It's Not That Important: Demoting Personal Information of Low Subjective Importance using *GrayArea*

Ofer Bergman†, Simon Tucker†, Ruth Beyth-Marom‡, Edward Cutrell •, Steve Whittaker†

†Sheffield University
211 Portobello St., Sheffield, S1 4DP, UK
{o.bergman, s.tucker, s.whittaker}
@sheffield.ac.uk

‡The Open University of Israel
108 Ravutski St., P.O. 808, Raanana
43107, Israel
ruthbm@openu.ac.il

•Microsoft Research
One Microsoft Way, Redmond WA
98052-6399, USA
cutrell@microsoft.com

ABSTRACT

Users find it hard to delete unimportant personal information which often results in cluttered workspaces. We present a full design cycle for *GrayArea*, a novel interface that allows users to *demote* unimportant files by dragging them to a gray area at the bottom of their file folders. Demotion is an intermediate option between keeping and deleting. It combines the advantages of deletion (unimportant files don't compete for attention) and keeping (files are retrieved in their folder context). We developed the *GrayArea* working prototype using thorough iterative design. We evaluated it by asking 96 participants to 'clean' two folders with, and without, *GrayArea*. Using *GrayArea* reduced folder clutter by 13%. Further, 81% of participants found it easier to demote than delete files, and most indicated they would use *GrayArea* if provided in their operating systems. The results provide strong evidence for the demotion principle suggested by the user-subjective approach.

Author Keywords

Personal information management, files, demotion, subjective importance, user-subjective.

ACM Classification Keywords

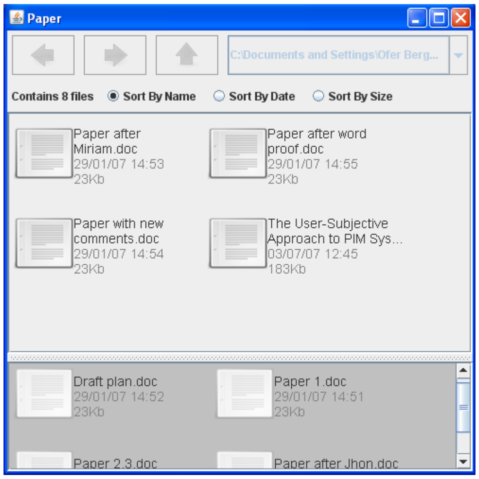
H.5.2 User Interfaces: Evaluation/methodology, User-centered design.

INTRODUCTION

People experience problems in deciding the value of personal information and we all spend large amounts of time making judgments about whether to keep or delete such information. Personal information items such as files, emails, bookmarks and contacts vary in their subjective importance. Some are highly important, while others are of low subjective importance. Some items of low subjective importance may be older items that were once highly important, but have since lost their relevancy. Yet others were kept for anticipated future retrieval which never occurred [1, 13, 20, 29].

Jones [20] claims that the decision whether "to keep or not to keep" information for future usage is prone to two types of costly mistakes: First information not kept is unavailable when it is needed later. On the other hand, if kept, irrelevant information items can create clutter. They compete for the user's attention, obscuring important information relevant to the current task. Indeed it is well known in the field of cognitive psychology that in visual search the number of irrelevant distracters increases the time taken for people to identify a target object [24, 27]. Thus keeping irrelevant information not only causes guilt about being disorganized [5] it also increases retrieval time. In addition there is a "deletion paradox": while unimportant information items distract attention and increase retrieval time for the target item, it takes time and attention to review items to decide whether to keep or delete them [7]. These are significant problems. Millions of users organize and retrieve their personal information several times a day, and these problems will be exacerbated as personal collections continue to grow in size.

Personal Information Management (PIM) studies repeatedly demonstrate that people experience problems in deciding



C:\Documents and Settings\Ofer Berg...

Contains 8 files ● Sort By Name ○ Sort By Date ○ Sort By Size

Paper after Miriam.doc
29/01/07 14:53
23Kb

Paper after word proof.doc
29/01/07 14:55
23Kb

Paper with new comments.doc
29/01/07 14:54
23Kb

The User-Subjective Approach to PIM Sys...
03/07/07 12:45
183Kb

Draft plan.doc
29/01/07 14:52
23Kb

Paper 1.doc
29/01/07 14:51
23Kb

Paper 2.3.doc

Paper after Jhon.doc

Figure 1: The *GrayArea* prototype. Demoted files appear at the bottom of the folder.

CHI 2009, April 4–9, 2009, Boston, MA, USA.
Copyright 2009 ACM 978-1-60558-246-7/08/04…$5.00.

what information to keep. Whittaker and Hirschberg [29] documented how users spent hours trying to rationalise their paper archives as part of an office move, and in the end were only able to delete 22% of their overall archives. In addition a significant percentage of what they deleted turned out never to have been accessed. Similarly studies of email processing reveal that users often keep messages of unclear value, even though this action makes it more difficult to locate valued messages [4, 28]. The same behaviors are observed with contact management, where again users retain information about people they are unsure they will contact again [30]. This tendency to avoid deletion is also found in studies of filing: participants avoid maintenance work, preferring to use additional storage media [2]; When presenting their file collections during interviews, participants repeatedly notice unimportant files [12, 18]; Finally, a study of personal photo archiving reveal that users find it difficult to delete pictures even when these pictures are near duplicates of pictures already in their archive [22].

