Characterising Electronic Document Use, Reuse, Coverage and Multi-Document Interaction

Jason Alexander and Andy Cockburn
Department of Computer Science and Software Engineering
University of Canterbury
New Zealand
{jason, andy}@cosc.canterbury.ac.nz

ABSTRACT

What documents do you use? How much of your document are others likely to read? How much time do you spend using documents? Desktop electronic document manipulation is one of the most common activities performed by computer users, yet there remains little empirical research into how documents are used in common document navigation systems.

This paper presents a 14 participant, 120 day study that logged user actions in Microsoft Word and Adobe Reader, with the aim of characterising document use. The study found that Microsoft Word documents are likely to be significantly shorter but have longer periods of interaction compared to Adobe Reader documents. Word documents averaged 6 pages in length and Reader documents 38 pages. Documents that were ten pages or less made up 80% of those that were opened.

Approximately half of the documents viewed were reopenings of ones previously used, however history mechanisms were poorly utilised. Document coverage in Microsoft Word was approximated by a normal distribution, while Reader document coverage decreased in a linear fashion, the further one moved toward the end. The time spent with multiple documents open decreased exponentially as the number of documents open increased. We briefly discuss the implications these findings have for the design of document navigation systems.

Categories and Subject Descriptors

H.5 [**Information Interfaces and Presentation**]: User Interfaces—*Interaction Styles*; D.2 [**Software Engineering**]: Human Factors in Software Design—*User interfaces*

Keywords

Human-Computer Interaction (HCI), electronic documents, document use, document navigation, client-side logging.

1. INTRODUCTION

Creating, editing, manipulating and reading electronic documents on the computer desktop is one of the most common activities performed by most, if not all computer users, from Information Technology specialists to casual home users. Yet there remains little empirical research into how we use these documents in common document navigation systems. Questions such as those posed in the abstract remain unanswered, yet they can provide valuable insights for designers of new navigation interaction techniques.

The goal of this research is to empirically characterise the use and manipulation of electronic documents in two common document preparation and navigation systems: Microsoft Word and Adobe Reader. It is intended to form the base for further research into these areas; it does not attempt to present more efficient mechanisms for document navigation.

This work forms part of a Ph.D. thesis into understanding and improving navigation within electronic documents. The thesis aims to empirically and contextually characterise document navigation, giving a solid grounding for the construction and evaluation of new navigation interaction techniques. Some aspects of this paper are described in more detail in [1].

The next section describes related work on document manipulation and techniques for logging user actions, followed by a brief description of our logging tool, AppMonitor. The method and results from a longitudinal study investigating patterns of document use, reuse, coverage and multi-document interaction are then presented.

2. RELATED WORK

This section examines two core areas of related-work that are important to this research: document manipulation and techniques for logging user activities.

2.1 Document Manipulation

There is a vast literature dealing with documents, ranging from document writing and reading (for example [10] and [19]) to document management systems (for example [21]). This section details the three of most relevant to this work: document interface styles, document switching and document revisitation.

2.1.1 Document Interface Styles

Our study involves both a Single-Document Interface (SDI, Microsoft Word) and a Multiple-Document Interface (MDI, Adobe Reader). SDIs display a single document in a single window, and are, anecdotally, easier to use, especially for less experienced users [17, 26]. They are recommended for use by the *Microsoft Windows*

User Experience Guidelines FAQs [17] and the *Mac OS X Technology Overview* [4]. Multiple-Document Interfaces use a single "primary" window and contain a set of child windows or documents *within* the primary window. Child windows share the menu- and tool-bars of the parent window [16].

2.1.2 Document Switching

Switching between documents is a subset of the task of window switching, for which much research has been performed. Several studies have investigated patterns in window switching and many have suggested alternative task-switching tools (for example [22]).

Researchers have studied window manipulation, both manually and empirically through automated data collection. Gaylin [7] videotaped workers during their everyday activities and drew results from manual analysis of the tapes. He observed that "cycling through windows" (i.e. switching to an active window as opposed to creating, deleting or resizing windows) was the most frequently used command, accounting for 63% of those observed.

More recent studies, such as Oliver et al. [20], implemented a system monitor to capture the Windows event stream to allow mechanical analysis of window manipulation. They wished to group related windows into tasks, by examining window titles and temporal closeness, to begin work on an automated task management system. Hutchings et al. [11] used their "VibeLog" application to log windowing events, allowing them to compare the difference between single and multiple monitor users for window management operations. They found that the average length of time a window was active was 20.9 *seconds* (significantly shorter than one might expect). They also analysed factors such as window visibility and empty space.

While our study does not provide a full insight into task switching, we can shed some empirical evidence on the simultaneous use of multiple documents with the applications under study.

