Towards a Task Aware Operating System User Interface

Jorge Cunha de Sequeira Amaral

A Thesis submitted for the degree of Mestre em Engenharia Informática (Master in Informatics Engineering), under the supervision of Prof. João Falcão e Cunha, Associated Professor at Faculdade de Engenharia da Universidade do Porto

Faculdade de Engenharia da Universidade do Porto, April 2006

Para a minha família e para a Liliana.

Acknowledgments

I would like to thank Professor João Falcão e Cunha for his interest on this work. His help and opinions were indispensable to the final results. Thanks also to Professor Dave Roberts for his insights.

Many thanks go to all test users for their patience on the interviews and tests, and also for their excellent suggestions.

I also can't forget everybody that, over the last years, shared with me their computer-related problems and took the time to discuss with me possible solutions and ideas. I hope this work will eventually help them.

Special thanks go to my family and to my girlfriend Liliana for all their patience and support.

Without all of these people, this work would not have been possible.

Abstract

Current user interaction with computers is handled, in great extent, by the user interface of the operating system. This mainly follows the traditional desktop metaphor, a representation of a physical desktop, with corresponding files and folders. This model gives the user access to hierarchical structures of folders, files and applications. Each application can also offer the user other hierarchical structures, like bookmarked links in web browsers, or folders in email applications. These and other structures are the basis of most tasks executed by the user with the computer.

Also, in most modern operating system user interfaces, the desktop, folders, files and applications have a direct relationship to the file system. Therefore, the desktop and folders are directories in the file system, and files or applications are files in the file system.

Although this has been, in many ways, an improvement over previous command line based interfaces, the profusion of distinct computer uses and applications led to a proliferation of objects that need to be involved in any one task. This situation requires from the user excessive self discipline and time consuming activities of keeping the desktop, folders and other structures clean and usable. It is quite usual for a specific task to require the use of several files, applications and communication objects, usually scattered all over different structures in the system.

This work proposes a new approach to interaction based on the task metaphor. The user works on an environment that is similar to a Web application and where all required resources for a determinate task are associated with it. In this way, multiple files, directories, applications, hyperlinks and annotations can be found grouped around a task. This permits easy navigation, access and improved workflow for the user. Tasks and subtasks can be added to the system to form a hierarchical structure, allowing the user to have a representation of its own task model, using a straightforward user interface.

Resumo

Presentemente, a maior parte da interacção com computadores é feita através da interface do sistema operativo. Esta interface utiliza normalmente a tradicional metáfora de ambiente de trabalho, com os seus correspondentes ficheiros e pastas. Este modelo dá ao utilizador acesso a estruturas hierárquicas de pastas, ficheiros e aplicações. Cada aplicação pode também oferecer ao utilizador outras estruturas hierárquicas, como os 'favoritos' num browser, ou pastas dentro de uma aplicação de email. Estas e outras estruturas são a base para a maior parte das tarefas executadas com o computador. Na maior parte das interfaces de sistemas operativos modernos, o ambiente de trabalho, pastas, ficheiros e aplicações têm uma relação directa com o sistema de ficheiros. Assim, o ambiente de trabalho e as pastas são directorias do sistema de ficheiros, e os ficheiros e aplicações são ficheiros do sistema de ficheiros.

Embora esta situação seja, de várias formas, uma melhoria em relação às interfaces baseadas em linha de comando, o crescimento dos possíveis usos de computadores e correspondentes aplicações resultou num aumento dos objectos envolvidos na utilização normal de um computador. É bastante comum serem necessários, para uma determinada tarefa, vários ficheiros, aplicações e objectos de comunicação, normalmente espalhados por diferentes estruturas no sistema.

O presente trabalho propõe uma nova abordagem à interacção baseada numa metáfora de tarefas. O utilizador trabalha num ambiente que se assemelha a uma aplicação Web e onde todos os recursos necessários para uma determinada tarefa estão associados à mesma. Desta forma, vários ficheiros, directorias, aplicações, hyperlinks e anotações são apresentadas associadas a uma tarefa. Este modelo oferece ao utilizador uma melhor navegação e acesso simplificado à informação. As tarefas podem ser adicionadas ao sistema na forma de uma estrutura hierárquica, usando uma interface concisa, o que permite ao utilizador obter uma representação do seu modelo de tarefas.

