Death to the Office of the Nineties: An HCI Perspective on Some Problems in Modern Office Information Systems

Andy Cockburn, Tim Bell, Bruce McKenzie

Department of Computer Science
University of Canterbury
Christchurch, New Zealand
{andy,tim,bruce}@cosc.canterbury.ac.nz

There is powerful evidence that office information systems and the information technology revolution are failing to deliver the expected productivity gains. Increased use of user-centred design and the associated methods and techniques of human-computer interaction (HCI) promise to improve on this poor performance. This paper demonstrates how empirical studies can help in understanding the use and efficiency issues of systems within the modern office. Three case studies examine usability problems that arise from different sources. Firstly, a study of the usability of differing email formats shows that the actions of one user can impact on the efficiency of many other users. Secondly, a study of the use of web page bookmarks shows that, even with effective tools, users can create information management problems for themselves. Thirdly, a study of next-generation 3D systems for file and document management shows that the temptations of technology can confuse both designers and users in striving for more efficient tools. Throughout the case studies we illuminate some of the conflicting issues and factors that make HCI a deeply challenging and rewarding area of research.

Keywords: Office Information Systems, HCI, empirical evaluation, email overload, World Wide Web bookmarks, 3D interaction.

1. INTRODUCTION

In his book "The Trouble With Computers" Thomas Landauer (1995) argues that information technology has largely failed to provide productivity gains. He states that the main causes of this failure are inappropriate uses of technology, poor understanding of the users' tasks, and inadequate user interface design. He proposes that a user-centred focus on the design, development, and deployment of computer systems can turn the tables on this poor performance.

Research into human-computer interaction (HCI) has the goal of improving the usability of computers. This is a wide ranging objective that demands the understanding of the human factors of perception, cognition and motor coordination, together with the societal impact of new technology. Furthermore, HCI researchers must develop and test new ways of supporting users, as well as finding improved methodologies for designing successful computing systems.

Manuscript received: ???
Associate Editor: Judy Hammond

There are a wide range of HCI tools and techniques that can be used to help improve the learnability, efficiency and subjective satisfaction of computer users. The objective of this entry in the 'Special Edition on HCI' is to illustrate the use of one of these techniques – empirical evaluations – in understanding the use of everyday computer tools. We present three case studies. As well as demonstrating empirical evaluations, we use the studies to illuminate the diverse range of conflicting factors that complicate designing systems for human use. It is these factors that make HCI such a challenging, rich and fascinating area of research.

2. CASE STUDY 1: EMAIL FORMATS

There are four main problems associated with working with email. First, there is 'email overload'. It is not uncommon for people to spend a substantial part of their working day dealing with overloaded email inboxes. Since Denning (1982) observed the arrival of 'junk email', many filtering and prioritisation systems have attempted to ease email overload, for example the Information Lens (Malone, Grant, Turbak, Brobst and Cohen, 1987). The second main problem is email 'flaming' where conversants become engaged in an escalating war of (textual) words. This problem is largely due to the absence of 'back-channels' such as facial expression and tone of voice in email communications. 'Emoticons' such as ':-)' have become common to reinforce the writer's intent in email, and mature email users carefully chose their words to reduce the likelihood of flaming. Viruses, the third major problem of email use, are a relatively new email problem. Sophisticated email systems support powerful capabilities such as automatically opening email attachments within particular applications. Although these capabilities can enhance the efficiency of working with trusted collaborators, they also create a security weakness that malicious computer users can exploit.

The fourth email problem, and the focus of this case study, is that of working with different email formats.

Motivating Anecdotes

Like many organisations, our institution recently embraced email as a medium for information exchange. Many staff, who would have previously spent much of their time in print and mailrooms, can now distribute their documents widely at the click of a button. But does this apparent increase in efficiency save time?

Consider the following quotes extracted from email messages recently received by the authors:

- PLEASE NOTE THAT THE ATTACHMENT HAS BEEN SCANNED AND SAVED AS A POWERPOINT FILE.
  Message sent to approximately 40 Heads of Department at our university.
- Two more trys!! A comment was made to me that some staff have no access to word... So here are two alternatives.
  Message sent to fourteen Academic Staff. Both attachments were unformatted, making the tabulated information they contained unusable.

Our supply of examples of inappropriate document distribution with email is extremely long. The two examples above demonstrate ingenuity (!) and frustrated sledgehammer approaches by the message creators. What the examples do not show is the loss of productivity caused for the recipients who must deal with the information received.

2.2 Efficiency of Different Email Formats

Our study investigates the relative costs of working with different email formats. The ultimate goal of the study is to develop guidelines that help office workers make appropriate decisions about document distribution.

Although there are many possible email formats that could be included in the study, we focus on five: raw-text email, html encoded email, 'url to html' email (meaning a raw-text email message that includes a reference to a 'normal' HTML web-page), 'url to pdf' email (meaning a raw-text email message that includes a reference to a web-page that is a PDF[1] document, and 'word attachment' email (meaning a message that contains a Microsoft Word attachment).

We break the 'costs' of working with different email formats into three categories, each described below: work for the sender, data transmission and storage costs, and work for the recipient(s).

2.2.1 Work for the sender

Busy office workers are often concerned about clearing items from their “To Do’’ list as quickly as possible. Naturally, the quickest and most convenient way of doing so is often the most attractive. Unfortunately, what is quickest for the sender need not be the most efficient for the organisation, once the costs of dealing with inappropriate formats are multiplied across the recipients.

