

# Understanding File Access Mechanisms for Embedded Ubicomp Collaboration Interfaces

Anthony Collins,<sup>1</sup> Anastasia Bezerianos,<sup>2,3</sup> Gregor McEwan,<sup>2</sup>

Markus Rittenbruch,<sup>2,1</sup> Rainer Wasinger,<sup>1</sup> Judy Kay<sup>1</sup>

<sup>1</sup>School of IT  
University of Sydney  
NSW Australia  
{anthony,wasinger,judy}  
@it.usyd.edu.au

<sup>2</sup>HxI Braccetto  
National ICT Australia  
Eveleigh NSW Australia  
{gregor.mcewan,markus.  
rittenbruch}@nicta.com.au

<sup>3</sup>MAS Laboratory  
SAP Business Objects Chair  
École Centrale Paris  
France  
anastasia.bezerianos@ecp.fr

## ABSTRACT

This paper explores the nature of interfaces to support people in accessing their files at tabletop displays embedded in the environment. To do this, we designed a study comparing people's interaction with two very different classes of file system access interface: *Focus*, explicitly designed for tabletops, and the familiar hierarchical *Windows Explorer*. In our within-subjects double-crossover study, participants collaborated on 4 planning tasks. Based on video, logs, questionnaires and interviews, we conclude that both classes of interface have a place. Notably, *Focus* contributed to improved collaboration and more efficient use of the workspace than with *Explorer*. Our results inform a set of recommendations for future interfaces enabling this important class of interaction — supporting access to files for collaboration at tabletop devices embedded in an ubicomp environment.

## Author Keywords

Tabletop interface, single display groupware, file system UI

## ACM Classification Keywords

H.5.3 Information Interfaces and Presentation: Group and Organization Interfaces

## General Terms

Human Factors, Experimentation, Design

## INTRODUCTION

Emerging ubicomp environments will include a range of new ways for people to interact with their digital artefacts: at interactive tables, wall displays and making use of carried projectors. These will complement people's increasingly sophisticated carried devices, such as smart phones, as well as more conventional portable computing devices such as laptops. Unlike a carried device, these embedded interaction

displays will provide large screens that are effective for people to collaborate on tasks like planning and to share information and discuss it.

If this is to be possible, an essential prerequisite is an effective mechanism for people to access their files. At this stage, it is unclear how to support collaborative file access for *Single Display Groupware (SDG)*. We can certainly draw inspiration from file access mechanisms that are currently available on desktop computers. Notably, these are familiar to most potential users of embedded ubicomp display interfaces. However, they were designed for a very different use, by a single user who is seated before a display. By contrast, these new embedded displays will commonly be used by small groups of people and, in the case of tabletop interfaces, the users may be seated around the table, at different orientations. Another important constraint on interaction with embedded ubicomp display interfaces is the limited input, with no mouse or keyboard available. This imposes a severe restriction on the number of different gestures that are natural and easy to learn.

This paper describes research intended to inform our understanding of how to support effective interaction for the fundamental task of accessing files for small group interaction at tabletop embedded ubicomp display interfaces. To do this, we selected two very different mechanisms and studied the ways that people made use of them for a collaborative planning task. At this early stage of exploring ubiquitous file system access, one natural mechanism is the existing widespread *Windows Explorer* because it is so familiar, being very similar to all commonly used desktop file system interfaces. Notably, it relies upon the hierarchical structure of file systems and the user navigates this hierarchy to access the particular files that they need. For a very different mechanism, we considered a search-like approach that has become standard on desktops, and which operates effectively without reliance on the file system hierarchy. However, we rejected it because the user needs to type the search query and, while it works well as a last resort mechanism for users who cannot re-find files in their desktop hierarchy, it is unsuited to the limited input restrictions of embedded ubicomp display interfaces. Also, it does not support *navigational* file access, which is preferable for file systems [3].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*UbiComp 2009*, Sep 30 – Oct 3, 2009, Orlando, Florida, USA.

Copyright 2009 ACM 978-1-60558-431-7/09/09...\$10.00

So, the second mechanism for this study was *Focus* [4]. This was explicitly designed for tabletop interaction. For example, it presents files oriented around the display so that it does not favour any one orientation. It also aids collaboration, supporting file access across multiple file systems. To take account of the limited input possibilities, it supports *similar-to-focus* file navigation mechanisms: this means that the user selects a *focus file* and dwelling on this causes the retrieval of all *similar* files from all the users' file systems. This could, potentially, be from the users' carried devices or their other computers.

The remainder of this paper is organised as follows. We first provide a background of related work. Next, we introduce the two interfaces that we have studied. Then, we describe the design of the study and report the results. We conclude by discussing the implications of our findings for file system access at embedded ubicomp interfaces.

## RELATED WORK

The vision for tabletops includes a role as a pervasive interface that people can walk up to and use as needed. When people need to access their files, these may be on a carried device or on a fixed, possibly remote machine. There are two parts to such file access: transferring files onto the tabletop; and then navigating the file collection loaded onto the table. We review both of these and then some of the key, known challenges of tabletop interaction.

Interface issues in the transfer of files from the user's personal storage to the tabletop include management of security and privacy, and support for selecting which files should be made available to the tabletop. Olsen et al. [9] created a new interface that essentially integrates a small screen device that the user carries with the information displayed on the table. This work took careful account of security issues, and involved interaction with a single file, by displaying parts of it in the tabletop. One approach to transferring files is illustrated in *UbiTable* [16], where people privately access files and then selectively transfer them to a shared region of the tabletop. A similar approach uses tangible drawers [6]: here, too, users privately access their file collection, moving selected files into a shared workspace on the tabletop. *MultiSpace* [5] takes a different tack, based on seamlessly moving files between several different devices. All these systems assume individual file access: collaboration takes place after specific files have been transferred from a private space or device and opened on the tabletop. Notably, all files are presented on the tabletop as a flat, unstructured collection. Collins et al. [4] explored remote file access across the computers of all users at the tabletop. Here, users were able to retrieve similar information across all users' file collections with a single interface action.

