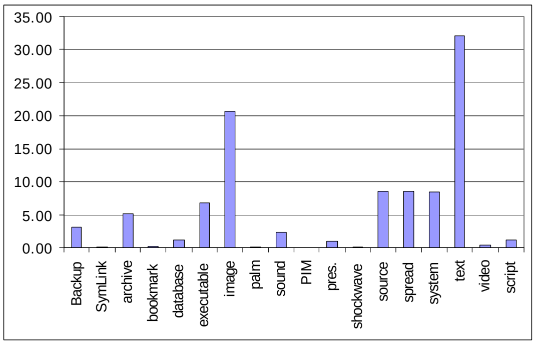

Figure 4. Average Percentage of PDS occupation by # of files

4.9 PDS Occupation by Number of Files

The average percentage of files of each class throughout the PDSs is depicted on Figure 4. We immediately notice that most files are either Text or Image files. With averages of 32.13% and 20.7% respectively, those classes include more than 50% of files on the PDS. An individual analysis of data shows that in only three cases they aren't the most common (although still accounting for a significant PDS portion). On one of them, the most common class was 'System'. Closer inspection showed those files to be temporary files left behind by some application (thus, not created explicitly by the user). On the two remaining cases, Spreadsheets and Audio were the most common, the first due to work-related applications, and the second for entertainment (that user likes to listen to some music while working). As already stated, even in those cases Text and Image files were widely used. In some PDSs those classes account for around 70% of all files.

Also interestingly, Symbolic Links (or its Windows equivalent, shortcuts) are rarely used. Only 0.06% of all files belong to this class (this was a consistent result across all PDSs). Sometimes referred to as a possible solution to multiple classifications of documents problems, in practice users don't bother with their creation.

Given the recent increase in support for multimedia formats, both by applications and operating systems, a fairly large number of those files were expected to appear on PDSs. However, those files accounted for only about 3% of the total number of files. This is a low value, even for the particular user group of the analysis. We propose it might be due to those file's usual large size and consequent difficulty in transferring them to the PDS, given the low capacity of most recordable media and low bandwidth Internet connections. Also seldom used are PIM files and PDA-related files. Although support for a wider range of applications is now available, users show a fairly high resistance to change of their work habits.

Only about 7% of files are archives (mostly compressed). The increasing availability of high-capacity hard drives has reduced the pressure on users to save disk space by compressing and archiving documents.

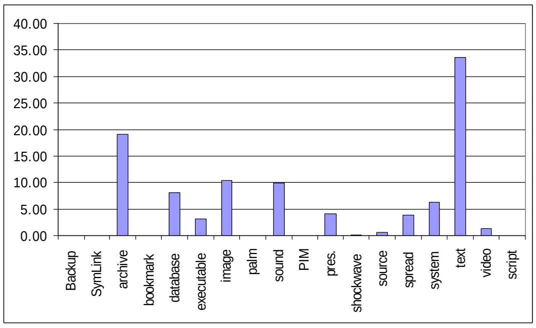

Figure 5. Average Percentage of PDS Occupation by file size

4.10 PDS Occupation by File Size

Comparing file sizes for each class in relation to total sizes of PDSs, we found the results represented on Figure 5. Interestingly, although Archive files account for only 7% of all files, they occupy 19% of PDS size. This shows that most such archives are fairly large and probably contain large numbers of files

Files of classes like Backup, PDA, PIM, Shockwave, Source Code and Scripts have almost no expression in terms of occupied size. Excluding Archives, the files that occupy more space are Text and Image files, as expected since they are by far the classes that appear the most.

Analyzing average file sizes for each class, we found that the largest files are Archives, Video and Audio files (a few Mb). Of medium size we found Image, Text, Executable and System files (hundreds of Kb). The remaining classes consist of fairly small files, with the exception of databases (more on this below). The difference in magnitude of average size values, even taking into account high standard deviations, allowed us to establish the ordering we just presented. The only class not included in the ordering is Database files. On average, they are the largest, but the standard deviation is so large (20Mb for an 11Mb average) that they could not be unequivocally positioned in the ordering. Oral interviews performed with the participants who had database files on their PDSs explained this size variation: some users handle large databases as part of their work, and maintain fairly small ones for private applications (managing their collections, for instance).

PDS Activity

We already discussed the number of files on each PDS. We've also tried to find how many of them were, in fact, used on a daily basis. To that end, we collected date and time information on all files' creation and last access and modification (Table 1).