Contents

List of Figures

Chapter 1

Introduction

Over the years, the author has been in close contact with computer users, be it in the workplace or in personal use situations. Watching users struggle with their operating system user interfaces, complaining about the complexity of actions needed to accomplish their tasks, was the main reason for this work.

The desktop metaphor, as used by most operating system graphical user interfaces, can be seen as responsible for many of these problems. Recent changes in the way users approach computer work are putting a strain on this kind of graphical user interface. The complexity of tasks is reflected on the number of files used, the number of different applications and the various information structures involved in the navigation of the user interface.

Different users approach the desktop metaphor in different ways, and therefore various distinct workflows can be identified. Some users follow the original desktop metaphor workflow, by placing various folders in the desktop, naming them, and building folder structures inside them which hold their files. These users will navigate their folders to find the needed files and will open them directly, while the operating system identifies and launches the corresponding application. Other users will choose an application-centric view of the user interface, first deciding the application they wish to work with and opening or creating a file within this application, and placing it somewhere in the file system.

When an interface presents no clear model and workflow for the user, and offers various alternatives for achieving the same goal, some confusion is inevitable, as the user has to decide what is the best way to achieve its organization objectives, normally without knowing all the possibilities available. It is also common for users to vary their workflow in different situations, which can also have a negative effect on the usability of the system.

The Internet also had a big impact on computer use, adding new information structures: email folders and bookmarks. As common users come to rely more and more on Web information and email communication, the complexity of tasks has also risen. It is common for a determinate task, for example writing a school essay or office memo, to incorporate information from various Web site or intranet pages and from emails exchanged between colleagues.

Current operating system user interfaces force separation between all these structures and information by being too closely restricted by file system and application dependencies.

Seeing some users embrace corporate intranets and Web applications with relative ease, and earing them praise these systems mainly because of their consideration for the tasks they wanted to do, inspired the form in which the user interface proposed in this work has been built.

Mainly due to the direct connection between the file system and the folder structures, current operating system user interfaces do not offer much in the way of annotation facilities to the user. Disorientation while performing tasks is easy to occur, especially when the same file or folder name is used in various places in the file system or when important files for this task are scattered around the file system.

The lack of simple ways for the user to document computer usage is a factor that makes life difficult for the common user. Due to the number and size of the different information structures, users forget what made them name a file, what made them place a file in a determinate point in a structure and how the different structures are related in terms of the tasks the computer is used for.

The reasons above identify both the problem and the motivation that is the basis of this work. To validate them, user interviews were conducted and later user tests of the proposed system prototype were made. These interviews and tests did not aim to prove that the proposed solution is better than current operating system user interfaces, but only to validate hypothesis and gather ideas. The selected users were a sounding board for the problems and the ideas presented.

1.1 Listening to users

To validate both the problems and the solution presented, a group of six users was studied. Due to the low number of users available, they were chosen to

be average computer users[47], with no specialized computer knowledge, that use computers in their work or at home on a regular basis.

Interviews were conducted with users, where information about their computer use, with focus on how they organized their work, was compiled. It was then possible to identify their daily use problems.

The proposed user interface was then tested with the same users. In these tests they were asked to use the system as if it was their personal or workplace computer, allowing both for a view on how users react to this model of user interface, and for the identification of opinions, difficulties and ideas for further developments. These tests were recorded and later reviewed.

Throughout this work, user reactions are included where applicable, in context with the current topic being presented.

Peeking into the user's computer The interviews gave a very interesting view of the different ways users approach current operating system user interfaces, how they choose a file centric or application centric way of working and the reasons for their individual folder structure.

Four of the interviewed users use mainly a file-centric approach, one uses mainly an application-centric approach and one use indiscriminately both approaches. Five users populate the desktop, one chooses to have the desktop clean. Of these five, all use shortcuts to applications, three use file and folder shortcuts and one places all his files into folders directly on the desktop.

Identifying the problems All of the users interviewed described situations of lost files, folders and not understanding parts of their folder structure some months after building it.

When asked about the organization of their work or home computers, two users classified their systems as organized, three as not organized and one as only partly organized (he only organized his work area, not his ludic area of the system). As for feeling in control of the information, two clearly though not, the other four said they felt in control, but sometimes didn't know where to find a file or folder.

None of the users felt their computer structure reflected a real office or desktop, and thought of folders, files and the desktop as just one way of organizing information. Three of them responded as never even realizing the connection between the computer desktop and real table desktop or an office environment.