We now consider the interfaces for accessing files that have been loaded onto the tabletop. Some work has explored the case of small collections of information, such as digital photographs [2, 7], or a virtual desktop of files [1]. In a quite different approach, *TeamSearch* [8] supported search of large collections of digital images by forming Boolean

queries with tokens. This highlighted benefits of collaborative searching, in contrast to independent individual searches on information at the tabletop and the importance of facilitating collaboration and awareness among group members. The *Personal Digital Historian* project [17] enabled small groups of people to access large collections of historical information at a tabletop. A study of their techniques for hierarchical visualisation and navigation highlighted "clutter" and "over crowding" within groups of information as problems in the design. Our previous work [4] reported a usability study that compared a new hierarchical file browser against *OnTop* (an earlier version of *Focus*, for associative access to multiple file systems from the tabletop). Both these interfaces were designed specifically for tabletops. This invited the question of how people would perform when using more familiar, conventional tools, such as *Windows Explorer*, at a tabletop. One study of *Windows* on a tabletop involved a single person using a large tabletop as their primary work computer for 13 months [18]. This suggests a conventional operating system may be usable, although the poor keyboard support was a severe impediment.

Another important set of issues apply for the design of tabletop interfaces. *Social protocols* may restrict people's ability to interact with all interface elements. For example, people may need to reach into their partner's area to touch a window toolbar or the task-bar. Ryall et al. [12] found that people tend to interact directly in front of them, showing reluctance to access or interact with objects directly in front of another person. Similarly, Scott et al. [14] found that people naturally establish a personal working area directly in front of them when using a large tabletop. *Orientation* also poses challenges: a document that is the right way up for one user will be upside-down for a user at the opposite end of the table, and at an angle for those in other positions. This may cause problems, especially for text readability. *Clutter* and its management is another problem. As tabletop interaction normally has input only from a stylus or touch, without a keyboard, the constraints on the interaction design differ from those in conventional interfaces. Notably, for tabletop hardware with touch for input, the *fat finger* effect must be taken into account.

## INTERFACE OVERVIEWS

The importance of file access and the limited understanding of how to support it in pervasive computing environments makes it timely to study potential approaches to designing interfaces for this. We now explain how we configured *Windows Explorer* for the study, to provide insights into that access approach based on the hierarchical file system model. We then introduce *Focus*, which serves as an example of a fundamentally different approach.

### Hierarchical File Access with Windows Explorer

The *Windows Explorer* interface initially presented two separate windows, one for each participant's file collection. *Explorer* can be configured to act either as a *spatial* file interface, where each folder is represented in a distinct window onscreen, or as a *navigational* file interface, where all browsing of the file hierarchy is contained within a single

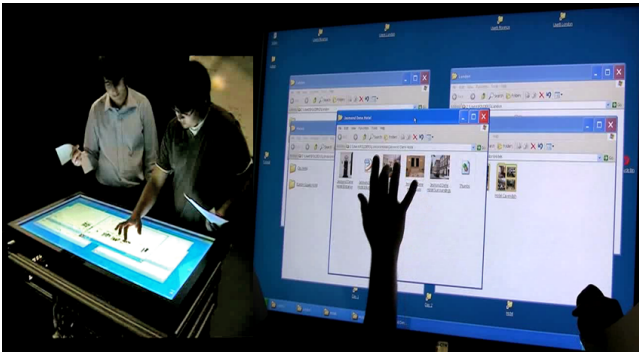


Figure 1. Two people accessing their personal file collections at a tabletop using Explorer (left), and a view of the screen (right) showing their separate file collections on each side of the table, with the folder they are currently discussing moved to the middle.

window. To support multiple people accessing their file collections, it is natural to assume that each person would use their own separate window to access files, and so we chose a spatial configuration for our study. Figure 1 shows our set-up, where two people are accessing their file collections on separate sides of the table.

A distinguishing feature of tabletop interface design is the relatively large workspace provided for users of the tabletop. Some features of *Explorer* (and the *Windows* operating system) would have side-effects on other users: for example, window maximisation — a feature which is often the default for many application windows. The large space available at a tabletop makes it sensible for users to be able to spread their folder and file windows around the table, rather than being restricted to a single window to browse a hierarchical collection. For example, in Figure 1, a person is able to move a specific folder to the middle of the table for discussion, while leaving the rest of the file collection in their personal area.

To support direct-touch tabletop interaction, *Explorer* can be configured to be usable with a relatively thick pointer, such as a human finger. Window toolbars and buttons were enlarged to be easily selectable by touch, and a ‘thumbnail’ view of folder content provided larger selection targets for opening files. We configured *Explorer* to open files with a single touch, instead of double-clicking.

Since we wanted to maintain and exploit the familiarity of *Explorer*, we retained many of its features. For example, *Explorer* assumes a fixed orientation of a single user, meaning that interface objects cannot be reoriented. This might be a problem for people viewing and interacting with the screen at different positions around the tabletop. Second, *Explorer* responds to only one touch (or click) at a time, as there is only a single mouse cursor present on the screen.

### Associative File Access with Focus

In the last section, we explained the *configuration* for the familiar *Explorer*; for the alternate interface, *Focus*, a new experimental system, we now briefly describe the user view.



Figure 2. Two people accessing their personal file collections at a tabletop using Focus (left), and a view of the screen (right) showing similar files to the current ‘focus’ file (a map of tourist attractions, which has also been enlarged by one of the people).

For a more detailed description of the design and implementation of its predecessor, see [4]. *Focus*, shown in Figure 2, is an *associative* file system interface that retrieves all files related to the current *focus file*, across each computer of the users at the tabletop. To make their files accessible at the table, users run the *Focus Exporter* application on their personal computer, making their whole file system, or specific parts of it, available for access at the table.