Some of the reasons for avoiding deletion are rational, after all the user can always think of a situation when the information item may be needed [29]. However there are also psychological reasons why people avoid deletion, many of which can be attributed to the decision making process as described in *Prospect Theory* [21]: (a) People judge losses and gains relative to a subjective reference point. There is an inherent asymmetry between the two decision alternatives. The decision to keep was already made when a file was first created or accepted. From then on, "to keep" is the default reference point, while "delete" is a new possible risky alternative. (b) People prefer alternatives which avoid a sure loss (even when the chance of incurring the loss is very small), and keeping avoids loss of an information item. (c) Small objective probabilities are subjectively perceived higher, thus the very small probability that a deleted information item might be needed seems significant. (d) Losses loom larger than gains, thus the possible loss of an information item emotionally affects the decision-maker more than the gains of having fewer distracters or reduced retrieval time.

As a result, numerous PIM studies indicate that users' folders are often cluttered with unimportant information items [1, 12, 20, 22, 31]. As one of Boardman and Sasse's [12] participants commented: "*Stuff goes in but doesn't come back out – it just builds up*" (p. 585). This problem, as well as others (concerning personal information management) are shared by different users who perceive them as their own personal fault. However, these problems can be addressed by interface design [10, 19].

In this paper we present and evaluate *GrayArea* a novel experimental folder interface designed to address the keep/delete dilemma by implementing a third option called demotion. The initial application area for our prototype was file management, as users typically consider files to be the most valued part of their PIM collection [6, 11]. *GrayArea* allows users to demote files of low subjective importance by dragging them to a gray area at the bottom of the folder (see figure 1). This combines the advantage of deleting (unimportant files do not compete for attention) with the advantage of keeping (these files can be retrieved in the context of their original folder).

We describe a full iterative design cycle for *GrayArea* (as suggested by [25] and revised by [17]). We motivate the design in two ways. First we present the user-subjective approach which includes the principle of demotion, allowing users to reduce the visual salience or files of low subjective importance as an intermediate way between keeping and deleting [7]. Second we present empirical data from current system usage that indicate a need for such demoting interfaces [8]. The present study focuses on the three next steps in the design cycle: We generated three possible demoting designs, and collected early user feedback on these. We then developed a working version of one design, the *GrayArea* prototype, and finally, we evaluated it in a study of 96 participants who were asked to "clean" two of their folders with, and without, *GrayArea*.

In the iterative design stage our research questions were: Do participants want a dedicated demoting interface? And if so, which of the three designs do participants prefer? In the evaluation phase we asked: Will the availability of *GrayArea* reduce folder clutter? Which is easier - deleting files or demoting them using *GrayArea*? And what are participants' attitudes towards *GrayArea*? In addition, we gathered feedback for future improvement of *GrayArea*.

THE USER-SUBJECTIVE APPROACH: DEMOTION

The user-subjective approach is the first design approach developed specifically for PIM systems [7]. It addresses the keep/delete dilemma suggesting demotion as part of the "subjective importance principle" which proposes that the subjective importance of information should determine its degree of visual salience and accessibility. Two sub-principles can be derived: *The Promotion Principle* proposes that very important information items should be highly visible and accessible as they are more likely to be retrieved. *The Demotion Principle* proposes that information items of lower importance should be demoted (i.e. making them less visible) so as not to distract the user. This differs from previous suggestion that attention directed properties will change dynamically over time [23].

The approach also suggests that demotion interfaces should keep information items within their original context. This is a critical difference between demotion and deletion or archiving (e.g. email archiving in MS Outlook and Gmail). In contrast to demotion, archiving and deletion remove the information item from its original context to an archive folder or the Recycle Bin (where it will eventually be removed entirely). Preserving this context is important because PIM research has repeatedly shown that users prefer to retrieve information items by navigating to the location where they stored them [3, 9, 12, 18, 22].

In the past, deletion was the most common strategy not only as a way to avoid user distraction but mainly to clear valuable hard disc memory space. Users had to choose between two main alternatives: keep or delete. However as memory storage has become larger and cheaper, storage space is less of a problem and demotion is a viable third alternative.

The user-subjective approach is deliberately abstract and the demotion principle does not directly specify the way the information item will be made less visible. This abstractness allows for various possible designs (as will be demonstrated in the iterative design section). We return to the user-subjective approach in the discussion section.

TESTING CURRENT SYSTEMS: WORKAROUNDS

Current PIM systems *promote* highly important files making them more visible and accessible using direct manipulation or automatic design. For example, they allow users to manually place files (or a shortcut/alias to them) on the desktop. Alternatively users can access a list of most recently used files (from operating systems menus and specific software menus) on the assumption that recently accessed files are more likely to be used. However these systems currently have no dedicated design that allows users to demote files of low subjective importance.

As part of a larger study [8] we tested how PC users assess and manage documents of different levels of importance. Our 84 participants answered a questionnaire regarding their PIM habits. Results indicated that users exploit existing system design features to retrieve information items of high importance. For example on the average they evaluated that they retrieved 18% of their files using desktop shortcuts, and 12% using recent documents lists (Similar results in a larger sample of Mac users were later obtained in [9]).

More importantly, since participants did not have a dedicated design to demote files they used various workaround strategies (most which do not preserve the file's context): 40% transferred files to a general archive folder, 61% moved them to external memory such as a CD, 32% created a new folder and used the old one as an archive, and 24% created an archive folder within the original folder. Altogether, 79% of the participants used one or more of these alternatives to make their low subjective importance files less visible.

The use of these workaround strategies to demote files, confirmed the demotion principle suggested by the user-subjective approach and indicated to us an unfilled user need for dedicated demoting interfaces. This motivated our subsequent designs.