Three of the users admitted to stopping a few seconds to think on approaching the computer to execute a determinate task, as a way to adapt their thought to the actions they would need to do in the computer. All of

them said, however, that while completing one task, they had to navigate the different information structures of the user interface to search for files, websites and applications, effectively slowing down their work.

It was also very interesting to confirm that all of these users make extensive use of notes, either on paper or on text files in the computer. Uses of these annotation ranged from to-do lists for current tasks, identifying currently used files or folders and describing contents of certain files.

Seeking solutions Only one of the users interviewed was reticent to a change in the operating system user interface, but said if it was proved to him that it would help with his current problems, he would try it.

As for the task model, and the possibility of associating files, folders, applications, links and annotations to each task, all users were very interested and felt it would help them a great deal in day to day use. Four of them felt it would be excellent to be able to annotate their structure and files with longer descriptions and to-do lists. The other two said this was important, and that they would use it, but it could not entirely replace real pen and paper for scribbling and taking quick notes.

1.2 Contribution

The main contribution of this work is a task aware user interface model, with the task being the main focus of the user attention, and around which all other elements gather. It is possible, in this way, to build a hierarchical task structure that mirrors the user mental model, and associate to each task files, folders, applications, hyperlinks and annotations. This model is further detailed in chapter 4, "The task aware user interface proposal".

As a way to demonstrate and test the model, a prototype was developed, which runs in any web browser. All essential functionalities of the model were included, in such a way that the prototype could be tested. A detailed description of this prototype, as well as user reactions and test results, can be viewed in chapter 5, "The prototype".

1.3 Overview of this work

The present work aims to make the case for a change in operating system user interfaces, using a task based approach to interaction. The current state of operating systems is presented and the proposed interface is explained.

In chapter 2, “Motivation”, a series of reasons for the importance of this work are presented.

In chapter 3, “History of the Graphical User Interface”, the evolution of the user interface is detailed, from Paul Otlet’s and Vannevar Bush’s pioneering ideas (section 3.1) to present day operating systems like Windows XP and MacOS X (section 3.6) and Web applications (section 3.7).

In chapter 4, “The task aware user interface proposal”, a user interface model based on the user’s tasks is presented. Theoretical and technical choices are discussed (section 4.1), and the relationship between the proposed user interface and the underlying operating system is defined (section 4.3).

In chapter 5, “The prototype”, a prototype developed with base in this user interface model is described, and user reactions observed while testing are presented. Technical choices made in the implementation of the prototype are described (section 5.2) and an example of use is presented (section 5.9). User interviews and tests are described (section 5.10) and the results are presented (section 5.11).

In chapter 6, “Future developments”, some possibilities for future work in this interface are shown, both based on user ideas or in features that were not possible to develop for the present work.

Finally, in chapter 7, “Conclusion”, a review of results and lessons learned is presented.

Chapter 2

Motivation

Most common computer users encounter obstacles to achieving their tasks on a daily basis. It is common for these users to attribute these obstacles to their own ignorance of the system, and not to think that the problem is on the system itself. Due to the complexity of current operating system graphical user interfaces, a still intimate relationship between the user interface and the underlying file system, and other problems connected to the evolution of computer use, work has to be done to develop more intuitive and usable operating system user interfaces.

2.1 Complexity of current GUIs

With genesis in Xerox's 1981 Star[18], the current operating system graphical user interfaces (Microsoft Windows, Apple Mac OS, UNIX's Gnome and KDE) are now pushing the limits of Xerox's, Apple's and Microsoft's initial UI versions, by adding new elements and functionalities to deal with the evolution of computer use. Initially based on a relatively pure desktop metaphor[17] that organized files into folders that were accessible through icons in the desktop, these UIs were regularly updated to cater for big changes in the day to day use of computers.

Multimedia, entertainment and the Internet all played an enormous hand in changing the computer from an instrument of work, used to write letters, balance bank accounts and make inventories of book collections, into a hub for the 'digital lifestyle'. Now the computer is our photo studio, movie studio and music studio; it is our home cinema and our home stereo system; and perhaps more important of all, it is our communication and information center as it is now connected to hundreds of millions of other computers throughout the world.

This change drove modifications to operating system user interfaces mostly through the evolution of the file and folder browser to accommodate multimedia files and the integration of various applications like multimedia readers and web browsers into the operating system. Also, with the rising number of applications installed in each computer, application launchers were also added to save the user from navigation through folders to find the application's executable file.