In *Focus*, files look like objects placed on top of the tabletop surface, as there are no borders or other markings on them. When a file is touched, lines appear on it to indicate its selection areas. A file can be *moved* by selecting it inside the main area and dragging it. When moving, files have a realistic momentum so they can be *flicked* around the tabletop. Files can be rotated and resized (in a combined *resize* action) by selecting one of the corners and dragging it. A file can be *flipped* by selecting it within a stippled triangle on its edges, then dragging across to the opposite edge. Once flipped, the filename appears written on the back.

When *Focus* is first launched a broad *start view* shows the first file (alphabetically) in each exported directory of each remote file system in a radial layout. From the start view, file navigation is based on the notion of a *focus file*. Once a user selects a focus file, all other *similar*<sup>1</sup> files across the file systems are loaded and displayed on the tabletop. Retrieved files are presented as a single file-space, regardless of the file system they came from. On focus selections, the initial size of each file reflects its similarity to the focus, with irrelevant files automatically hidden, reducing clutter. Files previously displayed on the tabletop are presented at their last location, providing spatial consistency, and enabling users to create arbitrary spatial grouping (as in [11]).

*Focus* has additional widgets with special functionality. The *History Browser* (shown in the top middle of the tabletop in Figure 2) supports ‘back’ and ‘forward’ operations through focus selections, and its leftmost icon returns to the start view of the file systems. The *Black Hole* (top right of Fig-

<sup>1</sup>based on a comparison of any textual file content and other available meta-data, such as keywords, filename and last modified time.

ure 2) is a temporary “trash can,” but it is also designed to address some of the challenges of tabletop interaction design — objects can be *flicked* to the Black Hole in cases where a user cannot reach it, and it gives visual feedback that an object is about to be hidden (the object shrinks as it moves closer to the centre of the Black Hole). Finally, *Storage Bins*, shown at the bottom of the tabletop in Figure 2, are special folder-like objects with thumbnails of items dragged into them (similar to [13]). A space-filling thumbnail layout accommodates them. Storage Bins can be moved, rotated and resized on the tabletop just like any other interface object. For our evaluation, a text label on each bin indicates the part of the task it is for (e.g. Hotel).

## STUDY DESIGN

The central goal of our study was to gain an understanding of how to provide effective interfaces for file system access when small groups of people need to achieve a collaborative task with their files. We identified two main dimensions to study:

**Collaboration:** a central concern, as this is a new and important possibility for such ubicomp interaction displays. One of the key requirements of desirable file access mechanisms is that they facilitate effective collaboration, where this includes the amount and styles of collaboration and the level of interaction that each user has with others’ files.

**Screen Use:** deals with issues particular to large single display groupware interfaces, notably, clutter and its management, use of table real estate, orientation.

The first dimension of the study is important because tabletops should support small groups of people to collaborate effectively. The second dimension is for our goal of understanding how people use a tabletop with *Explorer*. By contrasting this with *Focus* on aspects that the literature identify as important for tabletop interaction, we can learn which aspects are robust across these very different styles of file system interface. Given the deep underlying differences between *Explorer* and *Focus*, we expected to see different group dynamics and approaches to conducting the tasks. *Explorer* has the advantage that users should be familiar with it from existing computer use. Thus, we expected users to be efficient in using it to access their own file system for a well-organised collection. However, as *Focus* is designed to automatically retrieve similar files from multiple collections, we expected this to lead to different collaboration dynamics, and ultimately, more files being discussed.

## Experimental Procedure

We chose a holiday planning task as it exemplifies the type of collaborative task that people would reasonably do at a tabletop. The study was within-subjects double-cross-over, where participants repeated the same planning task, choosing hotels and tourist activities, for each interface, but with a different holiday destination (either London, England or Florence, Italy). The holiday destination changed so that participants used different files for each repetition of the tasks. The ordering of use of file system interfaces was balanced. The combination of file system interface and holiday destination

was also mixed. The task was designed to involve retrieving and discussing many different files from separate personal file collections, and deciding, with the partner, which tourist attractions to visit and which hotel to stay in. The stages of the study were:

**Preliminary Questionnaire:** This asked about background: age, gender, occupation, typical amount of computer use, past tabletop experience, self-assessed skill in file management, how long they had known their partner.

**Preliminary Task:** Participants were told they would be planning a holiday in London and Florence and relevant information was on their computer. They needed to organise the supplied information, for use later with their partner, at the tabletop. They were asked to select 3 hotels and 3 tourist attractions that seemed most appealing and to store information about them.

**Conditions 1 & 2:** At this stage, participants joined their partner at the tabletop to select 4 tourist attractions, 2 per day (from the 6 on their combined shortlist) and a single hotel (from the 6 on their combined shortlist) based on its location and amenities. Selected files were dragged to an indicated area on the tabletop. For half the participants, this involved *Explorer* first and for the other half, *Focus*. The tasks were repeated with the second condition.

**Post-Experiment Questionnaire:** The *closed* questions had two parts: teamwork aspects (10 questions); and the task and interaction (22 questions). They required a 5-point Likert-scale response for each condition. Five questions asked which interface was: easier to use; easier to understand; faster for performing your work; useful for your work; your preferred one. The *open* questions (asked for each condition) were about: ease of finding all information needed; aspects liked best for the task; aspects disliked most; and aspects they would change for collaboration. Ordering of the questions matched the ordering of interface conditions in the experiment.

**Interview:** The semi-structured interview had 8 questions about how well the interfaces supported collaboration; difficulties finding information needed; conflicts in doing task; satisfaction with the task outcomes; satisfaction with the interfaces; assessment of the quality of outcomes with the two interfaces; effect of orientation and clutter. It was 5–10 minutes long.