To provide an estimate of the costs for the sender of various formats, we timed how long it took one of the authors to prepare a simple one-sentence message “How about lunch at 3pm on Friday?” in the five formats identified in the previous section. This specific analysis addresses only one scenario of email usage, and it ignores issues such as distribution of pre-existing documents. Initial guidelines and recommendations for the use of different email formats in specific usage scenarios are discussed in Bell (1999).

The email system used in the study was Netscape Mail, and it was running in an iconified window when the clock started for each format. Netscape Mail's default text editor was used: plain text rather than HTML text.

The raw-text email message took 17 seconds to send.

The HTML email message took 36 seconds to send. A substantial fraction of the message composition time was spent changing the preferences for the message creation tool to include HTML markup capabilities. Had the preferences for HTML editing already been set, the total time would have been similar to the raw-text time.

The URL to HTML format took 75 seconds to send. The actions necessary to complete the task included authoring the web-page (with Netscape Composer), saving the file on a web-server, setting access permissions on the file, then composing an email message to refer to the URL for the page. The technical proficiency required to carry out this task, and the opportunity for error, are substantially higher than the previous two formats.

The URL to PDF format took 90 seconds to send. Creating the PDF document involved using 'distill' to create a PDF document from a Postscript file which was generated from a LATEX letter file. The PDF document was copied to a web-server and the access permissions were changed. Finally, an email message referring to the URL was composed and sent. The technical demands of this process are relatively high, but some applications greatly ease the process. It is also worth noting that the final PDF document can be a highly polished publication, including organisational logos, quality fonts, and so on.

The MS Word attachment took 26 seconds to send. This included the start-up time for Word, typing time, menu-selection time, and the time to complete two dialogue boxes for identifying the message recipient. The technical demands of using this technique are low and similar to those for the raw-text and HTML formats.

[1] Adobe Portable Document Format

2.2.2 Data costs

Although our primary interest is in the human costs of working with different email formats, there are also very different media costs associated with the different formats. Mail messages are normally stored on both the sender's and the receivers' computer disks, and network bandwidth is absorbed in transmitting the email and web files.

Three of the five formats (raw-text, HTML and word attachment) are primarily 'push' methods for information sharing, where the sender delivers the full information to the email inbox of all of the recipients. With 'push' methods, the recipients have no choice about whether to commit disk space to the incoming email.

The 'URL to HTML' and 'URL to PDF' formats are primarily 'pull' methods, where each recipient receives a notification of the existence of a web-page and can subsequently choose whether or not to visit the page. The 'pull' methods are substantially cheaper on disk-space and network bandwidth because only those recipients who are interested in the information go to the expense of downloading it. Also, data downloaded through web-browsers is normally automatically deleted from the cache after a relatively short time (usually less than two weeks), but most email applications are cautious about deleting email (in Netscape Mail, 'Deleting' a message moves it to a trash folder, and only a subsequent 'Empty Trash' frees the disk space). At the time of writing, one of the authors was shocked to find that his email Trash folder was consuming over 26 megabytes of diskspace.

One advantage of 'push' methods is that the recipient has a personal copy of the information. When using 'pull' technologies there is a risk that the recipient will be unable to retrieve the information because the file has been moved from its original location or because a web-server is unavailable.

Table 1 compares the file sizes associated with each format. The 'Email' row shows the amount of data transferred to each recipient in the email message. The 'Web-server/Cache' row shows the amount of data transferred (and stored on disk) when the recipient of an email message follows the link to a specified URL in the message. The main data-saving in using the 'pull' formats ('URL to HTML' and 'URL to PDF') arise because only those recipients who suspect that they will be interested in the document incur the data cost of accessing the web-server. Note that each copy of the Word Attachment format requires approximately fifty times more data to be transmitted than the raw-text format.

	Raw-text	**HTML**	**URL to HTML**	**URL to PDF**	**Word Attachment**
Email	527	1367	527	527	27517
Web-server	0	0	309	2213	0
Total	**527**	**1367**	**836**	**2740**	**27517**

Table 1: Data costs (in bytes) for a one-sentence plain text email message in differing formats

2.2.3 Work for the recipients

The common one-to-many relationship between email senders and recipients indicates that it is the efficiency of the email recipients in working with different email formats that most heavily influences overall efficiency. This contrasts with the desire of the individual senders to deliver their messages in the quickest and most convenient method *for themselves.*

We measured the times taken to send replies to one-sentence email questions that were prepared in the five different formats identified above. Each of the questions asked a simple 'Yes/No' question such as ‘Is your hair red?’. Fourteen professionals took part in the study which took place in their offices using their normal computing environment. Seven of the subjects were Unix users, four used one of the Microsoft Windows operating systems, and three were Mac users. The seven Microsoft Windows/Mac users all used Microsoft Word as their normal document preparation system, but none of the Unix users did. For the purpose of data analysis the Windows and Mac users were treated as a single group: comparing Mac/Windows users who primarily work with direct-manipulation interface with Unix users who primarily work with a command-line interface. A one hundred and sixty second time limit was placed on all of the tasks.

We analysed the task times using a two-way mixed analysis of variance (ANOVA) for factors ‘format’ and ‘operating system’. The format factor (within subjects) had five levels: raw-text, html, url to html, url to pdf, and word attachment. The operating system factor (between subjects) had two levels: Unix users, and Windows/Mac users.