ITERATIVE DESIGN: THREE PAPER PROTOTYPES

We next designed three demoting interfaces, and conducted an evaluation using low fidelity paper prototypes to obtain early feedback regarding the general principle of demotion and specific feedback about each design.

In each design, we wanted to ensure that items are not removed from the folders where they were originally stored. Retaining the original location means that, the items can be retrieved in their original context so the user does not have to remember an additional storage place associated with them. One simple way to do this is to exploit an existing OS feature: using *chronological order* as a default sorting. Two additional demotion solutions are based on new design schemes: *GrayArea* and *FadedItems*.

Chronological Sorting

Recently accessed information items are generally more likely to be needed than those which have not been accessed for a long time. Thus, an item's most recent access date can often indicate its subjective importance. To allow fast access to important information items, chronological ordering could replace the alphabetical one as a default sorting for file folder lists. This allows the user to ignore unused files appearing at the bottom of the list. Notice, however, that this solution (sorting by *access* time) is different from the current file chronological sorting in MS Windows - which uses the time a file was *changed*. More time-related sorting designs can be found in [14, 15, 16].

FadedItems

In the *FadedItems* interface, users manually change the visual character of information items of low subjective importance making them appear gray rather than black. This makes them less apparent, while more important information items become more salient (see Figure 2). The user can do this by right-clicking on the information item and choosing the Fade option (or the Unfade option to reverse the action). The faded information items appear, by default, at the bottom of the folder's list.

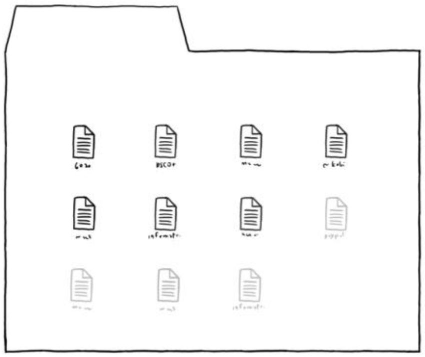


Figure 2: *FadedItems* design scheme.

GrayArea

GrayArea is a folder feature that allows users to drag (or move) information items of low importance to a designated location at the bottom of a folder (see Figure 3). Information items in that area are demoted to the area's gray background, and are presented in a small space (if information items exceed the allotted gray area, a scroll bar is used). The demoting action can be reversed by dragging (or moving) information items from the gray area back to the standard folder space.

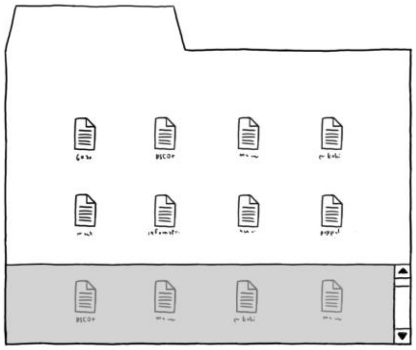


Figure 3: *GrayArea* design scheme.

There are three important differences between these design schemes:

Direct manipulation vs. automation: GrayArea and *FadedItems* use direct manipulation [26], users themselves specify items to be of low importance. In contrast, in *Chronological Sorting* the OS automatically assigns long untouched information items to the bottom of the list. Notice that importance (as a criterion for demotion) is differently defined in the two kinds of designs: In direct manipulation, importance is defined subjectively by the user. In the automatic process, importance is defined by access time.

Designated vs. undesignated location: In *GrayArea*, information items of low subjective importance are kept in a separate designated location (the gray area at the bottom of the folder). In the *FadedItems* design they are kept in their original location along with all other items (though they can be differentiated by their fading color). In *Chronological Sorting*, items of low subjective importance appear at the end of the items list, but with no physical differentiation between the two classes (important and unimportant files).

Continuous vs. categorical demoting: In *Chronological Sorting*, the importance of the items is a continuous variable - the lower on the list the less important. In our two new proposed design schemes, information items are either unimportant (dragged to a gray area or faded to gray) or important. Unlike continuous demoting, categorical demoting is reversible and users can easily un-demote files.

Having designed these demoting schemes, our aims in the low fidelity evaluation were to obtain general feedback about the demotion principle, choose which of the three demoting designs was the best to develop and formally evaluate, and finally to obtain detailed feedback about that specific design to improve its implementation.

Research Questions

1. Do participants want a dedicated demoting interface?
2. Which of the three designs do participants prefer and why?

Method

Participants were 79 personal computer users studying in Israeli Universities, aged 19-47. Of these participants, 55 were females, 21 males, and 3 did not disclose their gender. All participants completed a paper and pencil questionnaire which included Likert-type (1 - not at all, 2 - to a small extent, 3 - to some extent, 4 - to a large extent, 5 - to a very large extent) and open ended questions.[1] The questionnaire began with an explanation of what was meant by "files of low importance". Then each of the design schemes was concretely presented (the low fidelity paper prototype along with a detailed explanation of the interface and its operation), followed by questions regarding it. Order of presentation was counter-balanced between participants. Finally participants were asked to choose between four options: their current folder design (which offers no dedicated way to demote files) and the three proposed demoting design schemes.

Results

The main quantitative results relevant to our two research questions are presented in Table 1.

Design Scheme	**Preference** ***N (%)***	**Like to Use** ***Mean* (*SD*)**
Chronological Sorting	20 (28.6%)	3.24 (1.29)
GrayArea	18 (25.7%)	2.97 (1.26)
FadedItems	21 (30.0%)	3.18 (1.2)
Current design	11 (15.7%)	-

Table 1: Participants' preferred design (comparing three demoting designs to their current one), and the extent they would like to use each of the three demoting designs (on a 1-5 Likert scale).