2.1.3 Document Revisitation

Document revisitation has most widely been studied in the context of the Internet. Numerous researchers [5, 6, 18, 23] have empirically characterised the revisitation patterns of web users. The most recent of these studies ([18], 2007) found that 43.7% of web page requests were revisitations of pages previously viewed. The usefulness of history mechanisms to aid revisitation has also been studied. Obendorf et al. [18] found the 'back' button was used for 31% of all page revisits, while the combination of bookmarks, the homepage button, the history list and typed URLs were only responsible for 13.2% of page revisits. The web, however, is quite a different context to that of interest to this paper—desktop electronic document manipulation.

2.2 Logging Strategies

There exist several strategies for observing and recording the actions of users inside document navigation systems. This section considers four of these strategies: direct human observation, screen recorders, macro-based recorders and client-side logging techniques.

2.2.1 Direct Human Observation

Human observation of user interaction provides the most contextually rich understanding of user actions. Experimenters may silently observe users, or they may ask participants to use think-aloud protocols through contextual inquiry [9], to understand their thought patterns and reasons for making choices. These techniques are often backed by video taping the sessions to allow later analysis or confirmation of notes made. These techniques are time-consuming (more than one-experimenter hour per observation hour—two or three-times if video analysis is involved) and can only collect high-level empirical results. Human observation is most likely to succumb to the Hawthorne Effect [15], where participants change their actions because they are being observed.

2.2.2 Screen Recorders

Screen recorders, such as TechSmith's Camtasia Studio (`www.techsmith.com`), record both the screen contents as well as mouse and keyboard actions made by the user. As per direct human observation, the output of screen-recorders is very time-consuming to analyse, as effectively it is a video of the user's session. Screen-recorder software is also very resource-intensive, capturing and writing to disk the same stream of pixels being sent to the screen. This means study participants will be unwilling to have a monitor such as this installed for a long period of time when they will notice a performance impact.

2.2.3 Macro-Based Recorders

Macro-based recorders log user mouse and keyboard actions, and semantic information regarding the application without the need to record screenshots, as per screen recorders. These applications are designed to let mundane, manual tasks be easily repeated. Macro instructions are recorded in a human-readable form to allow modifications or manual editing, and hence could be employed to record the actions of a user over a long period of time. Examples of commercial macro recorders include: Iolo Technologies' "Macro Magic" (`www.iolo.com/mm`), Tethys Solutions' "Workspace Macro" (`www.tethyssolutions.com`), CprinGold Software's "Smack" (`www.cpringold.com`) and Jitbit Software's "Macro Recorder" (`www.jitbit.com`). Researchers have used Microsoft Word's macro system to record high level commands issued to the menus and buttons within the application [14].

2.2.4 Client-Side Logging Techniques

Client-side application loggers monitor interaction at either the application or Operating System level. They produce structured log files that allow mechanical processing of large amounts of data.

The simplest form of client-side loggers are mouse and keyboard loggers. Several of these loggers exist for research purposes: Datalogger [27] for Windows 3.1 and DOS, InputLogger [24] for the Apple Macintosh, and RUI [25] for Windows and Mac OS X.

Macro-based recorders have an advantage over simple mouse and keyboard loggers in that they can record semantic information as well as mouse and keyboard interaction. However, they do not log the state of the application of interest. For instance, information such as scrollbar positions, document zoom, view or length are not recorded. Custom client-side logging applications are able to record all of this information.

Targeted client-side logging applications are able to record application state, as well as semantic information regarding user interaction. The most common example of client side logging applications is for studying interaction with web-browsers. These studies have often required the user to abandon their preferred software in favour of customised logging-equipped versions of open source browsers (for example [23]) or roll-your-own solutions (for example [13]). While these options allow full logging of the application, users have to learn a new interface. Custom interfaces often offer a subset of interaction techniques with which users are familiar, possibly leading to inaccuracies in the data recorded.

3. APPMONITOR

AppMonitor is a Microsoft Windows based program that logs user actions in Microsoft Word and Adobe Reader. This system is fully described in [2], however, due to its integral part in this work, we briefly describe it here.

AppMonitor allows logging of all user actions in unmodified Microsoft Windows based programs. Once installed, AppMonitor requires no input from study participants. It automatically starts when a user logs into the computer and captures events whenever an application of interest is opened. Events are stored in a structured file locally. Log files are then uploaded to a web server when a local buffer size limit is reached.

AppMonitor can record low level events, such as mouse movement and key presses, as well high level interaction such as menu selections and button presses. It also tracks changes in document state such as the current view and document length, scrollbar position(s) and zoom level.

The set of events logged is configurable by the researcher. Each event is entered into a log file with a date, timestamp, and window handle information, allowing individual actions to be linked to specific documents and applications, even when multiple documents are open simultaneously.

The system was developed and tested on Microsoft Windows XP (but also functions correctly on Microsoft Windows Vista), using Microsoft Word 2003 and Adobe Reader 7. Only minor changes are expected to be needed to the system to allow it to correctly log actions in any version of these applications, or indeed any Windows based program.