Figure 1 shows the mean task completion times for each of the five formats across the two operating system types. Error bars show one standard error above and below the mean. The means for the two operating system types were not significantly different ($F(1,12)=2.66$, $p=0.128$). Unsurprisingly, the means for the five formats were significantly different ($F(4,48)= 14.74$, $p<.01$). There was also a significant interaction between the two factors ($F(4,48)=6.82$, $p<.01$). The cause of the interaction is almost entirely due to the ‘Word’ format task: Figure 1 clearly reveals that the Windows/Mac users were able to quickly solve the task using this format, while those using Unix were not. Removing the ‘Word’ task from the ANOVA analysis shows no significant interaction between ‘operating system type’ and ‘email format’ ($F(3,36)=0.16$, $p=0.923$). In other words, each of the formats *except* the Word Attachment format resulted in similar task completion times for both operating systems.

Our observations of the subjects completing these tasks give further insights into the variety of problems that users have in dealing with different email formats. All of the formats except for ‘raw-text’ and ‘html’ required the subjects to make a context-switch to another window or application outside the email reader (a web-browser, Adobe Acrobat Reader, Microsoft Word, or some other application). Sometimes the users would open many new windows while attempting to read and reply to the message. We observed several cases of users suffering window management problems in returning to their email system. In one case, the user’s click to regain input focus on the email window had the side-effect of selecting a different email message in their inbox. They failed to

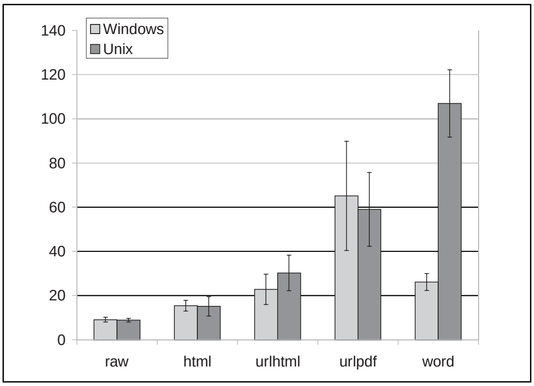


Figure 1: Mean task completion times for th efive email formats across thetwo operating system types. Error bars show one standard error above and below the mean.

notice the side-effect and issued the 'reply' command and typed their response to the wrong user. Only intervention by the observer stopped them adding to some innocent third party's email overload and confusion.

Two things are clear from the results: firstly, raw-text is an extremely efficient format for short email messages; secondly, exposing users to 'advanced' formats can result in serious problems for some users. Two subjects (one Windows user and one Mac user) were unable to complete the 'url to pdf' task within the 160 second time-limit, and two different subjects (both Unix users) were unable to complete the 'word attachment' task. A third-party would have been necessary for these users to reply to messages in these formats.

3. CASE STUDY 2: WEB-PAGE BOOKMARKS

The previous study investigated external sources of user problems: message senders causing difficulties for the message recipients. This study shows that there are also many ways that users can create problems for themselves. By investigating the use of Netscape Navigator's bookmarking tool we show that as computer tools become more popular and useful they can introduce new problems for users to contend with.

In a recent four-month study, we found a web-page revisitation rate of approximately 80% (McKenzie and Cockburn, 2001): of every five pages displayed in the user's browser, four have been previously visited. Given this high revisitation rate, it is clearly important that web-browsers provide efficient techniques for returning to pages. 'Bookmarks' or 'Favorites' are one of the main tools for returning to web-pages. These tools allow the user to create a short-cut access method for the page currently displayed in the browser. Abrams, Baecker and Chignell (1998) found that 84% of the subjects in their study had more than eleven bookmarks.

How do users use bookmarking tools, and what problems do they have with them? This case study provides further information on the use of bookmarks.

3.1 Problems with Bookmarks

This section summarises the anecdotal experiences and empirical findings of a four month study of seventeen IT professionals' use of web-bookmarks with Netscape Navigator. The full experimental details of the empirical study are described in McKenzie and Cockburn (2001).

3.1.1 Planning versus doing

Two of the major anecdotal problems of working with bookmarks arise from the explicit action that is necessary to add a page to the bookmark collection. First, users must remember to carry out the necessary bookmarking actions. Although this sounds trivial, anecdotal reports indicate that users frequently forget to do so (or remember too late). It seems that web-browsing is often a deeply engaging activity (searching for a particular paper, for instance), but bookmarking requires users to switch from a 'doing' mode of action to a 'planning' mode. It appears that deep engagement in browsing reduces the likelihood of the mode switch, and the consequent failure to bookmark the pages.

The second major anecdotal problem is that users often do not know that a particular page will become relevant to them. Several users mentioned that they have had to search for previously visited pages that, at the time, they did not realise they would want to return to.

3.1.2. Titles and page identification

Netscape Navigator's bookmarking tool and Microsoft Internet Explorer's Favorite tool both use the 'Title' tag extracted from the page's HTML to provide the default page identification label. In earlier studies we found that approximately 5% of pages have no Title tag (McKenzie and Cockburn, 2001), and that errors in Title tags are common (Cockburn and Greenberg, 2000).