Table 1. % of files manipulated by time frame

	1 day	**1 wk.**	**1 mo.**	**3 mos.**	**6 mos.**	**1 yr**	**> 1 yr**
Create	0.9	1.7	6.0	36.7	51.8	66.0	100
Access	1.7	18.1	36.4	49.0	64.7	82.8	100
Modify	0.3	1.2	3.7	15.6	29.1	45.3	100

Most files are not used daily at all. A quick inspection of the table shows that only 6% of files were created in the last month, and only 3.7% were modified in that period. In fact, only 66% of files were created in the past year, and 45.3 were modified in the same time span. Access dates are less trustworthy because there are lots of ways in which files can be accessed without in fact being consulted by users (such as automated file search mechanisms, or our own data-collecting program). We recorded the data anyway. Even considering that the dates have probably been distorted, only 82.8% of files were accessed in the past year.

We should also notice that some values are somewhat larger than what was found for the majority of PDSs. A couple of PDSs presented patterns that greatly differed from the average. As Figure 6 depicts (the graphic shows Modification dates, but a similar trend was found for the others), values tend to be lower than average. The outliers were even more influent in access dates. Removing them, we find that only 18% of files were accessed in the past month (rather than the 36.4 on the table above). In short, about 80% of PDSs are inactive (not used for about a month) at any given time. When developing applications to help users index and retrieve their files and cope with both memory issues, we should remember this number.

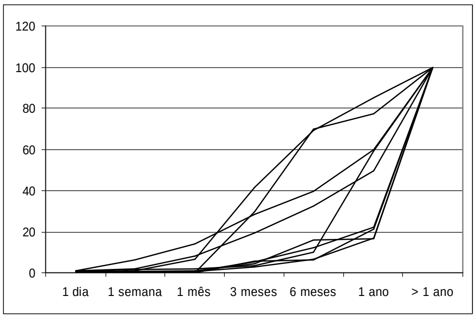

Figure 6. Percentage of files by modification date by PDS

4.11 Directory Occupation

We tried to evaluate to what extent files of each class occupy the directories they are stored in. Image, Text, Source-code and Spreadsheet documents tend to be the main occupants of their directories (they account for 50% of all files therein). Other classes have occupation averages around 30%, with high standard deviations. The only conclusion we can reach about them is that they don't dominate in the directories they are in. We performed a similar study in terms of file size, but no significant pattern was detected.

4.12 File Names

We analyzed the names of all files and classified their constituent elements (apart from alphabetic characters we assumed all file names contained) into six different classes: Numbers (excluding date numbers), Dates (extracted in a wide range of formats with the help of regular expressions), Hyphen, Underscore, Space and Non-Alphanumeric characters (apart from those already mentioned and including accented characters).

As can be easily seen in Figure 7, the most common elements are numbers. Nearly 60% of file names contained them. Next, we find that 21.7% of files have underscores on their names, and only 7.5% and 6.3% of files have hyphen and spaces, respectively. Taking a closer look at the data, we discovered a rather interesting pattern: users that name their files resorting to underscores seldom use spaces and hyphens, and vice-versa. Non-alphanumeric characters are used in 8% of file names. Dates are notably absent. In fact, only 0.33% of files have them in their names.

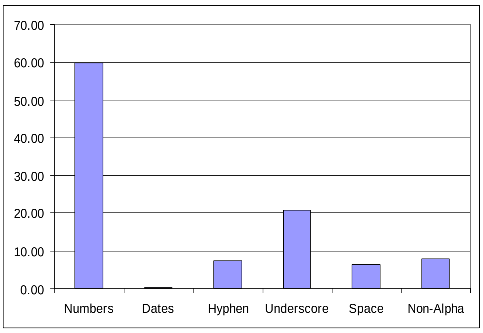

Figure 7. File name elements

As for the size of file names, the average value is 12.56 (std.dev.=8.13). This value seems to be a legacy from when file names were limited to 8 characters on some operating systems. The seemingly high standard deviation is due to some extreme values. The names for 90% of files have lengths similar to the average value. Of the remaining 10%, most are about 20 characters long. Only the largest files have significantly higher values (between 50 a 100 characters).

5 Discussion

We just presented some patterns and properties of PDSs in general, inferred from the results despite some significant value variations. Those patterns lay the foundations upon which new PDS-handling applications can be built.