We ran the study with pairs of participants. This avoids complex group interaction problems but is realistic for the sorts of planning tasks that people often need to do. Participants were provided with disjoint collections of files related to the two holiday destinations. This was to encourage collaboration — participants needed to rely on their partner’s files to complete the task. These files were images, fact sheets and tourist information (PDF documents) related to a selection of tourist attractions, hotels and general area information. The files were labelled to clearly indicate the attraction or hotel they refer to. Each participant was assigned 91 files, consisting of information for five attractions and five hotels, at least one map, and a travel guide. This number of files was realistic for the planning task — it is plausible that a user will only have a subset of *shareable* files that they wish

to access at the tabletop, making it unnecessary for them to allow collaborative access to the entire file system.

For the *Preliminary Task*, participants were seated at individual computers (at opposite sides of the room to discourage the participants helping each other), and used *Windows Explorer* in its default configuration. The unsorted files were presented in a window on the left side of the screen. Participants were given time to review this information, and organise it into a suitable hierarchy for sharing at the tabletop later. A template file hierarchy, consisting only of empty folders, was provided for participants to organise the files into. However, participants were free to deviate from this structure if they wished (by removing, renaming or adding folders). The file hierarchy was presented on the right side of the screen so that participants could drag the files between windows.

A challenge in designing the study is that file systems are tools for organising *personal* information. If each participant had used their own personal file system, with their own files, the results would have been more difficult to compare. Further, it would have demanded even more time from participants, to make their own sets of files for the task (for example, by collecting and saving information related to the holiday destinations). Our study provided users with a collection of relevant hotel and tourist attraction files in a single folder, that they needed to organise. This file familiarisation stage compensated for the fact that we provided the files, so that we could have a consistent data-set for all experiments. There was no time limit for the initial organisation task, to ensure they had sufficient time to familiarise themselves with the travel files. After viewing and organising the files, participants were asked to choose their favourite three hotels and attractions to access at the tabletop for discussion with their partner.

Both *Condition 1* and *Condition 2* tasks began with a short tutorial, provided by the experimenter to ensure that participants were familiar with the file access tasks required in the experiment. Participants then completed the planning task (choosing four attractions to visit over two days, and one hotel to stay in).

After completing the task using the first condition, participants were asked to sit down at opposite sides of the room, so that they would re-approach the tabletop (positioned in the centre of the room) from opposite sides, and possibly reconsider where to position themselves around the tabletop for the next condition that they would use.

### Hardware and Data Capture

The tabletop for the experiment had a 46-inch LCD touchscreen with a native display resolution of 1920×1080 pixels, using camera-based techniques to sense touch. The screen had a 16:9 (widescreen) aspect ratio, making it wide enough for two people to stand side-by-side. The tabletop was raised to a comfortable standing height in a flat, horizontal configuration. While the tabletop hardware is capable of detecting two touches at a time, *Explorer* cannot support this so all interaction was restricted to single-touch.

The experiment was audio and video recorded from two camera angles, one to mainly capture the participants and the other to capture the tabletop screen. In addition to the experimenter, a note-taker was also present.

### Participants

We recruited 20 participants (11 male, 9 female), in pairs who knew each other beforehand. They came from a range of backgrounds, including university students, researchers, human resources advisors, managers, and a pensioner. They ranged from 18 to 66 years, with an average age of 30. Ten had previously participated in very different tabletop experiments, on the same hardware but none had used a tabletop for file system access with either *Explorer* or *Focus*.

### RESULTS

All pairs of participants were able to complete the tasks satisfactorily. Participants were highly engaged in the task, and took care to plan their holiday so that the attractions and hotel were nearby, even though we did not set any such constraints. We now report analysis of each of our sources of information about collaboration and file access.

### Video Coding Analysis

The experiment videos were coded<sup>2</sup> to gather key features of interaction. Two conditions of the experiments, selected at random, were independently coded by a second investigator to test the accuracy of the primary video coding. There was over 92% consistency, higher still on measures reported in this paper. We now discuss the results in terms of our key evaluation dimensions: collaboration and screen use.

### Collaboration

#### *File Ownership and Interactions with Each Other's Files.*

For each participant, we gathered data for the number of interactions, both with their own and their partner's files. We defined an 'interaction' to be a direct touch and purposeful manipulation of a file or folder. Thus, 'interaction' included primitives such as open, enlarge (*Focus* only), scroll (in the case of a PDF document), and selecting a focus file (*Focus* only). Interactions that occurred to help a partner (for example, passing them an object that they wanted but could not reach) were not included in this analysis. We assumed people would feel a social expectation to help their partner reach things when they are closer to the object, and so we excluded these interactions as the person did not choose to interact with their partner's files without their partner encouraging it. A Wilcoxon signed-rank test indicates that participants interacted significantly more with their partner's files using *Focus*, than with *Explorer* ( $F$  mean = 30.2,  $E$  mean = 12.15,  $z = -3.07$ ,  $p < .05$ ).

With *Explorer*, it was rare that a participant would reach into their partner's files and interact with them. We observed the opposite with *Focus*, where participants freely interacted with all objects on the tabletop, despite many of them belonging to their partner. While there is more gestural interaction involved in using the *Focus* interface (for example, it

<sup>2</sup>using *Anvil* (<http://www.anvil-software.de>)

may be necessary to rotate objects to view them at a particular position at the table), the results still suggest that participants freely performed these gestures on their partner's files without the ownership boundaries observed with *Explorer*.

#### Speed of Task Completion and Number of Files Viewed

In nine out of ten trials, people were faster<sup>3</sup> completing the tasks with *Explorer* than with *Focus*. A Wilcoxon signed-rank test shows participants took significantly longer to complete the tasks using *Focus* ( $F$  mean = 16.7,  $E$  mean = 8.8,  $z = -2.52$ ,  $p < .05$ ). The range of task duration was 4.9–17.7 minutes for *Explorer*, and 11.2–25.9 for *Focus*. We analysed the number of files viewed per minute; there was no significant difference, indicating that people viewed files at the same rate with both interfaces. A Wilcoxon signed-rank test showed that during this time participants viewed significantly more files with *Focus* as part of the collaboration ( $F$  mean = 30.2,  $E$  mean = 12.15,  $z = -2.88$ ,  $p < .05$ ).

