When Information Navigation Divorces File Systems – Database Surface Prototype Results

Rikard Lindell
Mälardalen Real-Time Research Centre (MRTC)
St:a Ursulasv. 2A
722 23 VÄSTERÅS
Sweden
+46(0)21151759
rikard.lindell@mdh.se

Abstract

All PCs have file systems. The design of this most vital part of personal computers has had the consequence that the design of user interfaces, for better or for worse, has become wed to the file system. The file system can be traced in interfaces from today's graphical user interfaces, for which the desktop metaphor helps explain to users how to treat their documents and folders, to yesterday's command line user hostile DOS (Disk Operating System) interfaces.

What if one was to exchange the spine of computer system from a file system to a database? Imagine the unstructured stream of ones and zeros you normally put in a location on a hard drive instead be put in a content aware database. Monolithic applications are divided into components that are put in the database. All the different content files are turned into objects that are put into the database and so forth. This introduces a number of questions that include how to handle: information access; information visualisation; user collaboration; multi modal interface usability; program architecture; scalability over different platforms; to name a few.

This paper presents a design rationale for how to visualise the content of the database. I have chosen a comprehensive approach to the zoom interface paradigm. All content is presented on a flat and infinitely large two-dimensional database surface. There are no windows. The information has only the state: open; there is no closed state for which an application opens the content in a file. Users zoom in and out on the information as the main navigation technique.

The application area first to be investigated was music creativity. I have interviewed musicians, both young female novices and male experts in their 30s. Both groups agreed as to what the problems with current tools are. With the interviews as inspiration I formed the design rationale and constructed a prototype accordingly. Eventually this prototype was evaluated with user studies based on the collaborative evaluation method. The users felt mentally head over heels at first, but thought this dramatically different approach to what they were used to was both usable and amusing.

Introduction

Today's design of the desktop metaphor interfaces found on various platforms is very similar to the original design of the Star system at Xerox PARC in the late 70's. The original design was for office applications and desktop publishing. However the same interface paradigm appears in numerous application areas. The computational resources available to the Star were weak with limited memory and storage space. Since then, the flow and repositories of information that users have to handle has increased by an order of magnitude.

The availability of the Internet created a new arena for people to meet, share information and collaborate. Yet the tools and the interfaces of computers are still unsatisfactory. For instance collaborating in the creation of a content rich document, such as music, animations or movies, are still limited to sending files back an forth attached to emails or on shared file servers. For software development there are version management systems, for instance CVS, that are quite useful for program language skilled people, but these systems are for experts and are not usable for the non program language knowledgeable user.

Many of the most devastating user errors comes from the user not recognising the mode of the system, errors are known as mode-errors [1]. There are other modes such as the dialogue boxes that prevent users from further actions in an application until they has completed the dialogue.

Some advocates one should yield to that it is impossible to create a modeless interface. Even in guidelines such as the Macintosh Human Interface Guidelines [2] that supports modelessness devotes the bigger part of the modelessness section to guide the correct use of modes.

Instead of yielding to modes I believe one should get to the root of the problem. The root in my opinion is the file. Files have two modes, open and closed. In the closed state users have to navigate their files with a file management tool. Only a few attributes, most commonly name and file extension, provides the user with a cue to the file's content. Other attributes might be used; icon, creation date, modification date, preview, and note. In fact application programs, known to users of Macintosh or Windows systems as double-clickable files, are indeed in themselves modes [2].

The functionality or service, in other words the application programs, that extends the computer's system abilities are distributed as files. Most of the time these applications programs work with private formats, known only to the vendor that delivers the service.

It is the users task to appoint the application program to a file type. Even though Macintosh have used file type and file creator identities since 1984, they work only under the assumption that one file goes with only one application program. Some formats have received more acceptance than others, but due to the design of desktop systems, applications are in focus and not the document.

There are many situation where interaction designers used peoples' knowledge in world outside the computer to explain the behaviour of an application program. In the case of the desktop metaphor, peoples' knowledge about desktops, documents and folders were utilised to explain the behaviour of file-trees. On the desktop you were able to put all the files related to current activities, but if you did not put the files for completed projects deeper into the hierarchy you eventually ended up with a mess. The trouble with metaphors are that they might be too literal. The risk in this case is that the computers power would be restrained by real world constraints. When you tidy the mess on your real world desktop, you will later on have trouble recalling where you put things. As with the computer desktop, the reason you had to tidy it was that there was no more room.

The desktop obeys real world constraints such as limited size, but the computer world is virtual world and obeys the rules we make. What happens if we remove the size constraint for the desktop?