3.1.3 Page redirections and broken links

Web-sites are easily modified and many sites are frequently modified through the addition, deletion, and re-organisation of material. This is a problem for bookmarking tools that record a static URL for the bookmarked page. We found that 25% of our subjects' bookmarks did not retrieve valid pages two months after the study (McKenzie and Cockburn, 2001).

3.1.4 The overloaded bookmarker

Our empirical analysis of bookmark use indicated that users suffer information management problems in dealing with large bookmark collections.

One of our subjects had 587 bookmarks and was adding to his collection at three times the rate of deletion. Across the seventeen subjects, the mean number of bookmarks held by the subjects was 184, and all had addition rates outweighing deletion.

There are three ways that the empirical results indicate that there are problems in managing bookmarks. First, the normal user interface mechanism for selecting bookmarks is a drop-down cascading menu. Each item in the top-level of the user's bookmark structure is rendered as a separate entry in the top-level of this menu. The mean number of entries in the top-level structure was 42, with a maximum of 130. Clearly, one hundred and thirty items in a single level of a menu is an unwieldy user-interface component!

The second indication of problems in managing bookmarks is illuminated by investigating the number of items in the user's top-level bookmark structure over time. Figure 2 (extracted from McKenzie and Cockburn, 2001) shows the number of items in the top-level bookmark structure plotted against time for three users over the four month study. The clear steps in each line of the graph indicate that the user has spent time re-organising their bookmark structure. Further analysis showed that when these steps occurred, few bookmarks were deleted. Rather, the existing bookmarks were placed into folders at lower-levels in the bookmark structure.

The final indication of bookmarking problems is the number of unknown duplicate bookmarks found in the users' collections. Across all users, approximately 5% of bookmarks were duplicate entries, and many of these were unknown to the users.

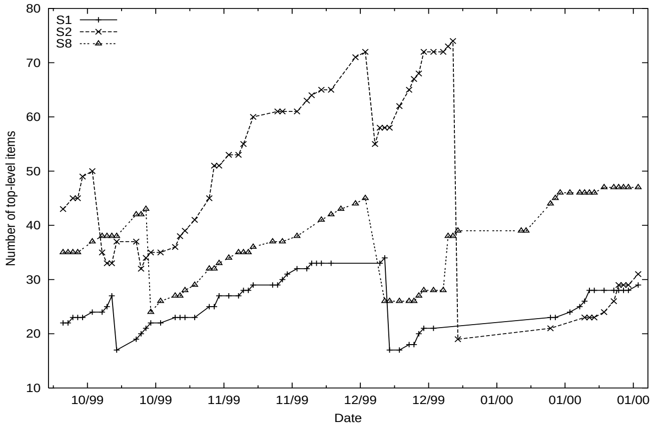


Figure 2: Number of items in the top-level bookmark structure against time

3.2 Possible solutions?

Although our study only analysed the use of Netscape Navigator's bookmarking tool, it seems reasonable to suspect that similar results would be found in a study of Microsoft Internet Explorer's 'Favorites' tool. Regardless of the generality of results, the study provides interesting insights into some of the problems that interface designers face.

The original design motivation for bookmarking tools addressing was probably something like: "how can we support fast and effective access to the small set of pages that users will want to return to frequently?" The designers succeeded in easing this user problem, but the existence of effective bookmarking tools has changed the way that people try to keep track of pages, and now managing the tool is becoming a problem.

This lead-on effect between one problem-solution creating a new problem is a good example of 'homeostasis'. Homeostasis (Thimbleby, 1990) describes the tendency of humans to maintain equilibrium: if a user interface makes tasks easier, it will be used to do more difficult tasks. In the case of bookmarks, the interface allows users to effectively manage more web-pages, but ultimately it is used to manage so many pages that management is no longer effective.

The question for the designers of the next-generation web-browsing interfaces is "what can be done to remove the problems of managing large numbers of bookmarks?"

Some of the bookmarking problems identified above are already being addressed through the iterative development of browsers. For instance, in helping users manage their bookmark collections, Netscape Navigator supports an 'Update Bookmarks' option that automatically checks all of the bookmarked web-pages and flags those that do not address an accessible web-page. This would help users identify broken bookmarks (approximately 25% in our study) as candidates for deletion.

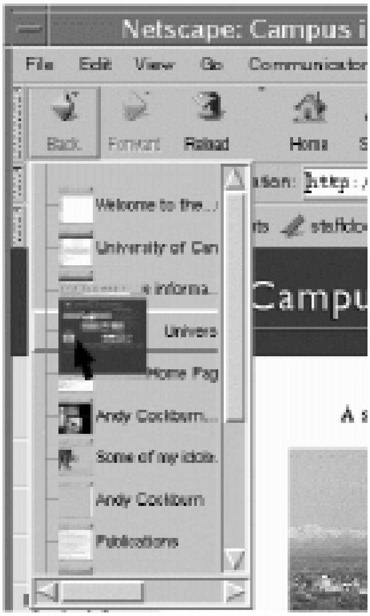


Figure 3: A recent version of WebView that is integrated with Netscape's Back/Forward buttons. Moving the mouse over any thumbnail causes it to zoom to a larger size (see top thumbnail).