4. LONGITUDINAL STUDY

The longitudinal study monitored the navigation actions of 14 volunteer participants over a period of 120 days. The participants were Computer Science staff and postgraduate students, two of whom were female. All participants rated themselves as “advanced” or “expert” users. They were asked to install AppMonitor on their computer and continue with their everyday work as normal. Participants all used Windows XP, Microsoft Word 2003 and Adobe Reader 7.

In this study, we disabled AppMonitor’s full key-logging ability to allay any privacy concerns of the participants (as theoretically document reconstruction would have been possible). We also disabled logging of mouse movement, to reduce both the size of the log files and the processing demands of AppMonitor.

5. RESULTS AND DISCUSSION

We divide the results of this study into four areas of interest: document use, reuse, coverage and multiple document interaction. Each of the results are presented with a discussion.

These results maintain the distinction between the data obtained from Microsoft Word and the data from Adobe Reader. This allows us to compare whether the type of application (document editor/reader vs document reader) and the tools available within it affects the type of documents, and how they are manipulated in each of these interfaces. Note, we are not attempting to compare the two interfaces to find the “best” or most efficient.

5.1 Document Usage Sessions

To ease analysis, we introduce the concept of a Document Usage Session or DUS. A DUS is a period of time where the user is interacting with their document, or more accurately, a period of time where AppMonitor has recorded at least one event from that document. A session begins when an event for that particular document is registered. It then continues until a period of five minutes has passed without AppMonitor collecting any events. The document is then classified as *idle*, meaning it is assumed that the user is not using or interacting with the document in any manner. A DUS is terminated when the document is closed. A document always contains at least one DUS.

It should be noted, as stated earlier, that full keyboard and mouse logging was disabled for this study. Potentially, a user may spend a large amount of time reading a fixed part of the document (without moving its position) and after five minutes this interaction time will no longer be counted. A user could also spend a long period of time typing, without using any navigation keys (arrows, enter, tab, home, end etc). This situation is unlikely, as typing is an activity that requires corrections and re-writing, and will inevitably require the system to automatically scroll to keep the cursor position onscreen. We believe any “missed” interaction time is balanced by the time where users move immediately away from the application while we continue to think they are interacting.

DUS’s are useful when applying time analysis to the data recorded by AppMonitor. We noticed that users regularly leave their documents open for hours, overnight, or even days without interacting with them. DUS discards this idle time when a user is not interacting with a document.

5.2 Document Use

Participants opened a total of 2342 Microsoft Word documents and 1706 Adobe Reader documents during the 120 day period. On average, at least one of the applications was used for 49.6 days (s.d. 20.0 days) of the 120 day period, and on these days of use, there was an average of 3.72 (s.d. 1.62) Word documents and 4.12 (s.d. 5.57) Reader documents manipulated, per person. Application usage is sporadic and “bursty”—opening one document indicates a high likelihood of opening other documents on the same day.

AppMonitor records the length of the document (in pages) as it is opened[1]. A distribution summary of document lengths is shown in Figure 1. The average length of Microsoft Word documents was 6.3 pages (s.d. 4.47 pages), and Adobe Reader documents was 38.2 pages (s.d. 35.7 pages). The longest document recorded in Microsoft Word was 160 pages, and in Adobe Reader was 1743 pages; the smallest were 0 and 1 respectively.

Adobe Reader documents were significantly longer than those opened in Word. This difference is put down to the popularity of the PDF file format used by Adobe Reader for distributing over the Internet many forms of documents, including books and manuals. Documents opened in Word are more likely to be created, edited or reviewed by the user, and are less likely to be large manuals or books, given our group of participants.

[1]Some lengths were unavailable, so are omitted from this analysis.

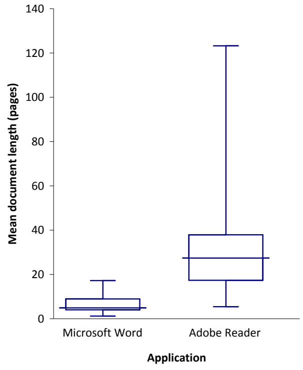

Figure 1: Distribution of mean document lengths per person, by application

Figure 2 shows the distribution of document lengths across all participants. We have truncated this graph at 20 pages, less than the average document length of Reader. Many large documents (in the ranges of hundreds of pages) were opened in Reader, with lengths quite disperse. Adding these values to the graph was impracticable.

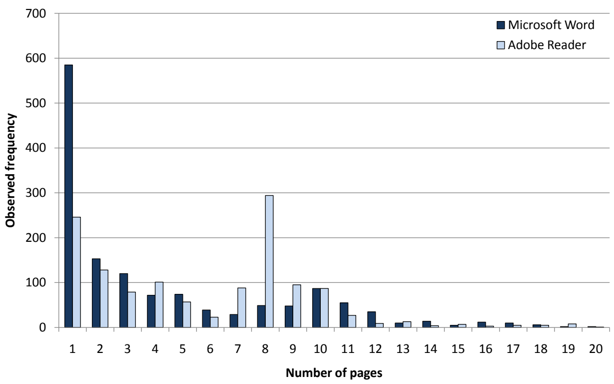

Figure 2: Document distribution, by length

Documents ten pages or less accounted for 80% of those manipulated. The figure shows a spike in the document lengths in the 7–10 page region. This is due to our volunteer users having a high likelihood of manipulating academic conference papers, often with lengths in this range.