Do Participants Want a Designated Demoting Interface?

The left column of Table 1 shows that when asked to choose between the four design schemes (their current system and the three new designs) only 11 out of the 70 participants who answered this question (16%) chose their current one, while the other 59 participants (84%) preferred one of the three new designs. When asked to evaluate their *chosen* interface their answer on a 1-5 Likert scale was 4.22 on average (*SD* = 0.64).

We next analyzed the reasons participants gave for choosing one of the proposed designs. Their answers can be

[1] To receive the questionnaire, please contact the first author.

assigned to the following categories: *Reduction of visual overload* ("these files [of low importance] don't distract me [any more]"), and *tidiness* ("creates some order from a mess"); *Helps retrieve important files more easily* ("highly important files stick out") and *efficiency* ("helps find [files] fast"); *Accessibility of files of low importance* ("it puts important things in the foreground, but keeps the less relevant things on the surface"); and *Presents an alternative to deletion* (regarding *GrayArea*: "You can drag it to a different area instead of deleting it"). The 11 participants who preferred their current interface to the novel ones explained their choice by providing conservative reasons ("I've got along well with the way I worked up until now").

Which of the Three Designs do Participants Prefer?

When asked to what extent they would like to use each of the three new interfaces, average results ranged from 2.97 to 3.24 on a 1-5 Likert scale (right side of Table 1). However, there was no preference for a specific design: a within-subject analysis of variance with repeated measures was not significant ($F(2, 150)=1.17$, $p>0.05$). When participants were asked to choose their preferred design, their preferences were very similar: each of the new designs was chosen by 25%-30% of the participants (see Table 1, left column). Thus the results don't show differences in relative preference between the three design schemes.

We then analysed participants' spontaneous comments regarding the main differences between the proposed designs. We found 12 comments in support of direct manipulation and 16 in favor of automatic methods. The results were clearer regarding the question of file location: 26 spontaneous comments supported presenting unimportant items separately from other files, while 15 preferred to see both unimportant and important files together at the same location. There were no spontaneous comments regarding the continuous vs. categorical demoting distinction.

Choosing a Prototype

Our results support the notion of a dedicated interface for demoting files of low subjective importance, as 84% of participants preferred one of the three new design schemes to their current design. They strongly wanted to have such an interface implemented, and most of them referred to at least one aspect of demoting when describing the benefits of their chosen new design. However they did not indicate a preference for one specific design scheme. To make an informed decision regarding the choice of an interface for development and evaluation of a full prototype we needed further investigation of each.

Chronological Sorting: As part of a larger unpublished study, we videoed 233 participants navigating to target files using their PCs ($N = 190$), Macs ($N = 38$) and Linux ($N = 5$) systems. We analyzed the last step of the navigation (after they had found the folder that contained the target file). We were surprised to find that only 3 of these participants (1%) ever used chronological sorting. These results make Chronological Sorting look less promising: why change the default to an option users hardly ever use?

FadedItems: We implemented *FadedItems* using Windows XP faded presentation (an option which is not used by default). This is currently used for hidden system files as well as user documents which programs backup automatically (i.e. for unexpected terminations). We quickly created a working prototype, changing this option to show faded files, programming the right click interaction and adjusting the sorting so that the faded items appeared at the bottom of the file list. However as soon as we developed this prototype we realized that it did not scale well. Folders soon became overcrowded with hidden files - adding to distractibility instead of lowering it. In addition participants declared a preference for having a designated location for demotion to keep them in a place where "the mess stays".

GrayArea: GrayArea seemed to have four advantages over the two other demoting interfaces: (a) Participants' spontaneous comments showed a clear preference for presenting unimportant items in a designated location separately from other files. (b) Unlike *Chronological Sorting* it does not require any changes in sorting that may potentially reduce usability. (c) Its interaction (drag and drop) is simple and straightforward. (d) Categorical demotion (files are either important or unimportant) enables the application of additional features: reversibility, compression, and search extensions (which we describe further in the discussion section). (e) It is distinctively a novel interface. These led us to choose *GrayArea* for subsequent formal evaluation.

Using Feedback to Improve *GrayArea*

We also analyzed user comments to incorporate their feedback into the final *GrayArea* design. One participant was concerned that when the number of files becomes large, important files will not be shown while demoted ones in the *GrayArea* will. Other participants pointed out that there are times when they prefer not to see files within the *GrayArea*. As a result, we changed the *GrayArea* prototype design from having a fixed height that included a line of files (see Figure 3) to having variable height: users can drag the upper border of *GrayArea* and adjust its height (see Figure 1).

DEVELOPMENT: WORKING JAVA PROTOTYPE

In developing *GrayArea* we decided not to attempt to extend Windows Explorer. This would have been a substantial undertaking and since Explorer is proprietary software it would be difficult to effectively integrate *GrayArea* with the conventional Explorer view. Extending Explorer would also present a further problem for users - the file collection is typically users' most valuable PIM collection [6, 11] and we did not want participants to take any risks (real or hypothetical) by allowing us to make any changes to their filestore.

We therefore took a different approach and developed a working prototype as a Java application which simulated and extended the typical filesystem Explorer interface. The interface was essentially a view on the users' filestore where any interface actions that would normally have resulted in a permanent change to the filestore (e.g. deleting a file) only affected how the view was rendered. Thus we were able to design and test a working prototype without manipulating the underlying files. The properties of the view could either be deleted when the application was closed, or stored permanently, so that view changes were preserved when our application was rerun.