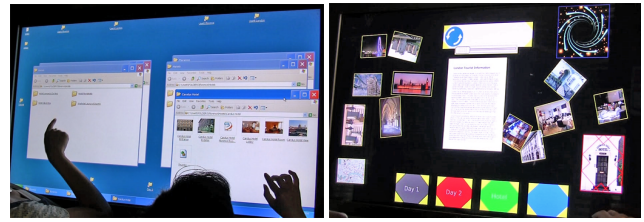
We note that counting the files viewed was not straightforward. For example, in *Explorer*, when a large folder was opened in thumbnail view, it was unclear how many of those thumbnails the participants actually looked at. Similarly, in *Focus*, when many files were scattered around the tabletop it was unclear how many the participants looked at. Thus, we defined a “file view” as follows. In *Explorer*, it was an explicit opening of a file. In *Focus*, it was the action of enlarging a file to view its content in more detail (as users do not explicitly open files in *Focus*). This gives us a comparable indication of the number of files that were specifically looked at by participants in each condition.

#### Serendipitous File Discovery

A striking feature in the *Focus* videos was *serendipitous file access*; this occurred when a participant noticed a file and this caused the conversation to shift. For example, when participants were discussing a tourist attraction, but one noticed a nearby hotel, they shifted the discussion to include this and to take account of these two, hotel and attraction, together in their planning. Another example was when a participant noticed one of their partner's files on the table and thought it looked interesting; they had not requested it explicitly, but it was presented because it was similar to the current focus file. In two cases (Pairs 1 and 9), participants started discussing a hotel only after it was presented as a similar file while they were determining which attractions to visit. In two other cases (Pairs 4 and 8), the conversation shifted as a participant noticed one of their partner's objects that they liked, when it appeared as a similar file. Overall, four out of ten pairs experienced serendipitous file discovery when using *Focus*.

#### Screen Use

The collaborative nature of interaction at an embedded ubiquitous display creates new challenges for managing the screen space, compared with a conventional desktop. Notably, with multiple users, there is the potential to open many files. More-



**Figure 3.** Two examples of the tabletop partitioning schemes observed. Ownership partitioning (left) was used commonly with *Explorer* to keep each other's windows separate. Semantic partitioning (right) was commonly used to group files on the tabletop (attraction files on the left, hotels on the right).

over, deciding to hide a file is an easy decision on a single user desktop but not on a shared display. It may be bad manners to just remove files without consultation and, in a planning session, such discussion may be distracting, perhaps even rude. When the display has limited resolution, this, combined with good manners, may require keeping files quite large. This makes clutter a problem for such displays. So our analysis assessed how the different file access mechanisms affected the layout strategies employed. While there are many techniques for aiding clutter management, these are orthogonal to file access mechanisms and are not part of this work.

To determine how the tabletop workspace was used for each interface, we coded any grouping or partitioning strategies employed by participants. There were two common layout strategies. *Ownership* partitioning occurred where a participant organised their own files in one area, most often directly in front of them, though some also placed their files in a particular area they chose on the tabletop. The other grouping approach was *semantic* partitioning, where objects were grouped together based on their content or function in the task. For example, both participants may put all the hotels of interest in one group. Figure 3 shows an example of ownership partitioning in *Explorer*, and semantic partitioning in *Focus*. With *Explorer*, participants had any windows representing their file hierarchy directly in front of them, occasionally moving a file or folder of interest to the middle of the workspace for their partner to view. By contrast, with *Focus* we commonly saw semantic groupings of files, formed in different areas of the tabletop, regardless of who the files belonged to.

Semantic partitioning was primarily used to manage clutter for both conditions. For example, by grouping all files, based on the part of the task for which they are needed (such as grouping attraction files or hotel files, as in Figure 3 right), participants could see a subset of files needed for the discussion. In some respects, *Focus* encourages the latter behaviour since its retrieval may present a mixture of different types of files, attractions, hotels and maps. So participants who wanted to discuss just hotels may well move the files for them into a pile and the others to another pile. At the same time, semantic organisation points to lower effects of individual ownership.

A Wilcoxon signed-rank test showed ownership partitioning occurred significantly more with *Explorer* than *Focus* ( $F$

<sup>3</sup>note that participants were not put under any time pressure to complete the tasks.





**Figure 4.** The affinity diagram containing qualitative feedback obtained from the post-experiment questionnaire.

mean = .1,  $E$  mean = 3.4,  $z = -1.83$ ,  $p < .05$ ). Semantic partitioning did not occur significantly more with *Focus*. However, this partitioning occurred in nine out of ten trials for *Focus*, and only three out of ten trials<sup>4</sup> for *Explorer*. We observed it 21 times for *Focus*, 17 times for *Explorer*.

### Open Questions

To analyse the open questions from the post-experiment questionnaire, we used an affinity diagram that supported categorisation and visualisation of the main themes emerging from the free responses. The affinity diagram constitutes another rich source of evidence about the qualitative differences between files access using *Explorer* and *Focus*.

The final diagram, shown in Figure 4, was organised into 25 categories — 11 related to *Explorer*, 13 to *Focus*, and 1 for general comments on the tabletop itself. Comments related to more than one category were duplicated and assigned to each. In total, 170 comments were added to the diagram, with 67 for *Explorer* (23 comments stating only “yes” were omitted from the board, but considered in our analysis), 96 for *Focus*, and 7 general tabletop comments.

Clutter was raised as an issue with *Explorer* 13 times, but only twice for *Focus*. Clutter was confirmed as a major issue during the tasks in the post-experiment interview, being more severe for *Explorer*, where the primary cause was windows overlapping, with insufficient support to manage and switch between them.