Database
It is my conviction that the file-system should be replaced with a database. All contents that users normally locate on a computer's file system volume should be put into this database. The database should also contain all the service components for manipulation of the contents. One way to categorise the information and data types for the service components would be Bernsen's modality taxonomy [3]. A service component should work only with services for a data type.

Today more and more services are database driven, for instance web-news pages and corporate business systems have been using databases. Database technology does also have a number of features that in itself would be very valuable for the backbone of personalised computation powers.

Concurrency
Database concurrency would allow many users to access the same data. This feature enables collaborativity, sharing and communication to be embedded in the system. For instance users would simultaneously be able to work on a document across networks. It is today the responsibility for the application program developer to coerce a collaborative mechanism on top of network protocols and file servers.

Persistence
An immediate consequence of database persistence is the modelessness of the contents; the users do not have to transform data from different states. Contents become immediately persistent at creation time. The user does not have to invoking open, close or save commands. One objection that would be expected to be raised against this design, is that the save action confirms that the user wants to keep the data. However, for this to be truly usable, people actually have to learn about primary and secondary memory of the computer. An explicit save action made sense in the era of floppy disk, today it is obsolete. Very important in my case is that users should feel safe: data representation that supports unlimited amount of undo reduces user anxiety.

Queries

With the enormous storage capacity of the current systems the need for powerful search utilities are immense. Information query mechanisms are ubiquitous in database servers, thus database systems are well suited for information access tools and search tools. However, in users perspective search must preserve the context of the content for which the search condition was satisfied. In other words it is not useful to present the result to the user as a long list of labels. It would also be unfruitful to leave users with a language such as Structured Query Language to control the content of a database.

Zoom Interface Paradigm

The big issue and the one that I had to approach was to find a suitable way to visualise the content of this comprehensive database. With such vast range of different types of objects and information a general visualisation technique had to be found. Turning to the theory of cognitive maps [4] and cognitive collage [5], it seems that spatial semantics would be a usable for the visualisation of data. A cognitive map is a metaphor for the mental model of a persons environment. It is not a precise map, it consists of landmarks and their relations. However most people rely on multiple maps, a cognitive collage, where land marks and their relations can mismatch.

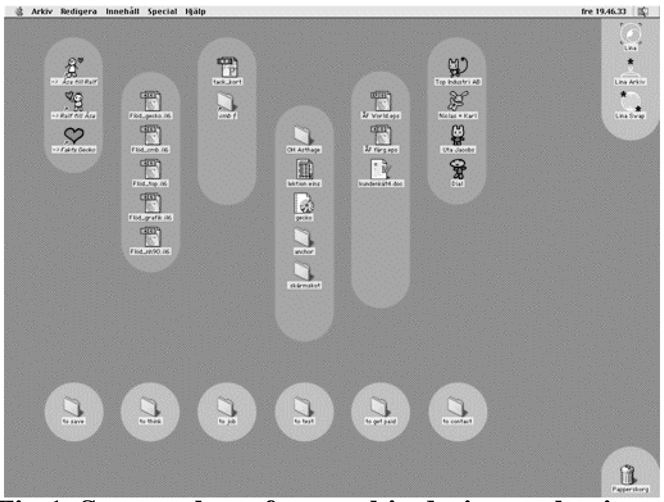


Fig 1. Screen-shot of a graphic designers business management system, based entirely on spatial semantics.

Barreau and Nardi showed that people rely on spatial semantics and context when organising their documents [6]. Dourish et. al. have successfully enhanced file navigation by enabling the users to depend on attribute spatial semantics to locate files in the Presto Document System [7], however the files are still modal and attached to specific application programs. To open a file it need its associated application program.

In user observations I have found support for use of spatial semantics with an ordinary desktop. For instance one user I observed, a graphic designer, had created a project management system with a desktop image serving as a background created a rule context were files and folders were organised manually for each project. See fig 1. As each project progressed the files and folders were moved closer to the finish line. A completed project was moved from the desktop into an archive folder. Contacts with customers and other agencies were also included in this system.

Another fellow at the same office also used spatial semantics, though at the first glance it looked more like chaos see fig 2. The user had no trouble locating information: everything is on the desktop. Also each customer has a region for which files and folders are located. A completed project is put into a folder while the folder is left on the desktop. The user explains: "I may need to reuse some of the stuff for this [the users pointed at a location of the desktop] customer, it may be a logo or a photograph. If I remove it [the folder] from the desktop I forget where I put it".