We implemented functionality that allowed users to navigate in their filestore (by selecting folders, using the familiar Forward, Back and Up buttons, or by selecting disk drives from a drop down menu). We also allowed users to sort files by name, date or size. Users could also open files by double clicking on them to scrutinize them to evaluate their importance. Drag & Drop and Cut & Paste functionalities were used to handle the movement of files to and from the *GrayArea* – but again this was a property of the view rather than a change to the filestore. As a result of early user feedback, demotion of a file to *GrayArea* could be reversed using the control-z shortcut.

We compared the *GrayArea* prototype with a control UI that directly emulated standard Explorer behavior (navigation, movement, opening and deletion of files). For the above reasons (user concern about deletion of their collections) we again didn't want to use their real Explorer interface. The *GrayArea* prototype and the control UI were identical in all respects, with the exception of being able to demote files to the *GrayArea*.

EVALUATION: CLEANUP WITH & WITHOUT *GRAYAREA*

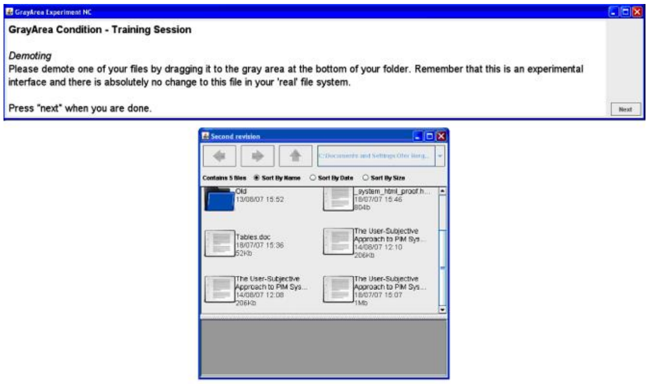
The main goal of *GrayArea* is to support users in managing their information by demoting unimportant information items, allowing them to focus on more important information. We wanted to gather subjective feedback about the interface as well as behavioral data about how people used it in the context of a realistic user task. We therefore evaluated our prototype by asking users to carry out a common PIM maintenance activity [12, 31], i.e. to "cleanup" two of their folders in two conditions. They did this once with *GrayArea* and once without it with no time constrains. We wanted to know whether providing *GrayArea* would lead them to remove more unimportant files from a given folder during cleanup than with the standard OS setup. We logged participants' cleanup behaviors in detail. We also asked them to complete a short questionnaire to determine which of the two 'cleaning' options (demote/delete) was easier to decide on, to obtain feedback about participants' attitudes towards *GrayArea*, as well as suggestions for future development and evaluation.

Research Questions

1. Will the availability of *GrayArea* reduce folder clutter?
2. Which is easier - deleting files or demoting them using *GrayArea*?
3. What are participants' attitudes towards *GrayArea*?

Method

Participants were 96 Windows XP users, mostly from the U.K., U.S.A. and Israel. Their age ranged from 19 to 62 years (M = 35.6, SD = 10.66). 55 were males. We contacted participants by sending emails to large groups of people (e.g. workers and students in our universities) asking them to volunteer for the experiment. Participants were asked to download a Java application. After opening the application screen instructions provided an explanation of the experiment, followed by step-by-step instructions related to a simulation of their folders which appeared under the instructions window (See Figure 4).



GrayArea Condition - Training Session

Demoting
Please demote one of your files by dragging it to the gray area at the bottom of your folder. Remember that this is an experimental interface and there is absolutely no change to this file in your 'real' file system.

Press "next" when you are done.

Figure 4: Instructions window (top) and folder window (bottom).

We used a within-subjects design. Participants were asked to navigate to two folders and perform the cleanup task in the two conditions. In the *GrayArea* condition they could choose between keeping/deleting/demoting. In the standard condition (as in current systems) there was no *GrayArea* - so their choice was between keeping and deleting. The standard condition served as a control to *GrayArea*. Order of conditions was counter balanced between participants.

Each condition started with a training phase to make sure that participants were familiar with various actions they needed to perform for the cleaning tasks. These include standard actions (navigating to a folder, opening files, resizing folder windows, and deleting files). These actions were performed in an identical or similar way to Windows Explorer; however, before starting the cleaning tasks we wanted to make sure that the participants realized that they could perform all these actions in our simulated environment. In the *GrayArea* condition, the training session also included specific training relating to *GrayArea*: demoting a file to the *GrayArea*, restoring it to its original place, and changing *GrayArea* size. All participants had to successfully complete these training phases before proceeding to the experimental phases, in each condition. We also demonstrated to participants that their cleaning actions (deletion and demotion) induced changes only in the experimental software and had no effect on the real files in their computers.

In the *GrayArea* condition of the experiment proper, participants were given the following instructions: *"Please go over each of the files in the folder and choose one of the following three options: (a) If it is important to you, leave it exactly as it is (do nothing); (b) If it is of no importance to you, and you are sure you won't need it anymore, delete it; (c) If it is of low importance, but you think you may possibly want to use it in the future, put it in GrayArea."* In the standard condition, instructions were identical with the omission of the demotion option. After participants completed both cleaning tasks a log file that recorded their actions was automatically sent to us. Unfortunately, for 18 of the 96 participants these log emails were blocked by the participants' anti-virus software and did not reach us, leaving us with 78 logs. Participants then completed a Web questionnaire containing 1-5 Likert type questions (1 - strongly disagree, 2 – disagree, 3 - neither agree nor disagree, 4 – agree, 5 – strongly agree), other multiple choice questions and open ones.