Another key difference was in the important area of being able to find files needed and to be confident that all the right files had been found. Comments about being able to find all files needed were positive for *Explorer*, with all 20 comments (12 of “yes”) indicating hierarchical access was effective. For the same question about *Focus*, 16 of 20 comments indicated that access was effective (11 of “yes”), but 2 of those also mentioned it was time consuming compared to *Explorer*, and 3 mentioned they were unsure the system retrieved all of the appropriate files. The remaining 4 comments were “no” because they were unsure that the system retrieved all of the appropriate files. In all, 7 of the 20 comments expressed concern about *Focus* not retrieving all the appropriate files.

<sup>4</sup>two of the pairs used *Explorer* as the first condition, and the other pair used *Focus* as the first condition.

Question	Med. (SD) Explorer	Med. (SD) Focus	$z$
My partner’s actions were helpful for solving the task. (F)	4 (.71)	4.5 (.51)	-1.87
I could clearly see what my partner was doing on the table. (F)	4 (1.43)	4 (.75)	-1.68
I could always find what I was looking for in my own files. (E)	4 (1.25)	3.5 (.6)	-2.75
I could always find what I was looking for in my partner’s files. (E)	4 (1.21)	3 (.92)	-1.79

**Table 1.** Significant Likert-scale responses from the post-experiment questionnaire, summarised by median, standard deviation and  $z$ . (F)/(E) indicates the interface receiving the more positive response.

### Closed Questions

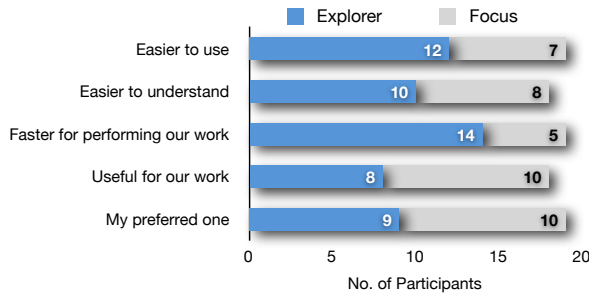
We asked participants to rate on a Likert-scale (range 1–5, with 5 the most positive response) aspects about their experiences using the interfaces, and their experiences working with their partner. Of the 32 aspects tested, a Wilcoxon signed-rank test showed 4 to be significantly different for the two interfaces (with  $p < .05$ ), as shown in Table 1.

The responses indicate that participants thought their partner’s actions were more helpful when completing the task with *Focus*. This follows from the fact that *Focus* searches all users’ file systems when a focus file is selected — and so, the participant’s relevant files were automatically retrieved from their own file system as a side-effect of their partner’s actions. Similarly, participants could more clearly see what their partner was doing on the table with *Focus*. This adds to the finding of clutter management being less effective in *Explorer*, as in the affinity diagram analysis.

The responses consolidate the finding that people felt better able to find files with *Explorer*, both in their own and their partner’s file systems. This matches the affinity diagram comments on the ability to find files. The hierarchically structured file organisation was favoured by two key factors: participants had just organised their files, so were more familiar with their file structure and the files; all participants had almost identical organisation structure. Thus, participants had some familiarity with their partner’s file system, as a side-effect to organising their own, simplifying navigation of the file hierarchy when accessing their partner’s files. Together, the open and closed questions add to the emerging picture that *Explorer* was faster and more predictable but that *Focus* had positive effects for collaboration.

### Interface Preferences and Physical Positioning

We asked participants in the post-experiment questionnaire to indicate which is their preferred interface for four different usage aspects, as well as overall. Figure 5 shows a graph of the results. Responses where no interface was selected (which occurred once), or both were selected (which occurred five times), were excluded from the analysis. The results indicate that 63% of participants found *Explorer* easier to use, 56% found *Explorer* easier to understand, and 74% found *Explorer* faster to use. However, 56% of participants found *Focus* a more useful interface for the collaborative task, and 53% chose it as their preferred interface.



**Figure 5. Interface preference results for four usage aspects, and a preference for the overall preferred interface.**

We also observed participants’ physical positioning around the tabletop. *Focus* is designed for orientation-independence and we took care not to influence participants’ decisions on where to stand at the tabletop. Despite this, in only two trials<sup>5</sup> did participants stand opposite each other when using *Focus*. In four other trials, participants started opposite each other, with one moving to be on the same side as their partner on starting the tasks. Interviews indicate that this was so that people could view objects in the same orientation as their partner, particularly for the map, when planning their attractions and hotel.

## DISCUSSION

We set out to gain an understanding of the important facility of file access in ubicomp environments. We did this by studying and comparing use of two fundamentally different mechanisms for enabling pairs of users to work on a planning task at a tabletop. Our trip planning task is representative of an important class of tasks that ubicomp embedded displays should support — small groups of people meet, potentially on the spur of the moment, to share and collaborate using a large shared workspace, requiring access to the relevant information from their carried and remote storage devices. This multi-user multi-machine ubicomp file access, which differs significantly from desktop file system access, prompted our investigation on how to provide effective support for it. While our study was in the context of a tabletop display, the main results for the deep characteristics of the two mechanisms studied are likely to be equally applicable to other embedded large screen interfaces such as wall displays and, potentially to interfaces based on carried projectors.

Our study made use of several sources of evidence: video analysis, open/closed questionnaires, and interviews. Together, these build a clear picture for understanding the potential of these classes of file access mechanisms. We summarise several of the main results in Table 2. We now discuss them in terms of three themes: collaboration, effective file finding; and use of the screen space.

Our results show interesting relative strengths and weaknesses for both approaches. To understand these, we need to consider the basic underlying characteristics of each and

<sup>5</sup>one pair used the *Focus* condition first, the other used it second.