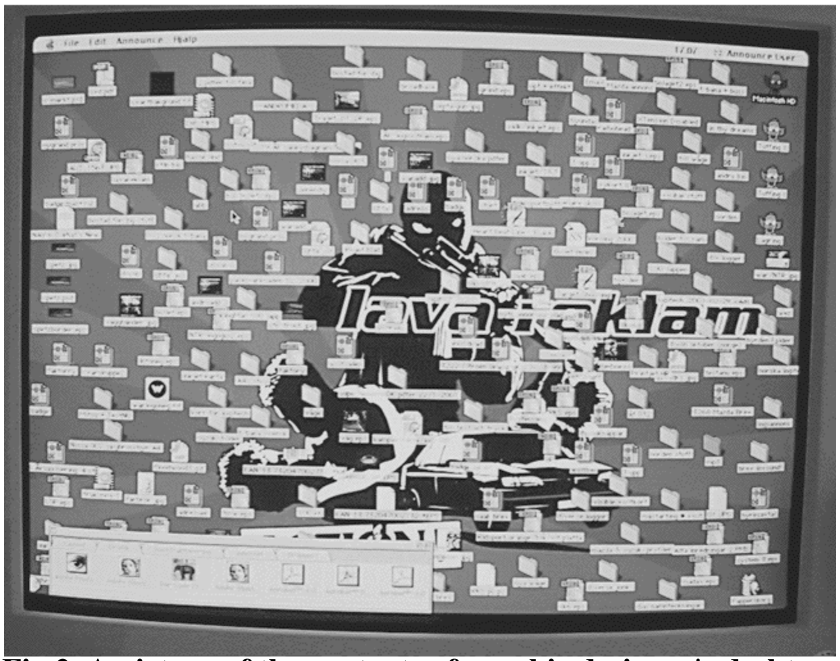


Fig 2. A picture of the contents of graphic designer's desktop. All the information is present, different project are located in different regions.

Also what can be found in current visualisation techniques? Zooming Interface Paradigm Bedersen’s Pad++ [8] looked promising, as did Kay's Squeak [9]. But instead of a local scope of zoom such as in Pad++ or discrete mode switchers and browsing such as in Squeak. I propose a global zoom interface paradigm for the entire database. The database is visualised as a infinitely large two-dimensional surface.

The cognitive psychology aspect of this approach is that the layout of the contents provides spatial semantics. The users know the kind of the information elements from their position. For instance, songs that the users have created themselves could be located in a region to the top right of the database surface, songs that the user ripped from CD:s could be located to the right-hand side, and songs shared with some friend could be on the bottom right-hand side. Note that the entire right area is devoted to music, but different subareas have different attributes to it.

There are no size restrictions to the database surface; hence the information space can grow indefinitely. For instance the user might have created a video that goes with one of the songs. The user puts the video in the region that holds the song, the system makes sure that the surface is grown and that other information content is pushed aside to make room for the video.

An illustration of how users share a part of each database surface is displayed in fig 3. User one's database surface encoded in light grey, the text documents to the left and right are private, but the document in the middle, with the background encoded in white, is shared with user two. User two's database surface is encoded in dark grey, the two text documents to the right are private, but the white encoded document to the left is the exact replicate of user one's document. Actions are echoed to each other to guarantee that the shared space is identical. There is an application area for which this design is already used: games. Multiuser games such as Black & White™ [10] and Galactic Battlegrounds™ [11] are entertaining examples of this approach.

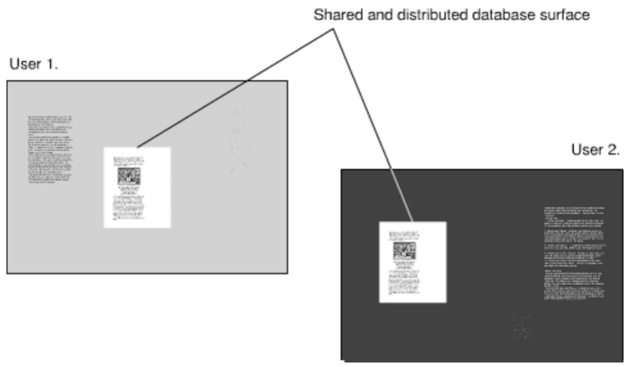


Fig 3. User 1 and user 2 shares the surface in white, whereas the light respectively dark grey regions are private to each user.