Another possible enhancement to bookmark interfaces would be to allow more page identification cues than simply the page 'Title' tag. As mentioned in Section 3.1.2, many pages are missing title tags (5% in our study) and many more have incorrect or misleading titles. Thumbnail images of pages provide another cue to page identification that would reinforce the page title and URL. Further discussion on page-identification cues is given in Cockburn and Greenberg (2000), and the webView system (Cockburn, Greenberg, McKenzie, JasonSmith and Kaasten, 1999) demonstrates a wide variety of visual and textual page identification cues.

Looking beyond bookmarks, web-browsers currently support a wide variety of different schemes for revisiting pages. These include Back/Forward buttons, "Go" lists, history tools, bookmarking/favorite tools, and past-URL selectors. Integrating these might reduce the learning overhead of these diverse tools while allowing users the flexibility of choosing the most effective revisitation tool for their current task. Figure 3 shows a recent version of our webView system which integrates zooming thumbnails of past pages with the Back/Forward buttons. WebView's history of past pages can be searched through dynamic queries, and pages can be bookmarked by simply clicking on the appropriate thumbnail.

4. CASE STUDY 3: MOVEMENT TO THREE DIMENSIONAL INTERFACES

The third case study investigates some of the risks associated with the enticing properties of new technology. We focus on current research systems that use three-dimensional (3D) displays for common office tasks such as file, document, and bookmark management.

This case study of 3D interfaces reveals a tension between empirical and subjective results in usability testing. We have found that subjects in our studies have expressed higher values for subjective satisfaction for 3D interfaces even when the interfaces have made them less efficient. We suggest that users, like designers, are susceptible to the 'sirens of technology', and that HCI researchers should be cautious about subjective measures that result from short usability evaluations of 'neat' and 'cool' systems. Furthermore, we encourage HCI researchers to isolate variables in their evaluations so that the HCI community can determine the impact of each component of innovative new systems.

4.1 Three dimensional systems for document and file management

Recent research into three-dimensional systems is extending the realism of the visual presentation of user interfaces. Three dimensional visualisation has been an essential component of 3D modelling systems such as CAD/CAM, but only recently have researchers begun examining 3D visualisations for the everyday office tasks of file and document management. Example systems include Cone Trees for browsing hierarchical file structures (Robertson, Mackinlay and Card, 1991), WebBook and WebForager for browsing collections of web pages (Card, Robertson and York, 1996), the Data Mountain for organising and accessing thumbnail representations of web pages (Czerwinski, Dumais, Robertson, Dziadosz Tiernan and van Dantzich, 1999), and the Task Gallery which is a 3D window manager (Robertson, van Dantzich, Czerwinski, Hinckley, Thiel, Robbins, Risden and Gorokhovsky, 2000).

Some of these systems have been evaluated in comparison to competing systems. For example, the Data Mountain has been shown to be more efficient for locating web-page bookmarks than Microsoft Internet Explorer's bookmarking tool (Czerwinski *et al*, 1999). However, there has been no serious attempt to investigate the impact of 3D (rather than 2D) in these interfaces. For instance, in the evaluation of the Data Mountain, it was unclear whether the efficiency gain was due to the use of 3D or to some other factor. Are designers (and users) succumbing to the tempting sirens of technology, or does 3D really enhance the user's experience?

4.2 Evaluations that isolate dimensionality

We have conducted two experiments that attempt to isolate dimensionality as an experimental factor. The 3D systems that we have studied are based on Cone Trees and the Data Mountain. Full details of these two studies are presented in Cockburn and McKenzie (2000) and Cockburn and McKenzie (2001).

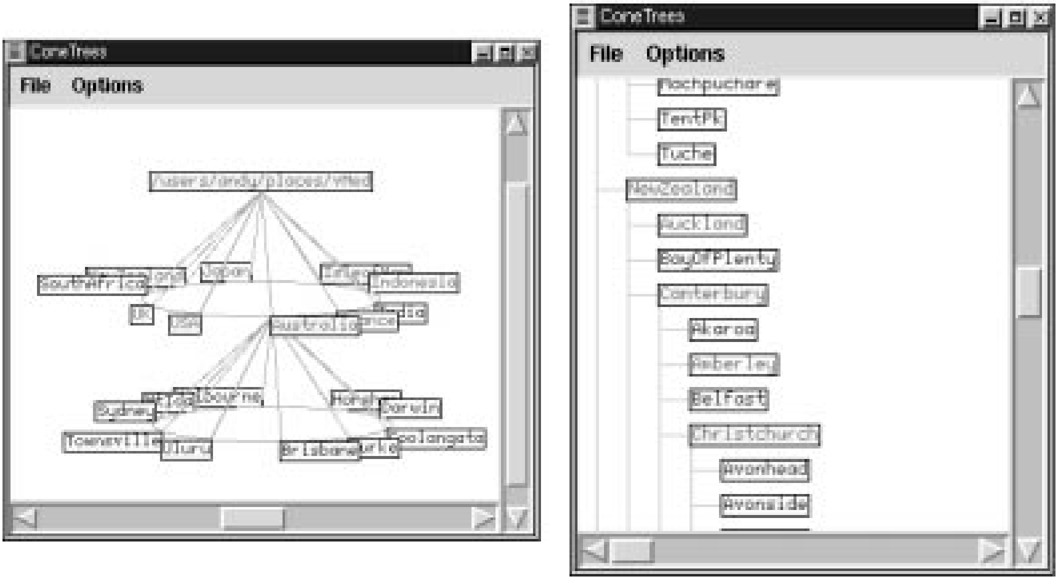
4.2.1 Cone Trees versus 'normal' trees

Cone Trees display hierarchical file structures by arranging the contents of each directory around an inverted cone. In Figure 4(a), a Cone Tree shows the contents of two directories, and in Figure 4(b) a 'normal' tree browser shows the same data structure. Each cone can be rotated to bring particular files to the front of the cone.