These modifications have, however, heavily diluted the original desktop metaphor and failed to build a coherent model to substitute it. And above all they are not designed to respond to the underlying change in user's view and approach to computers.

2.2 Relevance of the Desktop metaphor

The Xerox Star used direct analogies with real-world objects, such as files, folders and file drawers, in-baskets and out-baskets[21]. This is the desktop metaphor in its purest sense, as the icons are directly related to the corresponding physical objects. Files always contain data that can be worked by the user as any file in an office.

To the Star user, the desktop, the files and their folder structure were a direct representation of a well known model: the real office. The Star was intended to be an office automation system, and as such, catered only to the office user's necessities. The system was preloaded with a fixed application set that was automatically associated with data files, such as documents, business graphics, tables, databases and electronic mail[21].

Over the years, the computer has substituted most of the real objects that were present on the office desktop, to the point where files, folders, file drawers, in-baskets and out-baskets are only present in our offices as icons in the computer desktop.

Younger computer users will seldom use a real file or a real file drawer, as the computerized office is for them the 'real' office. Moreover, computers have reached well beyond the office, and have now their place in homes, schoolrooms, rucksacks and briefcases everywhere.

As proven by rising use of the Web and intranets, users are receptive and feel empowered by interfaces that are mainly driven by information and its structure, where they do not have to wade through analogies that are not quite right or are not right at all, as is the case in modern operating system UIs.

2.3 Reflection of the user's mental model

When we use a computer, it is with a goal in mind. Normally there is no direct representation of this goal in the user interface; what we get is a desktop with application launchers, scattered files, folders, application icons, and links. Associations have to be remembered by the user to help him navigate folders, application launchers and bookmarks, so that work may begin. Work is, however, rarely accomplished with a single application or in a single folder, and our information sources are usually scattered around the folder structure or even all over the Web or on an intranet.

The strain on the user when he tries to accomplish his goals should be reduced by approximating the user interface to the user's mental model[49] and their own goals.

2.4 Connecting the files with other structures

Operating systems have been forced to include application launchers into their user interfaces due to the rising number of applications installed in personal computers. These structures, possibly built automatically upon installation of each application, are separate from the files and folders structure, forming a parallel entity that has to be learned, navigated and built on its own. Web browser's bookmarks are also an example of a parallel structure that brings up the same problems as the application launcher, and adds a new one: it is accessible only when the Web browser is running.

It is time consuming to manage all these structures and it is cumbersome to navigate them. One flexible structure should be able to accommodate all this information.

2.5 Logging, annotation and comment

How many times do we question ourselves about the content of a folder? How many times do names of bookmark groups mean nothing? What applications are supposed to be 'Utilities', and what are 'Accessories'? When we loose the meaning of our own words, we need something else to help us identify information. The ability to log the use of the computer, to annotate structures and comment on data is an invaluable help, with implications not only to the use of the computer but also to our own mind, as it alleviates it of information.

Corporate intranets, wikis, blogs and other community driven systems have proved their value in helping communication between people; having

logging, annotation and commenting integrated with the operating system will help the user communicate with himself and organize his own work.

2.6 User reaction

As was noted in section 1.1, almost all of the questions raised here struck a chord with the users interviewed.

One user said this about the organization of his office computer: "... I think it is well organized. It would be better if there was an office standard. For me, it works, as I defined the organization and work well with it. I don't know if my work colleagues think the same, if they have to use my information they might not be able to find anything!" The same user, talking about his own personal computer, said the following: "It started well organized... but it is always moving into chaos. When it reaches a certain point, I have to make an effort to reorganize it.'

Another user thought: "... sometimes I don't know what is what... some files and folders have the same name, and I don't know what is in them. I have to open them to see.", "... I think my computer now needs a good cleaning! I have to do that once in a while..."

One other user had to use notes to keep track of the files in her computer: "It's somewhat disorganized. I have lots of information in my computer, and I have to keep track of what I'm working on with paper notes. If I don't work on something for two or three weeks, I don't remember where the important files are."

When asked about the relationship between the desktop user interface model and reality, one user replied: "I never really though about it. I think naming these objects desktop, folder and file in the computer helps to have familiarity with them, but I do not think about them like real world objects."

Another user thought: "I never think about real folders or files. It's just a way to organize information. The names might help in the beginning, to get an idea of what you can do with them. In the end, they are just ways to organize my information."