Innumerable research studies in the area of Information Visualization have concerned themselves with how to visually present file or document hierarchies. The TreeMaps approach tries to cope with large hierarchies and file numbers by displaying an overall representation of the entire hierarchy. It is not designed to identify individual documents, but good to visualize the hierarchy's global properties [12]. On the other end of the spectrum, we find techniques such as Data Mountain [19] where, due to screen real-estate, only comparatively low numbers of documents can be displayed. In between we find a wealth of different techniques, ranging from zoomable interfaces and fish-eye views [8], to Hyperbolic and Cone Trees [14][20]. A major concern in all these techniques is their degradation with extreme aspect ratios or large numbers of objects. For instance, Cone Trees are prone to cluttering problems with over 1000 nodes, and known to handle up to 10 layers without problems [20]. Likewise, some hierarchies lead to TreeMaps with rectangles of extreme aspect ratios, requiring special care to prevent inelegant and hard to understand maps [22]. Our results indicate that, to display Personal Document Spaces (and not the entire file system), visualization techniques need not concern themselves, in general, with such extreme aspect ratios. PDSs have narrow and not too deep tree structures and visualization techniques can be tailored for that reality. Likewise, file numbers in each directory are usually not too large. This suggests that they can all probably be shown at the same time. Also, we found that certain kinds of documents (images and text, for instance) usually occupy most of the directories where they are stored. Special visualizations of those directories (according to document type) could be considered.

A problem that none of those approaches addresses, however, is the representation of polyarchies. Indeed, we have seen that PDSs are starting to span several *locii*. Furthermore, the documents in those *locii* can be related to others in the PDS, regardless of their location. The development of techniques that allow the visualization of all *locii* in an integrated way, taking advantage of such relations, constitutes an important research area.

Even if PDS visualization and browsing are possible, they will not allow specific documents to be easily found. Total file numbers can reach the tens of thousands and filenames tend to be short, providing little information on the files' contents, making it difficult to identify at a glance. Simply relying on the user's memory on where in the PDS a document was stored is not effective with these kinds of numbers. Novel ways of managing and retrieving documents should, thus, be considered. One possibility are property-based approaches such as PACO [3] and the Placeless Documents [5]. Their need is reinforced by the realization that symbolic links are seldom used, even if classification problems persist. Techniques such as those, which provide alternate ways of organizing documents, are sorely needed, as user-computer ratios have reversed themselves in the past years and the number of computing devices at the disposal of users continues to grow. However, relying solely on properties will shift the memory load from remembering a document's classification and location to remembering arbitrary sets of properties and possible values.

An interface that allows users to freely 'tell the story' of a document will solve this problem. Humans are natural born storytellers and by relating important information elements in a story, they will be more easily remembered. Since we are on the verge of the arrival of ubiquitous computing, additional information, not only about the documents themselves but also contextual and auto-biographical can be gathered and will be crucial for a more natural, efficient document retrieval. The discovery of access and reading patterns and the automated retrieval of documents, inferring user needs by monitoring their actions, will also become more of a necessity as PDSs grow both in diversity and complexity.

Special support for managing texts and images should be considered, given that those are most commonly found file classes. This includes tools for automatically managing different versions of documents across *locii,* The abilities to look into a text document's contents and to find images from rough descriptions or sketches of their appearance should be considered. Archives should also receive special treatment, by inspecting their contents and allowing those to be handled like other documents in the PDSs. The large numbers of files contained in archives make this feature a necessity.

Finally, since most files in PDSs (up to 80%) are not active at any given time, PDS browsing, visualizing and organization tools should concentrate in providing easy access to active files. There are important implications for temporal-based approaches, such as Lifestreams [7], given that PDS activity is directly related with document age.

6 Conclusions and Future Work

We provided an in-depth description of several relevant aspects of typical modern Personal Document Spaces. We took into account recently acquired usage patterns (several *locii* for each PDS, managing documents between those *locii*, etc.). Thus this study is a valuable tool to help overcome some challenges HCI and application design will face in the upcoming years, as those usage patterns in particular, and ubiquitous computing in general, become more of a reality.

While some results confirmed our expectations, others were rather surprising. We found PDS tree structure to be narrow and not too deep, while, at the same time, fairly balanced. Only around 4% of files have no extension. Of those that do, we identified 90%, and showed text and image files are by themselves responsible by more than 50% of PDS occupation. 'New' formats such as multimedia files are still not generally used by users in the study group, despite all the recent hype on multimedia systems and applications. We also confirmed our expectations on the infrequent use of Symbolic Links and shortcuts. As for the activity status of PDSs, only about 20% is active at any given time. Numbers are often used while naming files. Dates are rarely found.