To test the idea of using the zoom interface paradigm to visualise the contents of database I constructed a couple of prototypes. In the quest to find better interfaces for computers it is more or less obvious to use a database. Also in my opinion it is apparent that there should be an incremental and interactive search tool included with the design. The idea could not only rely on theory of information visualisation and cognitive psychology, it had to be evaluated by few experiments. I constructed a few prototypes to test some of the aspects. The application area selected for examination was music creativity. There are a number of interesting questions regarding music creativity applications.

1. Music creation has many modalities, mainly temporal and acoustic. But in order to organise sound, this attributes have to be visualised. Also a song can be visualised as an organisation of sound elements.

2. Music creation is often a social activity. People get together to create music, however the current trend of software emulation and the design of desktop computers interfaces for one on one interaction impedes collaboration. How could an interface be designed to reintroduce collaborativity and communication to music creation?

3. Music creativity is a creative process with focus on experience and enjoyment as attributes for usability. The challenge here is to create aesthetically pleasing as well as usable interfaces that encourage creativity.

The first prototype was designed to test how musicians responded to the zoom interface paradigm for visualisation of a song and sound database. The song part displays all the detailed elements of a song.

The First Prototype

The first concept prototype was designed to test how musicians responded to the zoom interface paradigm for visualisation and navigation of a songs and sounds database of music creation tool. The concept prototype was developed with Macromedia Flash® [12]. RSC Technologies WebPAL® [13] tablet hardware was used to run the Flash prototype, the screen size was only 640 (width) times 480 (height) pixels. The hardware was too slow to run fluently, but the small tablet with touch screen was better suited to the users vision of portable music creation tool.

The fig 4 displays the inverted contrast image of the prototype tested by the users. The display shows the most out zoomed state, or the overview state, of the prototype.
The bottom of the fig holds all the song and sounds content. Three songs were visualised: "Beatnic", "Famous", and "Licker". To the bottom left the observant reader may also see that the content; the sound, the tracks, and the score, is visible for each song. The shrunken view of the content served as its icon.

To the bottom right lies the sound library. It is hierarchically organised with a big label for each category. All the sounds in each category is displayed in a shrunken view. Some categories, for instance "drums", have sub-categories.

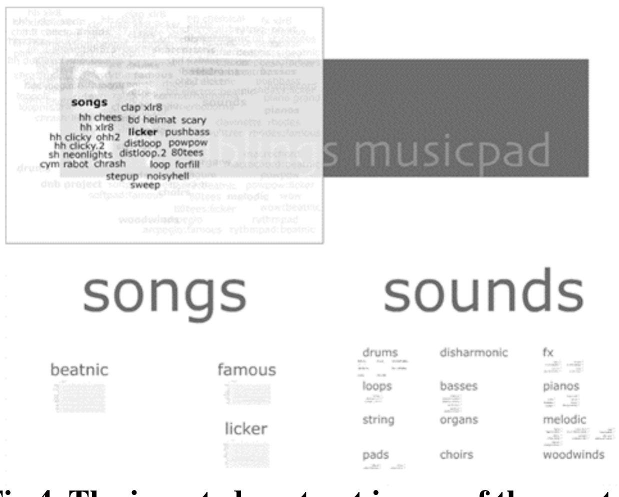


Fig 4. The inverted contrast image of the most out zoomed state for the prototype displayed to the users.

To the top left is a spatial semantic two-dimensional label visualisation tool called "Sens-a-Patch" (SAP) designed by Jonas Löwgren [14]. The initial idea was that the users

needed some kind of navigation tool or site-map to quickly navigate the database surface. This site-map should preserve spatial semantics yet allow labels of all the contents be visualised in limited space. SAP has that quality, however, this design concept was abandoned in future designs for the reason that it brought back many of the file-system's bad traits. Also to rely on label clusters was not consistent with the idea of a big surface for all information. The site-map's behaviour while it was unused was that it was shrunken to a small thumbnail in the top right corner. When it was invoked by the users for navigation it was enlarged on top of the database surface with a degree of transparency as seen in fig 4.
The entire top half of the screen for the overview state has a logo image for tool suite and content in this region of the database surface. At a first glance the image's size seems unproportionally large and a waste of space, however this is not the case due to the infinite size of the database surface.