The analysis presented here shows that document navigation systems need to be able to cope with a wide range of document lengths. The design of navigation mechanisms must be flexible and capable of handling both short documents (1 page) and long documents (1743 pages). Much research effort (for example [3, 8, 12]) has focused on creating document navigation systems that manipulate large cumbersome documents, however, designers should be aware that these larger documents are only used a small percentage of the time.

A reasonable hypothesis would be that long documents would have long periods of interaction to allow for searching and reading. Figure 3 shows the correlation between the number of pages in the document and average interaction time recorded with documents of this length.

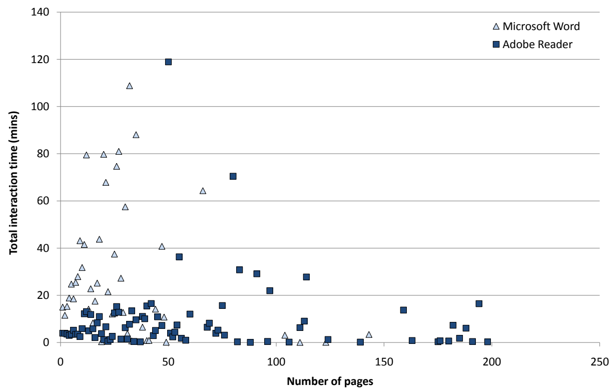

Figure 3: Average length of document interaction time

Word documents are, in general, shorter, but have a longer period of interaction, while Reader documents are generally longer, but have less interaction.

This is contrary to the expected result, but is explained by the difference in applications. Word users are more likely to be creating a document (which is inherently a more time-consuming task) than reading or searching a document, for which the Adobe Reader application is designed. From these results, we can hypothesise that larger documents can be assumed to only be used for reference, or partially read, and not open for long periods of time for complete "cover-to-cover" reading. This hypothesis is investigated further in section 5.4: Document Coverage.

Our participants had a habit of leaving unused documents open for long periods of time. Figure 3 details the interaction time with documents, not the length of time they were open. Word documents were, on average, were idle for 88% and Reader documents were idle for 85% of the time they were open. Designers of new task switching interfaces should be aware that often documents (and this may also generalise to applications) are left open for long periods of time without being required for use.

5.3 Document Reuse

The AppMonitor software records the title of documents as they are opened[2]. We use this information to investigate document uniqueness and patterns of document reuse. While it is possible that participants may use the same name for a different document (storing it in a different directory), this is considered to have a low probability, unlikely to significantly affect these results.

Recall that a DUS is a session of continuous (without a five minute gap) activity using a document. The average number of DUSs per person, per document was 2.2 (s.d. 0.7) for Microsoft Word and 1.3 (s.d. 0.3) for Adobe Reader. The average period of these interaction sessions was 9.6 minutes (s.d. 5.6 mins) for Microsoft Word and 3.0 minutes (s.d. 1.8 mins) for Adobe Reader. 78% of Word and 91% of Reader documents have a single session of interaction.

The data we gathered indicates that participants spend a small number of relatively short sessions interacting with their documents. Closer inspection of the data reveals that the extremes for these applications are quite different. The largest number of DUSs recorded

[2]Some titles were unavailable, so are omitted from this analysis.

for Word was 185 and for Reader was 12. However, with 97% and 99% of the DUS counts being less than 10 for Word and Reader respectively, we see that tasks are generally completed in these applications with only a small number of long (greater than five minute) interruptions.

5.3.1 *Document History Systems*

Document history systems within Microsoft Word and Adobe Reader are based purely on the last few documents (four by default in Word and five in Reader) that the user has opened[3]. This is in contrast to web-browsers such as Internet Explorer (`www.microsoft.com/ie`) and Mozilla Firefox (`www.mozilla.com/firefox`) where history systems maintain listings of all of the web-pages viewed over the last period of time (often two weeks), regardless of the number of pages visited. We now investigate how our observations have implications for history mechanisms in document navigation systems.

There are two factors that influence the usefulness of this type of history system: the length of time between reuse of a document and the number of other documents opened in between reuse of the original document. Figure 4 shows the average length of time between the closing and re-opening of the same document, according to the number of times the document was recorded to have been opened. Most documents followed a regular pattern of having a shorter period between closing and opening the more frequently it was used. The anomalies in this graph are interesting—the points where a user has opened the same document many times in the space of a day or two. These points may be explained by inadequacies in the document navigation software. One participant commented that parts of his logs may look strange, as he had to re-open Reader every time he recompiled his document using LaTeX; the ability to reload the document instead of closing and reopening may ease this user's navigation requirements.