The two interfaces used in our experiment (shown in Figure 4) have identical mouse bindings and display schemes. Directories are displayed with red text and yellow background, and files are identified by blue text on a white background. Clicking on a directory toggles the expanded display of its contents.

The subjects' tasks consisted of locating particular files within a hierarchical data structure of countries, regions, cities and places: for example, "Find Sumner in Christchurch in Canterbury in New Zealand''. We measured their task completion times and subjective assessment of how effective the interface was for the task, measured on a 5 point Likert scale.

As well as investigating the efficiency of the 3D versus 2D interfaces, we investigated two additional factors. The first was the relative efficiency of the two interfaces as the density of the data structure increased. The three levels of density (sparse, medium and dense) were measured with branching factors of six, ten and twenty files and directories per directory. The second additional factor was the depth of the search path. The two levels of depth (shallow and deep) consisted of two and four levels of search: for example, a shallow task might involve finding Canterbury in New Zealand, while a deep task might involve finding Kings Cross in Sydney in New South Wales in Australia.



(a) The Cone Tree interface *(b) The Normal Tree interface*

Figure 4: The experimental interfaces

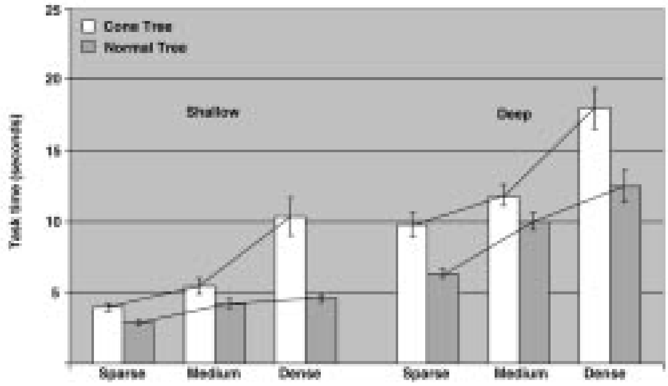


Figure 5: Mean task completion times for the Cone Tree and Normal Tree interfaces across differing densities and depths. Error bars show one standard error above and below the mean.

The experimental design was a three-way ANOVA with repeated measures for the factors: interface type (Cone Trees versus Normal Trees), density (sparse, medium and dense), and depth (shallow versus deep). Twelve subjects participated in the study.

The task performance results are summarised in Figure 5. The main effects for each of the three factors were significant. When using Cone Trees the subjects took significantly longer to complete tasks ($F(1,11)=71.98$, $p<.01$) with mean task completion times of 9.88 and 6.74 seconds for the Cone and Normal Tree interfaces. Unsurprisingly, depth ($F(1,11)=133.93$, $p<.01$) and density ($F(2,22)=64.12$, $p<.01$) were also significant: in other words, the subjects took longer to solve deep tasks than shallow ones, and they took longer to solve dense tasks than sparse ones.

There was a significant interaction between the factors of density and interface type ($F(2,22)=12.8$, $p<.01$). The cause of this interaction can be seen in Figure 5. As the density increases, there is a fairly linear increase in task time for the Normal Tree, but for the Cone Tree there is a larger increase in task completion times between the medium and dense conditions.

The subjects' responses to questions about the effectiveness of the two interfaces were significantly different, with the Normal Tree interface ranking higher than the Cone Tree interface. The reason for the difference was the problems that users had reading overlapping file and directory labels in the Cone interface. In the dense conditions, overlapping labels made the tasks substantially more difficult to complete. Despite these problems, many of the subjects stated that they liked the Cone Tree interface and that it gave them a better sense of the data-space structure than the Normal Tree interface.

4.2.2 2D versus 3D Data Mountains

The Cone Tree study suffers a major weakness. We wanted to isolate dimensionality as a factor for study, but the nature of interaction with Cone Trees makes this extremely difficult. When using Normal Trees, scroll-bars are used to navigate around the data-space. Scroll bars are directly under the user's control, and they can choose how quickly to move through the data. The Cone Tree equivalent to scrolling is cone rotation. If the part of the data space that the user wishes to interact with is ”at the back’’ of the cone, the user can rotate the cone to bring that region to the front. Cone rotation is a dynamically animated effect, and the speed of rotation is outside the user's control. There is, therefore, a necessary latency in waiting for the target item to come to rest at the front of the cone.

(a) The 2D Interface (b) The 3D Interface

Figure 6: The two versions of the Data Mountain

For this reason, the Cone Tree experiment does not completely isolate interface dimensionality as a factor for study. In aiming to overcome this limitation we carried out another experiment comparing 2D and 3D versions of the Data Mountain (Czerwinski *et al*, 1999).

The 3D Data Mountain allows users to locate thumbnail images of web pages on an inclined plane or ‘mountain’. Thumbnails diminish in size as they are moved further up the mountain, and they are dynamically redisplayed so that ‘nearer’ thumbnails occlude overlapping ‘further’ ones.