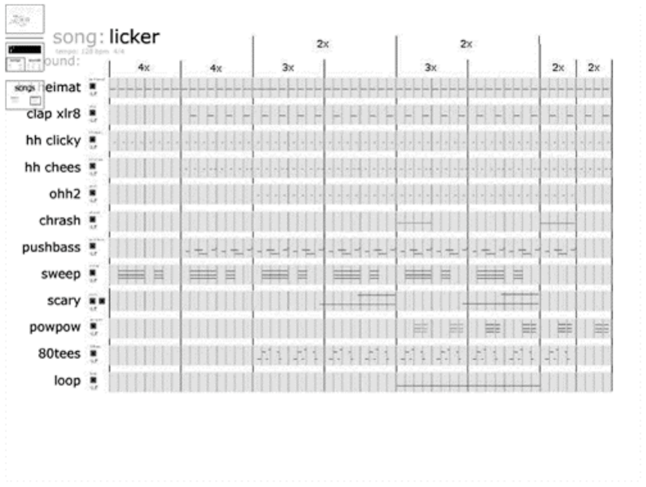


Fig 5. The zoomed state for the score of the song "Licker"

Navigation by Zoom

In this prototype navigation was done by zooming: the user taps on the content to zoom in. In fig 5 the user has tapped on the song "Licker". The database surface is zoomed to display the contents of the song’s score. The time line runs from left to right. To the left there is a list with all the sound labels for each track. Between the label and the track's score is a shrunken view of the sound parameters and source code. The tracks holds the score note values. All note values are displayed at once, users do not have to open an extra window to inspect the notes values for a track.

The top left of the fig 5 has a row of three (3) thumbnail icons. These icons represent the previous zoom state for the database surface. Fig 6a, 6b and 6c, displays these thumbnails in detail. Fig 6a. displays the shrunken site-map, the second thumbnail in fig 6b. and 6c. and the third thumbnail in 6c. displays the previous zoom states. Tapping one of these zooms out from the database surface. Notice also the rectangle into the thumbnail displaying where in the previous zoom state the current view is located.

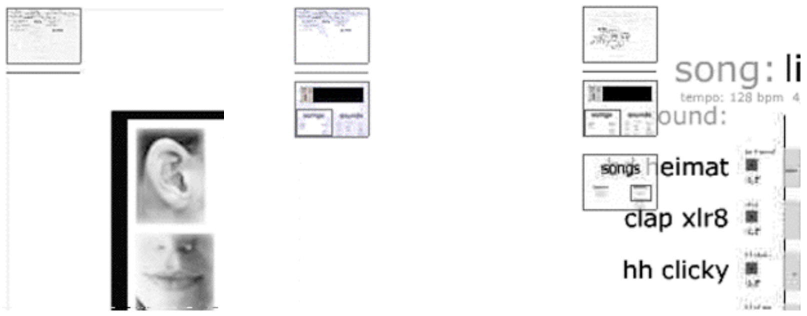


Fig 6a. Fig 6b. Fig 6c.
Each step of the zoom state is displayed as thumbnail icon of the previous screen. Here from the overview state (4a) to the zoom of the score of the song "Licker" (4c).

Prototype Evaluation
The prototype was shown to five (5) subject users who all were skilled music tool users and professional music producers. They were instructed to complete five (5) navigation tasks. The evaluation method used was the collaborative evaluation, a less restricted variant of the think aloud method in which the subject and the evaluator may communicate. The evaluation was debriefed with an interview. A tape recorder and manual was used for protocol.

The tasks were:
1. **Find the song “licker”**. A simple navigation task to find a song. The purpose for the task was that the subject users should get acquainted to zoom navigation, and to the prototype's database surface layout.

2. **Edit the sound “clp xlr8”**. The layout of the database surface was divided into songs and sounds, this and the state in which the subject users were predicted to leave the prototype in from the previous task, coerced the subject users to find the zoom out method; the zoom out thumbnails in the top left region of the display.

3. **Find the sound “distloop”**. To complete this task the subject users did not have to zoom out to the overview state of the database surface.

4. **Read the source code of “distloop”**. The purpose for this task was to see if the subject users recognised shrunken text elements on the database surface such as the source code text of a sound.

5. **Go to the song “famous”**, inspect a representation of the sound “ohh”, go back to the song “famous”. The task was designed to see if the subject users utilised the site-map.

Subject Satisfaction
The users were introduced to the prototype database surface. They were told that they could zoom in and out. However, the zoom out thumbnails were not explained, nor the behaviour of the site-map.

All but one of the subject users completed the tasks without any trouble. One of the subject users completed the tasks with a minimum number of actions. The user that experienced problems saw the images on the screen as a series of steps in a process but not as zoom scales on a surface. He did not perceive the zoom out thumbnails as objects of interaction. He also complained about the lack of an update or save button. He thought the design was too good looking! Supported by the utterance: “Kids will love it!”