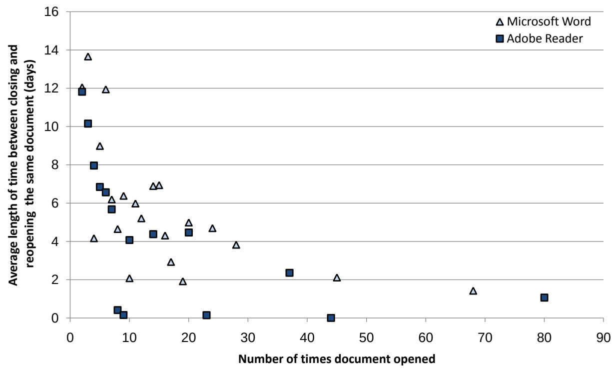

Figure 4: Average length of time between closing and re-opening of the same document

The number of other documents used between the re-opening of a document affects whether the document in question is still in the *recent documents* list. Figure 5 reports the number of documents opened between reopenings of a particular document. Word documents almost always have less than 20 documents opened between reopenings, however, most counts are still above ten, rendering the recent documents history mechanism unhelpful for quickly reopening documents. In Adobe Reader, this history mechanism is even less likely to be helpful, with majority having over 20 other documents used between reopenings.

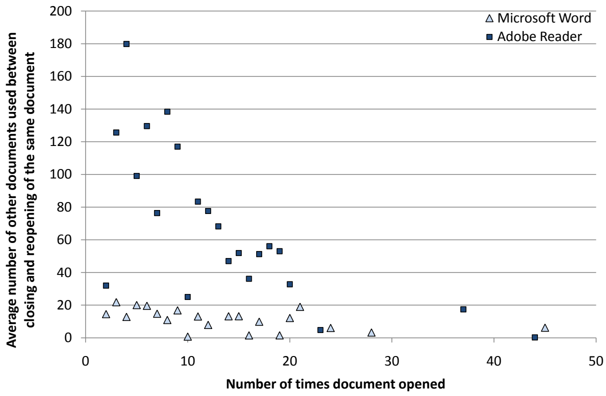

Figure 5: Average number of documents used between closing and re-openings of the same document

The number of other documents between re-openings shown in Figure 5 and the lengths of time shown in Figure 4 suggest that providing history mechanisms such as those in web browsers would be more useful to users than the current systems. A two week history list (14 days) would include all of those documents depicted in Figure 4.

5.4 Document Coverage

AppMonitor allows the monitoring of scrollbar movements as a document is navigated using tools and techniques such as the scrollbar and the mousewheel. This section utilises the scrolling data to examine the amount of time spent in different areas of a document and whether document coverage varies with the length of the document.

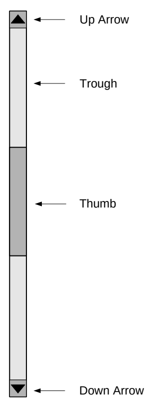

Figure 6: A standard vertical scrollbar

The scrollbar thumb (see Figure 6) not only provides feedback on the current position in the document, its size is also directly proportional to amount of the document currently visible on-screen. A small thumb indicates a small percentage of the document is currently visible, while a large thumb indicates a large percentage is visible. AppMonitor records the size of the scrollbar trough, the thumb position and the thumb size. We use this data to calculate the currently visible "window" of the document, along with the time spent in that window to form a measure of document coverage.

Figure 7 shows a summary of document coverage for all documents in the two applications of interest. For clarity, we have divided the document into 10% 'chucks'—each representing a section of the document. As with other measures reported in this paper, we use a DUS timeout of five minutes.

[3]Adobe Acrobat Standard and Adobe Acrobat Professional (the commercial, feature rich versions of Adobe Reader) do contain a further history mechanism that allows the user to view the documents from: today, yesterday, the last seven, fourteen or thirty days or the last twelve months

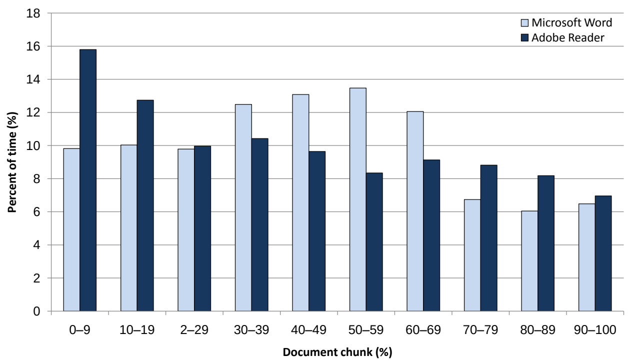

Figure 7: Document coverage by application

We observe two quite different patterns for the two applications. Adobe Reader documents, in general, have less time spent in a particular section the further towards the end one progresses. In contrast Microsoft Word documents follow almost a standard distribution curve, with the most time spent in the middle third of the document. These differences likely arise from the editing ability of Microsoft Word, with the middle of the document taking longer, or requiring more time than the extremities. This result highlights the importance of executive summaries or abstracts written at the start of an electronic document. They are more likely to be read than other areas of the document, possibly prompting further reading, printing or closing.