Results

Preliminary Tests

We conducted three preliminary tests to validate the log data and exclude alternative explanations of our findings.

Folder Size: The number of files in the chosen folder was not significantly different between the standard and *GrayArea* conditions, so results we report are not due to users choosing different size of folders in the two conditions (t(77)=0.88, p>0.05).

Folders Age: We logged the *recently changed* date for each file in the cleaned folders, and defined *folder age* as the # days that passed since this recently changed date averaged over all files in the folder. A paired t-test shows that on average *folder age* in the *GrayArea* condition (*M* = 552, *SD* = 578) was not significantly different from the standard condition (*M* = 498, *SD* = 552), t(77)=0.79, p>0.05. Thus, any difference in participants' behavior under the two conditions can't be explained by a difference in folder age.

Order of Conditions: Order of condition had no significant effect on results, allowing us to combine log results from counterbalanced groups.

Will the Availability of GrayArea Reduce Folder Clutter?

The number of files kept, deleted and demoted was calculated for each participant as well as the corresponding percentages. These percentages were then averaged across participants for each condition. Participants' logged actions in the two cleaning tasks appear in Figure 5 and Table 2.

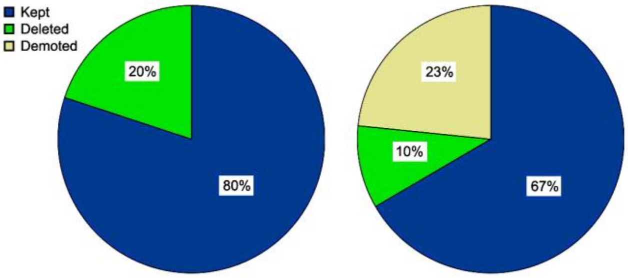


Figure 5: Average percentages of keeping and cleaning actions in the standard condition (left) and in the *GrayArea* condition (right).

We wanted to know whether *GrayArea* led participants to reduce folder clutter by 'cleaning' a higher percentage of files than in the standard condition. We conducted a paired t-test comparing the percentage of cleaned files in the *GrayArea* condition (%demoted + %deleted) with the percentage in the standard condition (%deleted). The average percentage of cleaned files in the *GrayArea* condition (33%) was higher than in the standard condition (20%), t(77)=3.94, p<0.001. The 13% difference between conditions means that there was less folder clutter after cleaning with *GrayArea* than in the standard condition: In the standard condition on average 80% of files were kept (38 files) whereas in the *GrayArea* condition only 67% of files were kept (27 files).

Action	**Kept**	**Deleted**	**Demoted**	**Total**
Condition	***Mean* (*SD*)**	***Mean* (*SD*)**	***Mean* (*SD*)**	***Mean* (*SD*)**
Standard no.	38.08 (77.84)	3.2 (3.91)	-	41.28 (77.18)
%	80% (22%)	20% (22%)		100%
GrayArea no.	27.33 (47.1)	1.73 (2.37)	4.13 (4.64)	33.19 (46.98)
%	67% (28%)	10% (12%)	23% (25%)	100%

Table 2: Participants' Logged Actions: average number and average percentages of files kept, deleted and demoted under the two conditions.

Subjective judgments mirror these results as reflected by participants' answers to the questionnaire: When asked about the extent to which they agreed/disagreed with the sentence "*GrayArea* can help reduce the clutter in folders" their average answer on a 1-5 Likert scale was 3.74 (*SD* = 0.92). To conclude, participants cleaned more files in the *GrayArea* condition leaving less actual and perceived clutter than in the standard condition.

Notice too, that the average deletion percentage dropped from 20% in the standard condition to 10% in the *GrayArea* one. Thus, we can view the 23% of demoted files in the *GrayArea* condition as comprising of 10% of files that in the existing keep/delete interface would probably have been deleted (although it seems participants prefer not to delete them), along with 13% of files that would have been kept (although participants prefer not to keep them).

Which is Easier - Deleting Flies or Demoting Them Using GrayArea?

A strong behavioral indication that demotion is easier than deletion is that when given the two options (in the *GrayArea* condition) participants chose to demote more files (23%) than delete files (10%), t(77)=4.6, p<0.001. Participants' answers to the questionnaire reflect their behavior: In the questionnaire we asked “which is easier?” letting participants choose between: “to demote a file using *GrayArea”*, “no difference” and “to decide on deleting files”. Of our 96 participants 77 (81%) selected “to demote a file using *GrayArea”*, 15 (16%) selected “no difference”. Only 3 participants (3%) selected the ‘to decide on deleting files’ option, and one participant did not answer. The vast majority of participants thus found it easier to decide on demoting files than on deleting them.

The demotion decision may have been perceived as easier because it is less ‘final’; with demotion users can always find the file in the context of the folder where it was stored and reverse the demotion.

What are Participants’ Attitudes towards GrayArea?

Participants were asked to what extent they agree/disagree with the sentences: “*GrayArea* complicates the interface” and “I would like *GrayArea* to be implemented in my file folders”. Their average answers on a 1-5 Likert scale were 2.57 (*SD* = 0.97) and 3.61 (*SD* = 0.94) respectively. We also gathered more open-ended feedback using the following question: “What do you like/dislike about *GrayArea*?” Of the 64 participants who answered this question 54 (83%) responded with positive and often enthusiastic feedback (“*OH MY GOD this is genius. I totally want to be able to say ‘Store this, but don't bug me with it unless I am specifically hunting data.’ That is exactly what GrayArea does. Please release it soon so that I can use it to clarify my life!*”). One (2%) gave an answer containing both positive and negative feedback. Just 9 participants (14%) gave negative feedback (e.g. “*I think this is only useful for people with a certain mentality, i.e. disorganized squirrels*”).