Talking about navigating various structures in the user interface, one user said: "I use many applications at the same time, files from different places in the hard drive and sometimes websites. It is cumbersome to search for all those things. Applications are in the start menu, files in the folders and my links are on the browser!"

As for taking notes while using the computer, all the interviewed users said they reached for the help of either pen and paper or created files just to add notes. "Always... lots of notes!", said one user.

Another user said: "I usually write things like, 'this is done', 'I have to do this tomorrow...' in a notebook I keep by the computer. Other times I write in my Word documents what I still have to do."

"I use a file to keep track of my work", another user said, "I always start by taking notes on this file, and them I mark some parts of it complete when I do the work."

2.7 Overview

As was shown here, there are many things wrong with the current state of operating system user interfaces. How did we get to this position? There are many reasons, as the evolution of the user interface was not straightforward, but the next chapter will try to shed some light into this intricate story.

Chapter 3

History of the Graphical User Interface

The first advancements in the design of user interfaces were done with the intention of aiding the knowledge worker to have simpler access to information, to speed up his work, and were viewed as knowledge and intellectual capacity enhancers.

We still follow this objective, and many along the years have tackled this problem. The history presented here tries to give a better understanding off how different individuals and companies contributed, and the resulting evolution of the computer user interface.

3.1 First developments

Even before the digital computer, Paul Otlet and Vannevar Bush worked to define a way to aid intellectual work. Using current technologies (index cards and microfilm) they presented machines that organized information and aided in viewing and creating this information.

Later, computer scientists Ivan Sutherland and Douglas Engelbart built systems that used digital computers to help the user perform tasks and augment its intellectual capabilities.

3.1.1 Paul Otlet

Paul Otlet (figure 3.1) was a Belgian lawyer born in 1868 who turned bibliographer and Utopian internationalist. He is considered one of the founders of bibliography and information science[2].

Figure 3.1: Paul Otlet

His Universal Decimal Classification system is the first (of few) full implementations of a faceted classification system, expanding the Dewey Decimal system to go beyond cataloging the books themselves, unearthing the substance, sources and conclusions that are inside - the facets of each book. He wrote the "Traité de documentation" and "Monde: Essai d'universalism", writings dedicated to the collection and organization of the world's knowledge[2].

His contribution to the computer User Interface is a description of a new kind of scholar's workstation (figure 3.2). Consisting of a moving desk shaped like a wheel, powered by a network of hinged spokes beneath a series of moving surfaces, it would permit users to search, read and write on a mechanical database of millions of 3"x5" index cards[3].

This workstation would also permit the user to annotate relationships between documents, which he called 'links'. Otlet also imagined a way to access this database through an electric telescope that used the telephone line, projecting an image of the document on a screen[3].

Paul Otlet's vision was to build a great web of human knowledge. His work was, however, stopped as Belgium was occupied by Nazi Germany, and much of his life's work was destroyed. He died some months before the end of the war[3].

This work has obvious connections with the modern World Wide Web, with searchable documents connected by links, but is also a blueprint for a personal computer user interface, as it permitted not only to read information, but also to write and organize the user's own work.

3.1.2 Vannevar Bush's memex

Vannevar Bush (figure 3.3), born in 1890, was an American engineer, inventor and politician, known mainly for his political role in the development of the atomic bomb and the idea of the memex[5].

Bush described the 'memex', a 'memory extender' device which pre-

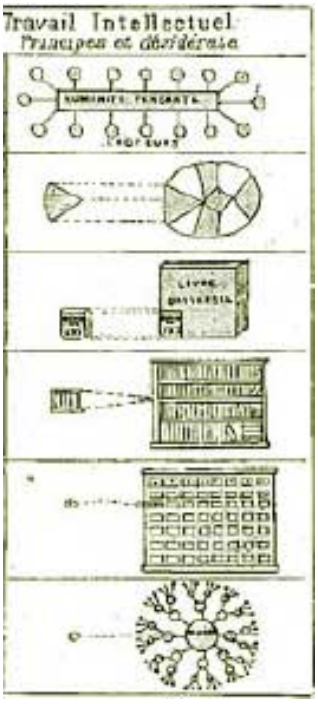

Figure 3.2: Paul Otlet's scholar's workstation

Figure 3.3: Vannevar Bush

sented books and films from an electronically linked library, following cross-references from one work to another. These links were not, however, hyper-links as they were only described by Bush as links from a whole document, not from a single word, sentence or picture[6].