We constructed a simple version of the 3D Data Mountain (Figure 6(b)) and a 2D version that is almost identical *except* for the 3D ‘distance’ metaphor (Figure 6(a)). In the 2D interface all thumbnails are an identical size, but the user can control the layering order of thumbnails: clicking on any thumbnail raises it above all the others.

The experimental design was a mixed two-way analysis of variance for the factors interface type (2D versus 3D, between subjects) and density (33, 66 and 99 thumbnail images in the display, within subjects). Each subject used either the 2D or 3D interface, with 14 subjects in each condition. The subjects’ tasks involved positioning 33 thumbnails of well-known web-pages in the display, and then locating – as quickly as possible – ten randomly selected pages from those 33. All tasks were timed. They then repeated the process, adding two more sets of 33 well-known pages to the display. After each set of ten page retrievals the subjects were asked to rate the effectiveness of the interface on a 5-point Likert scale. Full experimental details are presented in Cockburn and McKenzie (2001).

The mean task times are summarised in Figure 7. Although themean task completion times were lower for the 2D interface in each of the densityconditions, there was no significant difference between interface type($F(1,26)=0.289$, $p=.6$). As expected, there was a significant difference between densities ($F(2,52)=8.752$,$p<.01$): the subjects took longer to find thumbnails as the displays became more cluttered in the denser tasks. There was no significant interaction between interface type and density ($F(2,52)=.072$, $p=.93$): the subjects’ performance deteriorated at a similar rate as density increases in both interfaces.

Although there was no significant difference in task completion times between the two interfaces, there was a significant difference between the subjects’ assessment of the effectiveness of the two interfaces. The mean Likert response values to the question “Overall, the interface is effective’’ were significantly higher (better) for the 3D interface than for the 2D one (Mann-

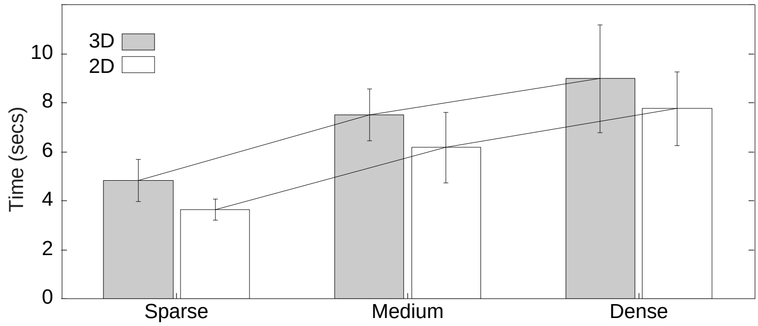


Figure 7: Mean task completion times for the 2D and 3D Data Mountain interfaces. Error bars show one standard error above and below the mean.

Whitney U Test, U=51, N_1=N_2=14, p<.05). There were many comments about the 3D interface being ‘cool’ and ‘a good way to organise bookmarks’, but there were few equivalent statements about the 2D interface.

4.3 Lessons Learned

In the Cone Tree study, we found that Cone Trees are less efficient than their 2D counterparts for tasks that involve searching for named files and directories. In the Data Mountain evaluation we found no significant difference in task performance between the 2D and 3D interfaces, but significantly higher subjective assessment for the effectiveness of the 3D interface.

So what generalisable lessons for office information systems can be taken from these two studies? We offer two cautions to usability professionals.

The risks of over-generalising results. It is often tempting to over-generalise the results of experiments. Cone Trees, for instance, may be valuable tools for different types of tasks than those tested in our study. Alternatively, a different implementation of Cone Trees *may* actually improve task performance in tasks similar to those we used. For this reason, it is critical that papers reporting on experiments provide sufficient detail for others to repeat the experiment and confirm or refute the results.

One of our major motivations in carrying out the Data Mountain evaluation was to investigate the extent to which the third dimension added value to the support it provided. We were impressed by the original results of the Data Mountain evaluation (Robertson, Czerwinski, Larson, Robbins, Thiel and van Dantzich, 1998) but were unclear about the extent to which its successful comparison to Microsoft Internet Explorer was due to its 3D properties and to its support for the subjects’ spatial memory about the location of the thumbnails. Following our evaluation, we would argue that spatial memory accounts for much of the performance increase, and that the value of the third dimension remains unclear.

The risks of believing subjects’ assessment of enticing technologies. Subjects in the 3D Data Mountain condition were clearly enthusiastic about the interface they were using, with many describing it with phrases such as ‘cool’. It is interesting that the 3D subjects rated interface ‘efficiency’ significantly higher than the 2D subjects, despite the absence of performance indicators to support this assessment. It seems likely that the relative ‘coolness’ of the 3D interface swayed the subjects’ assessment of its efficiency.

Naturally, subjective evaluation of interfaces is extremely important in establishing the degree to which users like their systems. However, it is not clear how enduring their assessment of a 'cool' but inefficient system will be, nor is it clear whether users will choose to pay for these systems and to adopt them into their everyday work practices. The short duration of typical usability studies may offer a heavily distorted impression of subjective satisfaction.

5. SUMMARY AND CONCLUSIONS

This paper presented three case studies that examined problems in the use of office information systems. Each of the case studies examined the efficiency impact of factors that are primarily caused by other users, by the users themselves, and by the system designers. The studies demonstrate evaluation techniques used by researchers in human-computer interaction to understand system use and to identify possible areas for iterative improvement of user interfaces. They also demonstrate the diverse factors that influence user interface design and use.