Description of Analysed Aspect	Explorer	Focus	z
Number of files viewed (mean)	12.15	30.2	-2.88
Interactions with partner’s files (mean)	30.2	12.15	-3.07
Time to complete task (mean minutes)	8.8	16.7	-2.52
Range of times (minutes)	4.9–17.7	11.2–25.9	
Able to find files (# times in free comments)	20	16*	
Serendipitous file discoveries	0	4	
Ownership partitioning (mean)	3.4	.1	-1.83
Semantic partitioning (# groups)	3	9	
Semantic partitioning (# times)	17	21	
Clutter (# times in free comments)	13	2	

**Table 2. Summary of analyses. Aspects with z values are statistically significant ( $p < .05$ ). \* 3 users noted that they were still not certain all had been found; 2 users noted it was slower.**

what our study tells us about these. This is far more important than fine grained details of the particular implementations of these deeper ideas. One key difference between the mechanisms is the *associative* file access mechanism of *Focus* compared with the dominant *hierarchical* mechanism of *Explorer*. *Focus*’ *associative* file access is characterised by a single ‘keystroke’ action of dwelling on a file, to retrieve a collection of similar files potentially from multiple folders crossing the file system hierarchies of all users connected to the tabletop. In the *Explorer* mechanism, the comparable keystroke action opens one folder on one user’s file system, displaying the files that they placed there. The deepest difference between the two is that the *Explorer* mechanism is highly predictable if the user remembers where they placed their files in their folder structure. The *associative* access is less predictable as it performs a multi-faceted search, causing unpredictability on two levels. First, the results include the files of the other users. Second, the results cross the user-created file hierarchy. We wanted to see the effects of these very different ways to access files for the important aspects of collaboration and effective use of a large embedded display. The study clearly indicates that the deepest characteristics of these two file access mechanisms have a strong effect on these key aspects of ubiquitous file access.

## Collaboration

This is a particularly important aspect because the need to support collaboration is one of the key differentiators between existing desktops and the target of our work, embedded large screen tabletop interfaces. The two mechanisms exhibited significant differences in terms of the amount of information viewed, and the level of interaction participants had with each other’s files, as indicated in the first two rows of Table 2. When using *Focus*, participants interacted more readily with *each others files* than with *Explorer*. This contributed to the number of files viewed, with significantly more viewed with *Focus*. We can explain this in terms of the core differences between the two file access mechanisms. With *Focus*, each new focus selection presented participants with a collection of files, some familiar, as they come from their own file-space, some new and potentially intriguing, or



at least novel, because they were from their partner's files. We observed this to encourage exploration of more files overall and it encouraged more discussion. In cases that we have described as serendipitous file access, *Focus* retrieved files that crossed the usual file system hierarchies, for example, encouraging users to shift their focus from just selecting a hotel to considering attractions at the same time. This suggests the potential for richer interaction and a difference in the way that a planning task can be supported. In our case, it happened when it made sense for the files to be thought about in terms of multiple hierarchies. This is something that is not well supported by hierarchical file systems. Broadly, our study indicates that collaboration was richer with *Focus* because it helped participants really consider each others' files more and to consider multiple issues at once.

Our study was constrained to a single session of about an hour. This meant that participants were very familiar with their own files, having just selected and organised them. Over a longer period of time, we predict that the cross-hierarchy retrieval of *Focus* would be even more valuable for the individual user who may well navigate the file system to one relevant directory but may be more likely to forget about others. Our planning task involved just two classes of files and these matched the users tasks of finding a hotel and selecting attractions. For many planning tasks which do not match the existing hierarchy as well, the associative file access mechanism has even greater potential benefits. For example, it retrieves relevant email, just as it retrieves other documents. Our study points to the value of the associative mechanism for richer collaboration on tasks such as planning.

### Ability to Find Files

Clearly, this aspect is at the core of a file access mechanism. As indicated in Table 2, both the timing data and questionnaire comments and responses indicate that *Explorer* was more efficient and participants were more confident that it enabled them to find the files they wanted. We now consider the underlying elements of this finding.

The associative access mechanism is inherently less predictable. This raises the question of how to make such an interface more predictable. A natural starting point is to have information about the matching criteria presented on the tabletop, so users can scrutinise and gain understanding of how the retrieval works. This also opens new possibilities for allowing users to fine-tune the retrieval system to work more efficiently for their information set and task — an important aspect to consider when users with very different file system organisations try to work together, potentially on a planning or other task that had not been predicted when the file system was organised and the files placed into a hierarchy. If an associative mechanism has a place for ubicomp file access, the issues of scrutiny and control of the retrieval system deserve further exploration. This would aim to achieve both greater predictability and efficiency.

Hierarchical access is predictable and efficient if a user needs to locate files that are stored in a known location. Further, users are already familiar with this class of interface and

have a strong existing mental model of how the system works. Despite this, the limitations of hierarchies for file system access have been widely documented (for example, [10]) — information can only be stored and retrieved from a single place. This causes problems for files that cannot be easily categorised, for example, where there are multiple natural hierarchies. Importantly, it is potentially difficult to navigate an unfamiliar hierarchy, such as that of a friend who is not present for the planning activity but is happy to make available the files that they collected for it. This may be a feature of planning that is done when people happen to meet and decide to discuss their plans that also affect others.

Overall, with the existing dominance of hierarchical file systems having a role for the foreseeable future, it seems that ubicomp file access mechanisms should support it too. However, the associative file access mechanism appears to have a complementary place for supporting fast and effective file access, particularly the serendipitous file access. But the issues of predictability and control over the mechanism need to be explored.

### Screen Use

The results summarised in Table 2 indicate that *Explorer* encouraged ownership-based partitioning, where participants kept their files in front of them and away from their partner's files. Typically, participants would have any windows representing their file hierarchy directly in front of them, and would occasionally move a file or folder being discussed into the middle of the workspace for their partner to view. With *Focus*, we commonly saw semantic groupings of files formed in different areas of the display as a technique for managing clutter (for example, grouping all files based on which part of the task they are needed for and moving them out of the way until needed). Also notable, *Focus* appears to have superior performance for management of clutter.