The summary of document coverage shown in Figure 7 summarised all documents, regardless of length. We now investigate how the length of a document affects what portion of the document is viewed. Figure 8 displays the correlation between document length[4] and the percentage of the document that is viewed for Microsoft Word. Data points displayed are the averages across users, document lengths that only occur once or twice are omitted (as their averaged values are bias).

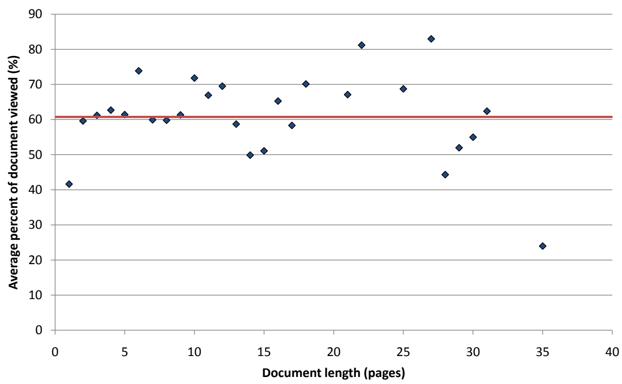

Figure 8: Percentage of document viewed in Microsoft Word

The average percentage of the document viewed for Microsoft Word is 61% (s.d. 12%). We observe that document length (in the 0–35 page range) has very little effect on the coverage, with most documents falling between 50% and 70% viewed.

[4]As noted earlier, some documents are omitted as their length was unavailable.

For clarity, the correlation for Adobe Reader is shown separately in Figure 9. For documents under 20 pages, we see the same pattern as observed for Word—most documents have between 50% and 70% coverage. For documents with larger page counts we observe a general trend of less of the document being viewed, the longer the document becomes. As we have no valid data points for longer documents for Word we are unable to tell how generalisable this pattern is.

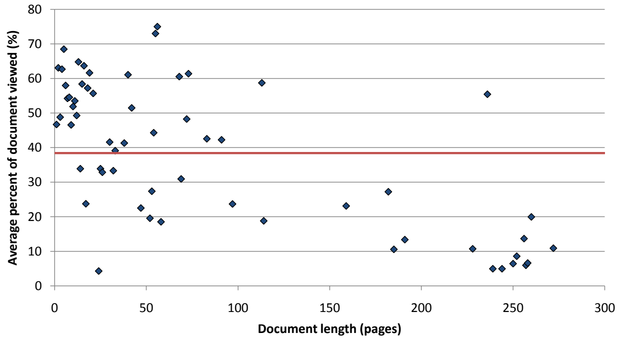

Figure 9: Percentage of document viewed in Adobe Reader

In line with the hypothesis of section 5.2, it is reasonable to deduce that users who open large documents are unlikely to be opening them to read from start to end, as they do not have large coverage values. Instead they are likely to be reference manuals or documents that one wishes to search using built-in tools.

5.5 Multi-Document Interaction

It is common for users to not only interact with multiple programs in multiple windows, it is also common to have multiple documents open in the same application. The distinction between Multiple-Document Interfaces (MDIs) and Single-Document Interfaces (SDIs) was made in section 2.1.1—recall that Adobe Reader is an MDI application and Microsoft Word an SDI application.

One measure of multi-document interaction is the amount of time users spend with more than one document open. Figure 10 displays the time distribution with various numbers of documents open. Note, that the percentage of time spent with a single document open is also included, for completeness.

Users typically spent 44% of their time with only a single Word or Reader document open. Thus, over half of users' time is spent with more than one document open. The time spent with multiple documents open decreases exponentially as the document count increases. Figure 11 isolates this behaviour into documents in the same application. Word users spent a larger amount of time with one or two documents open, while Reader users were more likely than their counterparts to have a greater number of documents open simultaneously. The type of interface (SDI vs. MDI) does not appear to have a significant impact on how multiple documents are used.