All the subject users commented that the approach was different to what they were used to. But after the completion of the first two tasks they grasped the concept and expressed their appreciation. The overall attitude towards the interface was that they enjoyed it, they liked the navigation style, and they were fond of the structure. Those who discovered the site-map liked it as a navigation tool. They appreciated the idea of having all information present at once, in contrast to moving around files from application to application, perhaps changing file format, opening and closing dialogue boxes, and other typical desktop actions.

The feature that the subject users appreciated the most was that all note data was visual in every track. However they thought that this design should be extended to include all sound parameters, allowing control of attributes such as volume, timbre, dynamics, reverberation, etc. Another suggestion was that, in live performance situations, rules and algorithms could generate musical elements. Another bash at the direct manipulation camp.

The Second Prototype

A new prototype was constructed based on a revised design from the evaluation result of the first one. Strong indications were found that zoom interface paradigm was a successful technique for visualising the database surface in the context of a music creation application. However, some of these results were not satisfactory. For instance why did one of subject users not perceive a flat surface, but instead a series of images? Would fluent zoom with smooth transitions clearer visualise the database surface and be more satisfactory than the discrete zoom steps of the first prototype?

It is worth mentioning that the first prototype was strongly constrained by the restrictions of Macromedia Flash capabilities, a trade of I made to rapidly design an acceptable visual appearance. The second prototype was actually two prototypes. I wanted to test fluent zoom with two different approaches to pan. The first of these utilised a grab-and-move metaphor to pan the database surface. Zoom was view centred. The other prototype used a trajectory zoom method that allow both zoom and pan in one action. An elegant method but would it be satisfactory for the subject users? Since the scope of what I needed to establish did not include the complex functionality of music creation the database surface was restricted to contain only a couple of articles from the BBC web news site.

Grab-and-Move

The storyboard in fig 7. illustrates how grab-and-move pan works. The user wants to read the text document to the right. The first step is to move the surface so that the text to be read is in the centre, the user presses down the left mouse button, which is the "grab"-action, then moves the surface to the left by moving the mouse in the same direction, illustrated in the transition from fig 7a to 7b. The user rolls the scroll-wheel away from himself/herself to zoom in on the surface. Zoom is always, both in and out, centred to the screen and unrelated to the cursor's position.

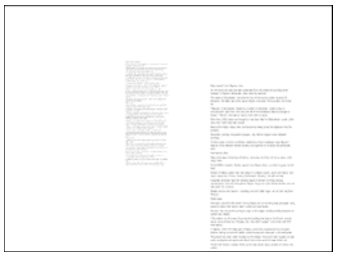


Fig 7a

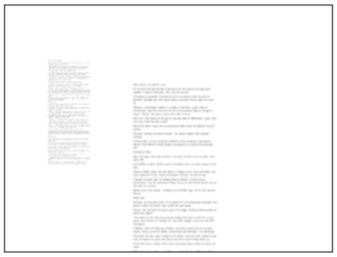
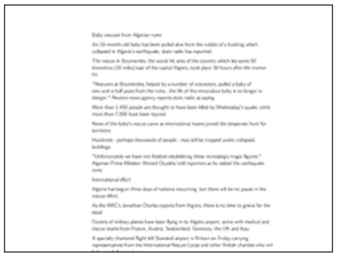


Fig 7b



Baby rescued from Algerian ruins

An 18-month-old baby has been pulled alive from the rubble of a building which collapsed in Algeria's earthquake, state radio has reported.

The rescue in Boumerdes, the worst hit area of the country which lies some 50 kilometres (30 miles) east of the capital Algiers, took place 36 hours after the tremor hit.

"Rescuers at Boumerdes, helped by a number of volunteers, pulled a baby of one-and-a-half years from the ruins...the life of this miraculous baby is no longer in danger," Reuters news agency reports state radio as saying.

More than 1,450 people are thought to have been killed by Wednesday's quake, while more than 7,000 have been injured.

News of the baby's rescue came as international teams joined the desperate hunt for survivors.

Hundreds - perhaps thousands of people - may still be trapped under collapsed buildings.

"Unfortunately we have not finished establishing these increasingly tragic figures," Algerian Prime Minister Ahmed Ouyahia told reporters as he visited the earthquake zone.

International effort

Algeria has begun three days of national mourning, but there will be no pause in the rescue effort.

As the BBC's Jonathan Charles reports from Algiers, there is no time to grieve for the dead.

Dozens of military planes have been flying in to Algiers airport, some with medical and rescue teams from France, Austria, Switzerland, Germany, the UK and Italy.