Previous findings [12, 14] suggest that people establish personal working areas in front of them, and show reluctance to interact with objects directly in front of other people. This is reflected in the ownership-based partitioning we observed with *Explorer*. However, we did not observe people creating the same personal file areas with *Focus*. Rather, files were considered a group resource, with loose ownership, and were organised on the tabletop based on their content, rather than who owned them. This led to more efficient use of the tabletop space, as participants spread-out their files to different areas of the tabletop instead of keeping them directly in front of them, as observed significantly more with *Explorer*.

Participants valued the fact that *Focus* preserved their layouts across different focus selections, and they could restore their spatial layout when returning to a previous view. *Explorer* does not allow users to organise files temporarily on the tabletop without making changes to the underlying file collections. For example, to organise all files for a particular subject into a folder, this would involve moving or copying the original files from their original location (possibly from separate file systems), which is undesirable when simply discussing them at the tabletop. With the large display area of

the table surface, the ability to form spatial groupings of information, regardless of the underlying storage locations, is important for this class of ubicomp display [15]. Our results point to the significance of considering file placement on large display groupware, such as in *Focus* which places files before all the users to encourage interaction and spatial organisation of all files on the tabletop.

## CONCLUSIONS

Our study aims to inform the design of ubicomp interfaces that will enable people to bring, browse and collaborate with their file systems — a crucial facility for people to be able to make effective use of their digital stores for ubicomp collaboration. We conclude that the existing hierarchical file access mechanism of *Explorer* has a place, particularly for an individual accessing their own files in a structure they know well. At the same time, our study points to the benefits of an associative multi-machine file access mechanism for improving collaboration, its capacity to overcome the silos of hierarchical file organisation and the ways that its characteristics facilitate greater depth of interaction for tasks such as planning. The *Focus* style of interface supported better management of the large display, encouraging semantic, rather than pure ownership clustering of files, and the improved management of clutter. The study points to the advantages of combining both access techniques, so that either can be used depending on the task. As the only aspect specific to tabletops is the varied orientation of the *Focus* presentation, we would expect these deeper results to apply for other embedded displays, particularly those that are multi-user.

## ACKNOWLEDGEMENTS

We thank our reviewers for their feedback, and our experiment participants for their time. This work is partly funded by the Smart Services CRC, the HxI Initiative and NICTA. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

## REFERENCES

1. Agarawala A. and Balakrishnan R. Keepin' it real: Pushing the desktop metaphor with physics, piles and the pen. In *Proc. CHI 2006*, pp. 1283–1292. ACM Press (2006).
2. Apted T., Kay J., and Quigley A. Tabletop sharing of digital photographs for the elderly. In *Proc. CHI 2006*, pp. 781–790. ACM Press (2006).
3. Bergman O., Beyth-Marom R., Nachmias R., Gradovitch N., and Whittaker S. Improved search engines and navigation preference in personal information management. In *ACM Trans. Inf. Syst.*, 26(4):1–24 (2008).
4. Collins A., Apted T., and Kay J. Tabletop file system access: Associative and hierarchical approaches. In *Proc. TABLETOP 2007*, pp. 113–120. IEEE Comp Soc (2007).
5. Everitt K., Shen C., Ryall K., and Forlines C. MultiSpace: Enabling electronic document micro-mobility in table-centric, multi-device environments. In *Proc. TABLETOP 2006*, pp. 27–34. IEEE Comp Soc (2006).
6. Hartmann B., Morris M.R., and Cassanego A. Reducing clutter on tabletop groupware systems with tangible drawers. In *Adj. Proc. UbiComp 2006* (2006).
7. Hilliges O., Baur D., and Butz A. Photohelix: Browsing, sorting and sharing digital photo collections. In *Proc. TABLETOP 2007*, pp. 87–94. IEEE Comp Soc (2007).
8. Morris M.R., Paepcke A., and Winograd T. TeamSearch: Comparing techniques for co-present collaborative search of digital media. In *Proc. TABLETOP 2006*, pp. 97–104. IEEE Comp Soc (2006).
9. Olsen Jr. D.R., Clement J., and Pace A. Spilling: Expanding hand held interaction to touch table displays. In *Proc. TABLETOP 2007*, pp. 163–170. IEEE Comp Soc (2007).
10. Quan D., Bakshi K., Huynh D., and Karger D.R. User interfaces for supporting multiple categorization. In *Proc. INTERACT 2003*, pp. 228–235. IOS (2003).
11. Robertson G., Czerwinski M., Larson K., Robbins D.C., Thiel D., and van Dantzich M. Data Mountain: Using spatial memory for document management. In *Proc. UIST 1998*, pp. 153–162. ACM Press (1998).
12. Ryall K., Forlines C., Shen C., and Morris M.R. Exploring the effects of group size and table size on interactions with tabletop shared-display groupware. In *Proc. CSCW 2004*, pp. 284–293. ACM Press (2004).
13. Scott S.D., Carpendale M.S.T., and Habelski S. Storage bins: Mobile storage for collaborative tabletop displays. In *IEEE Comp. Graphics and Applications*, 25(4):58–65 (2005).
14. Scott S.D., Carpendale M.S.T., and Inkpen K.M. Territoriality in collaborative tabletop workspaces. In *Proc. CSCW 2004*, pp. 294–303. ACM Press (2004).
15. Scott S.D., Grant K.D., and Mandryk R.L. System guidelines for co-located, collaborative work on a tabletop display. In *Proc. ECSCW 2003*, pp. 159–178. Kluwer Academic Publishers (2003).
16. Shen C., Everitt K., and Ryall K. UbiTable: Impromptu face-to-face collaboration on horizontal interactive surfaces. In *Proc. UbiComp 2003*, pp. 281–288. Springer (2003).
17. Shen C., Lesh N.B., Vernier F., Forlines C., and Frost J. Sharing and building digital group histories. In *Proc. CSCW 2002*, pp. 324–333. ACM Press (2002).
18. Wigdor D., Penn G., Ryall K., Esenther A., and Shen C. Living with a tabletop: Analysis and observations of long term office use of a multi-touch table. In *Proc. TABLETOP 2007*, pp. 60–67. IEEE Comp Soc (2007).