6. LIMITATIONS

The empirical analyses presented in this paper are based on a 14 person sample of Computer Science staff and postgraduate students. Some participants were observed to be regular application users, while others only occasionally used the applications under

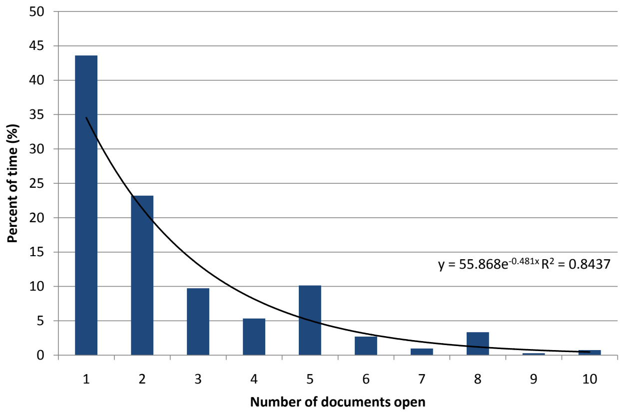

Figure 10: Distribution of time with multiple documents open

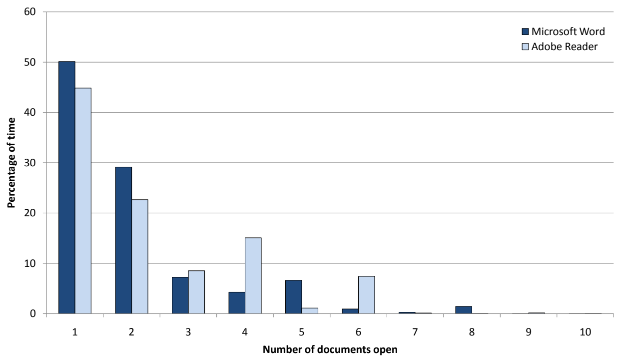

Figure 11: Distribution of time with multiple documents open in the same application

study. One reason for this was that for a few users, Windows was only installed on their secondary machine.

There is no reason to believe that our "expert" Computer Science user patterns do not generalise to any "expert" user. A larger, more broad field study, with participants who do not classify themselves as experts would be needed to determine whether navigation patterns change as proficiency increases.

7. CONCLUSIONS AND FUTURE WORK

We have presented the results of document use, reuse, coverage and multi-document interaction from a 14 participant, 120 day study. Several important properties concerning the use of electronic documents on the desktop have been presented.

Microsoft Word documents are likely to be significantly shorter than Adobe Reader documents, averaging 6.3 pages and 38.2 pages respectively. The largest document recorded in Word was 160 pages, and in Reader was 1743 pages. Documents ten pages of less made up 80% of those that were opened.

Word documents were found to be shorter, but have longer periods of interaction than the longer counterparts opened in Reader. These documents were, on average, idle for 88% of the time in Word and 85% of the time in Reader.

Approximately half of documents viewed are reopenings of ones previously used, however, between individuals we observed a wide variation of reuse. There was minimal to no use of recent document history mechanisms to access these 'popular' documents, giving rise to the possibility of applying web-browser style history lists to aid frequent revisitation.

Each of the applications were observed to have different document coverage patterns. Word documents followed an approximate normal distribution, with the middle third most viewed. Reader documents followed a more linear model, with a smaller amount of time spent in a region of the document, the closer it was to the end. The length of Word documents did not significantly effect the amount of the document viewed (on average 61%), but Reader documents had a larger length range, and it was observed that longer documents had reduced coverage rates.

Over half of the time, users have more than one document open. The time spent with multiple documents open decreases exponentially as the number of documents open increases.

This paper has presented a starting point for the empirical characterisation of document use. We expect other researchers (as well as ourselves) to use this data to begin producing theories and models for interaction with documents. The AppMonitor tool collects a vast amount of data regarding user interactions, of which we have only reported a small portion. Further analysis of this data, as well as that of post-study interviews is about to take place.

This study has utilised Microsoft Word and Adobe Reader, but there are a vast number of other document navigation tools available to computer users. Specialist areas, such as Computer Science also harbour a number of specialist tools, such as code editors. Extending this study into the domain of other software tools would be valuable.

8. ACKNOWLEDGMENTS

The authors would like to thank all of the study participants, especially those who assisted in the beta-testing stage of the software.

9. REFERENCES

[1] J. Alexander and A. Cockburn. An Empirical Characterisation of Electronic Document Navigation. In *GI '08: Proceedings of Graphics Interface*, In Press, 2008.

[2] J. Alexander, A. Cockburn, and R. Lobb. AppMonitor: A Tool for Recording User Actions in Unmodified Windows Applications. *Behavior Research Methods, In Press*, 2008.

[3] C. Appert and J.-D. Fekete. OrthoZoom Scroller: 1D Multi-Scale Navigation. In *CHI '06: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 21–30, New York, NY, USA, 2006. ACM Press.

[4] Apple Inc. Mac OS X Technology Overview. Online: `http://developer.apple.com/documecntation/MacOSX/Conceptual/OSX_Technology_Overview/PortingTips/chapter_9_section_3.html`, Accessed: December 2007.

[5] L. D. Catledge and J. E. Pitkow. Characterizing Browsing Strategies in the World-Wide Web. In *Proceedings of the Third International World-Wide Web Conference on Technology, Tools and Applications*, pages 1065–1073, New York, NY, USA, 1995. Elsevier North-Holland, Inc.