To measure participants’ inclination to use *GrayArea* if it were available in future we asked them to rate the following statement on the 5 point Likert scale: “If *GrayArea* was part of my future Windows file folders interface I would use it”. Participants’ average answer was 3.82 (*SD* = 0.9), and the distribution of results show that 12% of the participants responded negatively (under 3), 13% were undecided (3), and 65% of the participants answers were positive (over 3). This suggests that most participants would be inclined to use *GrayArea* if provided in their operating system.

GENERAL DISCUSSION

The results of our studies give strong and consistent evidence for users’ desire to use demoting interfaces and *GrayArea* in particular. With current systems (Windows and Mac) 79% of our 84 participants already worked around the lack of designated demoting interfaces using various alternative demoting user strategies [8]. When presented with three low fidelity paper prototypes at the iterative design stage, 84% of our 70 participants preferred using one of the demoting interfaces to their current interface. In the evaluation stage, our 96 participants cleaned up a higher percentage of their files in the *GrayArea* condition (33%), than in the standard condition (20%) - reducing the clutter in their folder by 13%. Eighty-one percent of participants indicated that it was easier for them to demote files than to delete them. Finally the majority of participants expressed positive attitudes towards *GrayArea* and indicated that they would use it if provided by their operating system. These results suggest that current delete/keep options are too rigid for current users, who value the greater flexibility that demotion provides.

Design Implications

User comments about the prototype also suggested various ways in which we could extend and improve demotion UIs.

Folder Demotion: Several participants complained that they could not demote entire folders to *GrayArea*. Actually the *GrayArea* prototype does allow folder demotion, however, this ability was disabled in the experiment to keep it as simple as possible. Folder demotion will be tested in future research.

Automatic Demotion of Old Versions: Old versions of documents are a particular class of files of low importance. Users typically don’t want to delete old versions of their files in case they ever need them [31]. As a result they find it hard to distinguish between newer and older versions of a file. *Old’nGray* [8] solves this problem by automatically ‘fading to gray’ old versions of documents when a new version is stored on the computer. The OS could do this by adding a hidden identifier to each file, enabling it to recognize versions of the same file stored under different names. In this way, users could distinguish the latest version from older ones, allowing them to easily retrieve the new version as well as demote older ones further away into *GrayArea* (See Figure 6).

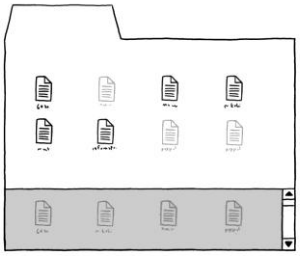


Figure 6: Automatic demotion of old versions.

Compression: Files placed in *GrayArea* could be automatically compressed as they are unlikely to be needed. Dragging them back to their original location in the folder uncompresses these files.

Demoted Search Results: When performing search, demoted files (from either *GrayArea* or *Old'nGray*) could appear faded in the results list, as they are less likely to be retrieved (see Figure 7). We predict that this would decrease irrelevant distraction and reduce retrieval time.

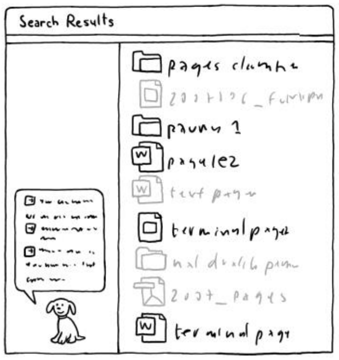


Figure 7: demoted search results are faded as to not attract attention.

Reflections on the User-Subjective Approach

We now re-examine our design approach in the light of feedback from the design cycle. *GrayArea* is just one possible way to implement the demoting principle and the demoting principle is only one application of the subjective importance principle advocated by the user-subjective approach. The user-subjective approach exploits a unique requirement of PIM systems [7]. In other systems, information is stored and organized by information professionals (e.g. Web site developers) for users who retrieve information according to their needs. To cater to the needs of different users and facilitate information retrieval, information professionals use general and objective attributes of the information for its organization. PIM systems are unique in that the same person who stores the information and decides on its organization is the one who later retrieves it. The user-subjective approach takes advantage of this unique feature and suggests that PIM systems should make systematic use of subjective, user-dependent attributes to facilitate the organization of personal information and its retrieval. This could either be done automatically by the system or manually by the user using direct manipulation.

One such subjective attribute is the importance of an information item. Importance is a subjective attribute because it is user-dependent: the same information item can be of the highest importance to one person and completely unimportant to the other. The demotion principle (as well as the promotion one) suggests how PIM system design could help the user make use of the subjective importance attribute. The user-subjective design approach presents many subjective attributes, design principles, and user-subjective design schemes to be explored [8]. The positive results regarding *GrayArea* obtained here provide evidence in favor of the user-subjective approach as a whole and should encourage the evaluation of further novel designs related to other user-subjective principles.

CONCLUSIONS

The results presented in this paper support the design principle of allowing to *demote* files of low importance. *GrayArea*, a design we implemented to explore an interface for file demotion, was positively evaluated by participants. We deliberately did not attempt to write any patents on *GrayArea* as we want users of all operating systems to benefit from it. It is our hope that the next generation of Windows, Mac and Linux operating systems contain *GrayArea* or similar demoting features, addressing a crucial but commonly overlooked aspect of everyday computer use.