A specially chartered flight left Stansted airport in Britain on Friday carrying representatives from the International Rescue Corps and other British charities who will

Fig 7c

Transition from fig 7a to 7b shows when the user moves the surface.
Transition from fig 7b to 7c shows when the user zoom in on the text in the centre of the screen. Accomplished by an upward roll on the scroll-wheel.

Trajectory Zoom
Trajectory zoom was used for the other prototype, trajectory zoom enabled both pan and zoom in one action. Fig 8. shows an example for how this is done. Again the user wants to read the right document. To move the content on the database surface down to the left, the user puts the cursor somewhere in the down left area. Then they roll the scroll wheel towards themselves to zoom out.

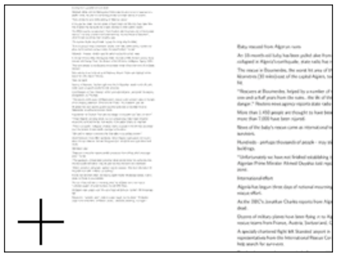


Fig 8a

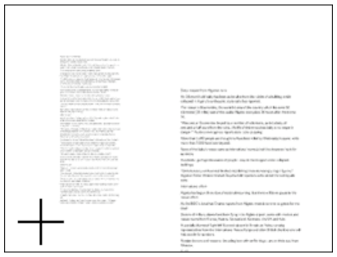


Fig 8b

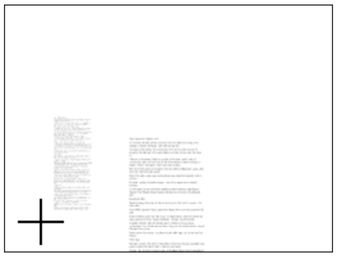


Fig 8c

Pan done by zoom out centred to the trajectory marked with a cross.

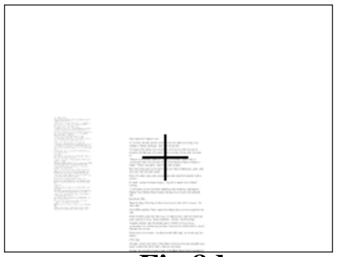


Fig 8d

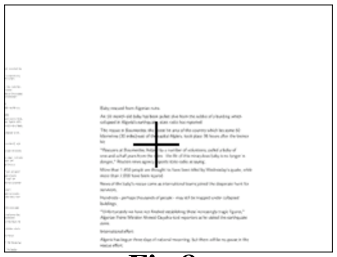
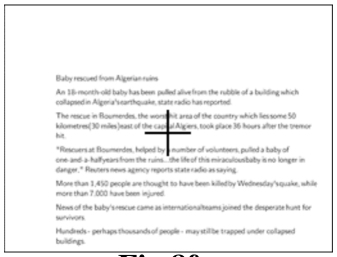


Fig 8e



Baby rescued from Algerian ruins

An 18-month-old baby has been pulled alive from the rubble of a building which collapsed in Algeria's earthquake, state radio has reported.

The rescue in Boumerdes, the worst hit area of the country which lies some 50 kilometres (30 miles) east of the capital Algiers, took place 36 hours after the tremor hit.

"Rescuers at Boumerdes, helped by a number of volunteers, pulled a baby of one-and-a-half years from the ruins...the life of this miraculous baby is no longer in danger," Reuters news agency reports state radio as saying.

More than 1,450 people are thought to have been killed by Wednesday's quake, while more than 7,000 have been injured.

News of the baby's rescue came as international teams joined the desperate hunt for survivors.

Hundreds - perhaps thousands of people - may still be trapped under collapsed buildings.

Fig 8f

Document of interest in focus accomplished by zoom.

Fig 8a-c. displays how the database surface shrinks and centres around the orthogonal trajectory from the view plane to the cursor's position on the database surface (marked in fig 8. with a cross). The user now has an overview of the document, but in order to read the document they need to zoom in on it. In fig 8d. users moved the cursor, and thus the trajectory, aiming to where they wants to read. In 8d-8f. the zoom magnification is displayed, where fig 8f. displays the readable view.

Prototype Evaluation

Evaluation was done by four (4) subject users. One of these subject users was a female graphic designer, the others were computer science students; two female and one male. They were given a written introduction to the test in which the different zoom techniques were explained. The test was set up in two sessions, the first tested the grab-and-move prototype, and the second tested the trajectory zoom prototype. In each session the tasks was to write down the answers to four questions about the articles on the database surface. Users were observed while performing the test. A video camera and manual notes were used for protocol. I wanted to know the subject users opinion of these interfaces, hence, a debrief interview summarised the evaluation.