In the future, we plan to repeat this study to gather more evidence concerning mobile devices. That will allow us to have an idea of the evolution of the patterns herein described, and provide an updated description of PDS structure. We'll strive for a wider range of audience (both in number and diversity). This will require using new ways to motivate users and alleviate privacy concerns, one of the major barriers to this kind of studies. If technology has in the meantime matured, the retrieval of information about PDA – and other mobile devices–based *locii* should be included in the study. Another aspect that should warrant some attention is the discovery of different versions of the same document, with slightly different names and contents, and perhaps on different *locii*.

References

1. Abowd, G.: Software Engineering Issues for Ubiquitous Computing. Proceedings of the 21st international conference on Software engineering. ACM Press (1999) 75-84
2. Abowd, G. and Mynatt, E.: Charting Past, Present, and Future Research in Ubiquitous Computing. ACM Trans. on Computer-Human Interaction, 7(1), ACM Press (2000) 29-58

3. Baeza-Yates, R., Jones, T. and Rawlins, G.: A New Data Model: Persistent Attribute-Centric Objects, Technical Report, University of Chile (1996)
4. Barreau, D. and Nardi, B.: Finding and Reminding: File Organization from the Desktop, ACM SIGCHI Bulletin, 27(3), ACM Press (1995) 39-43
5. Dourish, P. et al.: Extending Document Management Systems with User-Specific Active Properties. ACM Transactions on Information Systems, 18(2), ACM Press (2000) 140-170
6. Fertig, S., Freeman, E. and Gelernter, D.: "Finding And Reminding" Reconsidered, ACM SIGCHI Bulletin, 28(1), ACM Press (1996)
7. Freeman, E. and Gelernter, D.: Lifestreams: A Storage Model for Personal Data, ACM SIGMOD Record,25(1), ACM Press (1996) 80-86
8. Furnas, G.: Generalized fisheye views. Conference proceedings on Human factors in computing systems. ACM Press, Boston, Massachusetts, United States (1986) 16-23
9. Gifford, D., Jouvelot, P., Sheldon, M. and O'Toole, J.: Semantic File Systems. 13th ACM Symposium on Principles of Programming Languages (1991)
10. Gonçalves, D.: Users and Their Documents, Technical Report, Instituto Superior Técnico, (2002)
11. Hewagamage, K. and Hirakawa, M.: Situated Computing: A Paradigm to Enhance the Mobile User's Interaction. Handbook of Software Engineering and Knowledge Engineering, World Scientific Publishing Company (2000).
12. Johnson, B. and Shneiderman, B.: Treemaps: a space-filling approach to the visualization of hierarchical information structures. Proceedings of the 2nd International IEEE Visualization Conference. IEEE Press (1991) 284-291
13. Lamming, M. et al: Satchel: providing access to any document, any time, anywhere. ACM Transactions on Computer-Human Interaction, 7(3), ACM Press (2000) 322-352
14. Lamping, J. and Rao, R.: Laying out and visualizing large trees using a hyperbolic space. Proceedings of the 7th annual ACM symposium on User interface software and technology. ACM Press, Marina del Rey, California, United States (1994) 13-14
15. Malone, T.: How do People Organize their Desks? Implications for the Design of Office Information Systems, ACM Transactions on Office Information Systems, 1(1), ACM Press (1983) 99-112
16. Myers, B, Hudson, S and Pausch, R.: Past, present, and future of user interface software tools. ACM Transactions on Computer-Human Interaction, 7(1), ACM Press (2000) 453-469.
17. Nardi, B. and Barreau, D.: "Finding and Reminding" Revisited: Appropriate Metaphors for File Organization at the Desktop, ACM SIGCHI Bulletin, 29(1). ACM Press (1997)
18. Nielsen, J.: Supporting Multiple-Location Users, Jakob Nielsen's Alertbox, May 26, 2002. http://www.useit.com/alertbox/20020526.html
19. Robertson, G. et al: Data Mountain: using spatial memory for document management. Proceedings of the 11th annual ACM symposium on User interface software and technology. AM Press (1998) 153-162
20. Robertson, G., Mackinlay J. and Card, S.: Cone Trees: animated 3D visualizations of hierarchical information. Human factors in computing systems conference proceedings on Reaching through technology. ACM Press, New Orleans, Louisiana, United States (1991) 189-194
21. Rodden, K.: How do People Organize Their Photographs? Proceedings of the BCS IRSG 21st Annual Colloquium on Information Retrieval Research (1999).
22. Shneiderman, B. and Wattenberg, M.: Ordered Treemap Layouts. Proceedings IEEE Symposium on Information Visualization 2001. IEEE Press, Los Alamitos, California, United States (2001)