ACKNOWLEDGEMENTS

We thank our participants for their time and efforts. This research was partially funded by the European Union Marie Curie Grant, TOK30008.

REFERENCES

1. Abrams, D., Baecker, R. and Chignell, M. Information archiving with bookmarks: Personal web space construction and organization. In *Proceedings of the sigchi conference on human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co., (1998), 41-48.
2. Barreau, D. K. Context as a factor in personal information management systems. *Journal of the American Society for Information Science 46* 5(1995), 327-339.
3. Barreau, D. K. and Nardi, B. A. Finding and reminding: File organization from the desktop. *SIGCHI Bulletin 27*, 3 (1995), 39-43.
4. Bellotti, V., Ducheneaut, N., MarkHoward, Smith, I. and Grinter, R. E. Quality versus quantity: E-mail-centric task management and its relation with overload. *Human Computer Interaction 20* (2005), 89–138.
5. Bellotti, V. and Smith, I. Informing the design of an information management system with iterative fieldwork In *Proceedings of the conference on designing interactive systems: Processes, practices, methods, and techniques* ACM Press, (2000), 227-237.
6. Bergman, O. *The use of subjective attributes in personal information management systems*. Ph.D. Dissertation (in Hebrew), Tel Aviv University, Tel Aviv, Israel, 2006.
7. Bergman, O., Beyth-Marom, R. and Nachmias, R. The user-subjective approach to personal information management systems. *Journal of the American Society*

for Information Science and Technology 54, 9 (2003), 872-878.

8. Bergman, O., Beyth-Marom, R. and Nachmias, R. The user-subjective approach to personal information management systems design - evidence and implementations. *Journal of the American Society for Information Science and Technology 59*, 2 (2008), 235-246.
9. Bergman, O., Beyth-Marom, R., Nachmias, R., Gradovitch, N. and Whittaker, S. Advanced search engines and navigation preference in personal information management. *ACM Transactions on Information Systems 26*, 4 (2008), 1-24.
10. Bergman, O., Boardman, R., Gwizdka, J. and Jones, W. P. Personal information management. In *Chi '04 extended abstracts on human factors in computing systems*. ACM Press, (2004), 1598-1599.
11. Boardman, R. *Improving tool support for personal information management*. Ph.D., Imperial College, London, 2004.
12. Boardman, R. and Sasse, M. A. "Stuff goes into the computer and doesn't come out": A cross-tool study of personal information management In *Proceedings of the Sigchi conference on human factors in computing systems*. ACM Press, (2004), 583-590.
13. Bruce, H. Personal anticipated information need. *Information Research 10*, 3 (2005).
14. Cutrell, E., Dumais, S. T. and Teevan, J. Searching to eliminate personal information management. *Communications of the ACM 49*, 1 (2006), 58-64.
15. Freeman, E. and Gelernter, D. Lifestreams: A storage model for personal data. *SIGMOD Record 25*, 1 (1996), 80-86.
16. Gemmell, J., Bell, G., Lueder, R., Drucker, S. and Wong, C. Mylifebits: Fulfilling the memex vision. In *Proceedings of the Proceedings of the tenth ACM international conference on Multimedia*. ACM, Juan-les-Pins, France, 2002.
17. Harper, R., Rodden, T., Roggers, Y. and Sellen, A. *Being human: Hci in 2020*. Microsoft, Cambridge, UK, 2008.
18. Jones, W., Phuwanartnurak, A. J., Gill, R. and Bruce, H. Don’t take my folders away! Organizing personal information to get things done. In *Chi '05 extended abstracts on human factors in computing systems*. ACM Press, (2005)
19. Jones, W. and Teevan, J. *Personal information management*. University of Washington Press, 2007.
20. Jones, W. P. Finders, keepers? The present and future perfect in support of personal information management. *First Monday 9*, 3 (2004).
21. Kahneman, D. and Tversky, A. *Prospect theory: An analysis of decision making under risk*. Eugene, Ore., 1979.
22. Kirk, D., Sellen, A., Rother, C. and Wood, K. Understanding photowork. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, Montreal, 2006.
23. Malone T. W. How do people organize their desks? Implications for the design of office information systems. *ACM Trans. Office Inf. Syst. 1*, (1983), 99–112.
24. Neisser, U. Visual search. *Scientific American 210*, 6 (1964), 94-102.
25. Sharp, H., Rogers, Y. and Preece, J. *Interaction design: Beyond human-computer interaction*. John Wiley & Sons, 2007.
26. Shneiderman, B. *Designing the user interface: Strategies for effective human-computer interaction*. Addison-Wesley Longman Publishing Co., Inc., 1997.
27. Treisman, A. and Gelade, G. A feature-integration theory of attention. *Cognitive Psychology 12* (1980), 97-136.
28. Whittaker, S. Supporting collaborative task management in email. *Human Computer Interaction 20*, 49-88 (2005).
29. Whittaker, S. and Hirschberg, J. The character, value, and management of personal paper archives. *ACM Transactions on Computer-Human Interaction 8*, 2 (2001), 150-170.
30. Whittaker, S., Jones, Q. and Terveen, L. Contact management: Identifying contacts to support long term communication, *CSCW 2002*, New York, ACM Press.
31. Whittaker, S. and Sidner, C. Email overload: Exploring personal information management of email. In *Proceedings of the sighchi conference on human factors in computing systems: Common ground*. ACM Press, (1996), 276-283.