Subject Satisfaction

All the subject users agree to that this interface felt very different from what they were used to, except for one subject user who had implemented a zoom function as an exercise in a computer graphic course. All subject users did navigate the database surface without any difficulties, except for one subject user who had to put great effort in handling trajectory zoom.

One of the subject users had been using a graphic package that made her perform the zoom actions, using the grab-and-move prototype, with impressive precision. First she zoomed out to get an overview of the database surface. Then she moved the content to its proper position. Eventually she zoomed in on the document in one swift swoop, all other subject users had to do this actions in many repeated steps; first move a little and then zoom a little.

All the subject users easily mapped zoom direction to the scroll-wheel's roll directions of the mouse. They preferred the grab-and-move method for pan: they experienced it: "like moving around a big paper". One user claimed that he got a better overview when using the trajectory zoom. Two subject users made sure that the text had a broad margin to the right, this margin was used as a bar which was grabbed each time a users wanted to pan, they did not want to "touch" the text. Only one subject user gained interest in the trajectory zoom technique: she would have liked to continue to use and train for the it. Her comment was that she intuitively felt trajectory zoom to be more powerful than view centred zoom.

Conclusions

Two prototyped were created and tested. The first one to show if a database surface navigated by zoom interface paradigm could replace a file system. The selected application area for this prototype was music creation.

The other prototype was built to see if fluent zoom yielded better results than discrete zoom and to investigate what methods for pan were satisfactory to the subject users. The evaluations were base on qualitative user observation methods and cooperative evaluation.

The evaluation gave strong indications that the subject users enjoyed the experience of the database surface approach. They explicitly expressed satisfaction with the visual availability of the entire database content. Fluent zoom was more satisfactory than discrete zoom. Separated actions for zoom and pan was preferred by the users, thus a pan technique such as the grab-and-move metaphor should be used in the local scope, however trajectory zoom could not be rejected as navigation method for the database surface in the global scope.

The second prototype should be extended with an overview panel to indicate the user's current visual locus. The result from the debriefing interviews shows that users should be able to engage multiple view of the database surface.

My conclusion is that the basic idea of using a database surface instead of a file system, was supported by qualitative user evaluations presented in this paper.

Future works

The next step will be to implement the results from these evaluations in a prototype for music creation. Multiple views and overview panels will be included. A simple overview panel such as the zoom out thumbnails of the first prototype would further improve the design, it would provide a visual cue to the content that is not in locus of the current view. Also the user requested multiple parameter track, and thus the flattening of the database surface will be included. This prototype will be built with emphasis to live music creativity, collaboration, and concurrency. A database will be at its core, allowing collaboration and music creation across the Internet.

References

[1]. Donald A. Norman, "The Psychology of Everyday Things", Basic Books, New York, 1988

[2]. Apple Computer. "Macintosh Human Interface Guidelines". Addison-Wesley, 1992.

[3]. Niels Ole Bernsen, Multimodality in Language and Speech Systems – From Therory to Design Support Tool, Chapter to appear in Granström, B. (Ed.): Multimodality in Language and Speech Systems. Dordrecht: Kluwer Academic Publishers 2001

[4] Roger M. Downs and David Stea "Image and Environments" chapter 1 "Cognitive Maps and Spatial Behavior", ISBN 0-202-10058-8 Aldine Publishing Company 1973

[5] Barbera Tversky. "Cognitive Maps, Cognitive Collage and Spatial Mental Models", Proceeding of the European Conference COSIT 1993, Springer-Verlag

[6] D. Barreau and B. Nardi, "Finding and Reminding: File Organization from the Desktop", SIGCHI Bulletin, 27(3), July 1995

[7] Dourish P. Edwards W. K. LaMarca A. Salisbury M., "Using Properties for Uniform Interaction in the Presto Document System" ACM Computer-Human Interaction (UIST)'99.

[8] Benjamin B. Bederson and James D. Hollan, Pad++: A Zooming Graphical Interface Widget for Tk, Proceedings of the 1994 TCL/TK Workshop, 73-84.
[9] www.squeak.com
[10] Black & White is a trademark of Lionhead Studios Ltd
[11] Galactic Battlegrounds is a trademark of Lucasarts Ltd
[12] Macromedia Flash is a registered trademark of Macromedia Inc
[13] WebPAL is a registered trademark of RSC Technology AB
[14] Jonas Löwgren, Sens-A-Patch: Interactiva Visualization of Label Spaces In Banissi, E. et al. (eds.) Proc. Fifth Int. Conf. Information Visualization (IV2001), pp. 7-12. Los Alamitos, CA: IEEE Computer Society