As well as demonstrating HCI methodology, each case study illuminates a problem in working with modern office information systems. By analysing the problems of working with email, we provide concrete data on the usability impact of different email formats. This analysis could be used to begin the development of guidelines for message and document distribution. Our analysis of World Wide Web bookmark collections provides insights into usage patterns. This data could be used to inform the design of next generation bookmarking tools. Finally, our study of three dimensional interaction tools for file and document management indicates that researchers should carefully consider the efficiency of their tools before encouraging developers to adopt sophisticated user interfaces.

REFERENCES

ABRAMS, D., BAECKER, R. and CHIGNELL, M. (1998): Information archiving with bookmarks: Personal web space construction and organization. *Proceedings of CHI'98 Conference on Human Factors in Computing Systems*, Los Angeles, April 18, 23: 41–48.

BELL, T. (1999): Electronic document distribution: Why I can't read Microsoft Word attachments. http://www.cosc.canterbury.ac.nz/~tim/epublish/

CARD, S., ROBERTSON, G. and YORK, W. (1996): The WebBook and the WebForager: An information workspace for the World Wide-Web}. *Proceedings of CHI'96 Conference on Human Factors in Computing Systems*, Vancouver, April 13, 18: 111–117.

COCKBURN, A. and GREENBERG, S. (2000): Issues of page representation and organisation in web browser's revisitation tools. *Australian Journal of Information Systems,* 7(2): 120–127.

COCKBURN, A., GREENBERG, S., McKENZIE, B., JASONSMITH, M. and KAASTEN, S. (1999): Webview: A graphical aid for revisiting web pages. *Proceedings of the 1999 Computer Human Interaction Specialist Interest Group of the Ergonomics Society of Australia* (OzCHI'99). November 28–30 Wagga Wagga: 15–22.

COCKBURN, A. and McKENZIE, B. (2000): An evaluation of Cone Trees. People and Computers XV (Proceedings of the 2000 British Computer Society Conference on Human-Computer Interaction.) University of Sunderland, 4–8 September, Springer-Verlag: 425–436.

COCKBURN, A. and McKENZIE, B. (2001): 3D or not 3D? Evaluating the effect of the third dimension in a document management system. *Proceedings of CHI'2001 Conference on Human Factors in Computing Systems,* Seattle, Washington, March 31–April 6: 434–441.

CZERWINSKI, M., DUMAIS, S., ROBERTSON, G., DZIADOSZ, S., TIERNAN, S. and VAN DANTZICH, M. (1999): Visualizing impicit queries for information management and retrieval. *Proceedings of CHI'99 Conference on Human Factors in Computing Systems,* Pittsburgh, May 15–20: 560–567.

DENNING, P. (1982): Electronic junk. *Communications of the ACM* 25(3): 163–165.

LANDAUER, T. (1995): The trouble with computers: Usefulness, usability, and productivity. MIT Press.

MALONE, T., GRANT, K., TURBAK, F., BROBST, S. and COHEN, M. (1987): Intelligent information-sharing systems. *Communications of the ACM* 30(5): 390–402.

McKENZIE, B. and COCKBURN, A. (2001): An empirical analysis of web page revisitation. *Proceedings of the 34th Hawaiian International Conference on System Sciences,* HICSS34, Maui, Hawaiii, (CD ROM)}, IEEE Computer Society Press.

ROBERTSON, G., CZERWINSKI, M., LARSON, K., ROBBINS, D., THIEL, D. and VAN DANTZICH, M. (1998): Data mountain: Using spatial memory for document management. *Proceedings of the 1998 ACM Conference on User Interface Software and Technology,* November 1–4. San Francisco, California, ACM Press: 153–162.

ROBERTSON, G., MACKINLAY, J. and CARD, S. (1991): Cone trees: animated 3D visualizations of hierarchical information. *Proceedings of CHI'91 Conference on Human Factors in Computing Systems,* New Orleans, May}: 189–194.

ROBERTSON, G., VAN DANTZICH, M., CZERWINSKI, M., HINCKLEY, K., THIEL, D., ROBBINS, D., RISDEN, K. and GOROKHOVSKY, V. (2000): The task gallery: A 3d window manager. Proceedings of CHI'2000 *Conference on Human Factors in Computing Systems,* The Hague, The Netherlands, April 1–6: 494–501.

THIMBLEBY, H. (1990): User interface design. ACM Press, Addison-Wesley.

BIOGRAPHICAL NOTES

Andy Cockburn is a Senior Lecturer in Computer Science at the University of Canterbury. He gained his PhD from the University of Stirling, Scotland in 1993. His research interests are in human-computer interaction and Computer Supported Cooperative Work.

Tim Bell is a Senior Lecturer and Head of the Department in Computer Science, at the University of Canterbury. He holds a Ph.D. in Computer Science from the University of Canterbury. His research interests include text compression, computers and education, and computers for music. He is the author of about fifty journal and conference papers, and three books, one in its second edition.

Bruce McKenzie received a B.Sc. (Hons) in 1975 and a Ph.D. in 1978, both from the University of Canterbury. He joined the Department of Computer Science of the same University in 1980 and was Head of Department for seven years from 1991. His research interests are mainly in the broad areas of compiler construction, formal language theory, compression and HCI.