[6] A. Cockburn and B. McKenzie. What Do Web Users Do? An

Empirical Analysis of Web Use. *International Journal of Human-Computer Studies*, 54(6):903–922, 2001.

[7] K. B. Gaylin. How are Windows Used? Some Notes on Creating an Empirically-Based Windowing Benchmark Task. In *CHI '86: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 96–100, New York, NY, USA, 1986. ACM Press.

[8] Y. Guiard, M. Beaudouin-Lafon, Y. Du, C. Appert, J.-D. Fekete, and O. Chapuis. Shakespeare's Complete Works as a Benchmark for Evaluating Multiscale Document Navigation Techniques. In *BELIV '06: Proceedings of the 2006 AVI Workshop on Beyond Time and Errors*, pages 1–6, New York, NY, USA, 2006. ACM Press.

[9] K. Holtzblatt and S. Jones. *Participatory Design: Principles and Practice*, chapter Contextual Inquiry: A Participatory Technique for System Design, pages 180–193. Lawerence Earlbaum, 1993.

[10] W. J. Hunter and J. Begoray. A Framework for the Activities Involved in the Writing Process. *The Writing Notebook*, 8(1), 1990.

[11] D. R. Hutchings, G. Smith, B. Meyers, M. Czerwinski, and G. Robertson. Display Space Usage and Window Management Operation Comparisons Between Single Monitor and Multiple Monitor Users. In *AVI '04: Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 32–39, New York, NY, USA, 2004. ACM Press.

[12] T. Igarashi and K. Hinckley. Speed-Dependent Automatic Zooming for Browsing Large Documents. In *UIST '00: Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, pages 139–148, New York, NY, USA, 2000. ACM Press.

[13] M. Kellar, K. Hawkey, K. M. Inkpen, and C. Watters. Challenges of Capturing Natural Web-based User Behaviours. *International Journal of Human-Computer Interaction, In Press*, 2007.

[14] F. Linton, D. Joy, H.-P. Schaefer, and A. Charron. OWL: A Recommender System for Organization-Wide Learning. *Educational Technology & Society*, 3(1), 2000.

[15] E. Mayo. *The Human Problems of an Industrial Civilization*. Cambridge, MA: Harvard University Press, 1933.

[16] Microsoft Corporation. Design Specifications and Guidelines - Window Management. Multiple Document Interface. Online: `http://msdn.microsoft.com/library/default.asp?url=/library/books/winguide/ch10a.htm`, Accessed: December 2007.

[17] Microsoft Corporation. Microsoft Windows User Experience Frequently Asked Questions. Online: `http://msdn2.microsoft.com/en-us/library/ms997581.aspx`, Accessed: December 2007.

[18] H. Obendorf, H. Weinreich, E. Herder, and M. Mayer. Web Page Revisitation Revisited: Implications of a Long-Term Click-Stream Study of Browser Usage. In *CHI '07: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 597–606, New York, NY, USA, 2007. ACM Press.

[19] K. O'Hara. Towards a Typology of Reading Goals. Technical Report EPC-1996-107, Rank Xerox Research Centre, 1996.

[20] N. Oliver, G. Smith, C. Thakkar, and A. C. Surendran. SWISH: Semantic Analysis of Window Titles and Switching History. In *IUI '06: Proceedings of the 11th International Conference on Intelligent User Interfaces*, pages 194–201, New York, NY, USA, 2006. ACM Press.

[21] G. Robertson, M. Czerwinski, K. Larson, D. C. Robbins, D. Thiel, and M. van Dantzich. Data Mountain: Using Spatial Memory for Document Management. In *UIST '98: Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, pages 153–162, New York, NY, USA, 1998. ACM Press.

[22] C. Tashman. WindowScape: A Task Oriented Window Manager. In *UIST '06: Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, pages 77–80, New York, NY, USA, 2006. ACM Press.

[23] L. Tauscher and S. Greenberg. How People Revisit Web Pages: Empirical Findings and Implications for the Design of History Systems. *International Journal of Human-Computer Studies*, 47(1):97–137, 1997.

[24] S. Trewin. InputLogger: General-Purpose Logging of Keyboard and Mouse Events on an Apple Macintosh. *Behavior Research Methods, Instruments and Computers*, 30(2):327–331, 1998.

[25] F. E. R. Urmila Kukreja, William E. Stevenson. RUI: Recording User Input from Interfaces under Windows and Mac OS X. *Behavior Research Methods*, 38(4):656–659, November 2006.

[26] S. Vanophem and K. Vanstappen. Making Oracle Behave. In *CHI '06: CHI '06 Extended Abstracts on Human Factors in Computing Systems*, pages 153–158, New York, NY, USA, 2006. ACM Press.

[27] S. Westerman, S. Hambly, C. Alder, C. Wyatt-Millington, N. Shryane, C. Crawshaw, and G. Hockey. Investigating the Human-Computer Interface Using the Datalogger. *Behavior Research Methods, Instruments and Computers*, 28(4):603–606, 1996.