The memex was envisioned by Bush as a desk containing a combination of electromechanical controls and microfilm cameras and readers. Microfilm was contained inside the desk, and the user could add or remove microfilm reels[6].

Bush included in the memex the possibility of adding new information. Using a touch-sensitive translucent screen or a camera, the user could create microfilm and generate new entries in the memex. When the user added these new entries, he indexed them and added them to his personal code book, which he would use to follow his annotations and generated entries[6].

As Otlet's scholar workstation, Bush's memex is mainly seen as one of the seeds of the World Wide Web, but can also be considered as a description of a personal computer and the interface by which the user controls it.

3.1.3 Ivan Sutherland's Sketchpad

Ivan Sutherland (figure 3.4), born in 1968, was the inventor of the Sketchpad (figure 3.5), an ancestor of modern CAD programs and a major development on computer graphics. The Sketchpad was an influence on Douglas Engelbart's On-Line System and the development of the graphical user interface[8].

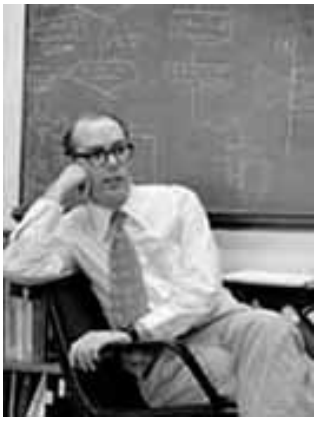

Figure 3.4: Ivan Sutherland

The Sketchpad used a graphical user interface, through an x-y point plotter display, a light pen and buttons for actions like 'draw' and 'move'. This revolution in user interface design was described by Sutherland:

Figure 3.5: The Sketchpad

"Heretofore, most interaction between man and computers has been slowed down by the need to reduce all communication to written statements that can be typed; in the past, we have been writing letters to rather than conferring with our computers. For many types of communication, such as describing the shape of a mechanical part or the connections of an electrical circuit, typed statements can prove cumbersome. The Sketchpad system, by eliminating typed statements (except for legends) in favor of line drawings, opens up a new area of man-machine communication."[10]

Sutherland went on to to create the world's first virtual reality system with head mounted displays and currently works for Sun Microsystems[8].

3.1.4 Douglas Engelbart's On-Line System

Douglas Engelbart (figure 3.6) was born in 1925 and worked in the Philippines as a naval radio technician during World War II. He was inspired by Vannevar Bush's "As We May Think" article and after the war he studied and worked with the objective of using technology to improve the way people worked[11]. Referring to Bush's article, Engelbart writes: "In six and a half pages crammed full of well-based speculations, Bush proceeds to outline enough plausible artifact and methodology developments to make a very convincing case for the augmentation of the individual intellectual worker."[13]

Engelbart's philosophy was that the state of current technology controls the ability to manipulate information, and therefore also controls the ability to develop new technologies. His objective was to create computer-based technologies for direct manipulation of information, and to improve individual and group processes for knowledge work[11]. In his own words:

"By 'augmenting human intellect' we mean increasing the capability of a man to approach a complex problem situation, to gain comprehension to suit his particular needs, and to derive solutions to problems. Increased capability in this respect is taken to mean a mixture of the following: more-rapid com-

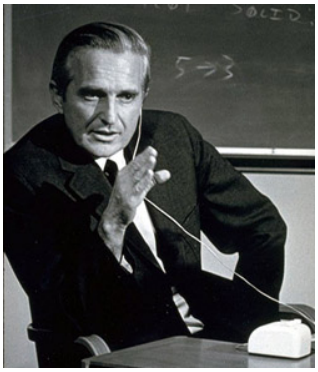

Figure 3.6: Douglas Engelbart

prehension, better comprehension, the possibility of gaining a useful degree of comprehension in a situation that previously was too complex, speedier solutions, better solutions, and the possibility of finding solutions to problems that before seemed insoluble. And by 'complex situations' we include the professional problems of diplomats, executives, social scientists, life scientists, physical scientists, attorneys, designers - whether the problem situation exists for twenty minutes or twenty years. We do not speak of isolated clever tricks that help in particular situations. We refer to a way of life in an integrated domain where hunches, cut-and-try, intangibles, and the human 'feel for a situation' usefully co-exist with powerful concepts, streamlined terminology and notation, sophisticated methods, and high-powered electronic aids."[13]

He and his team invented the computer mouse, and developed hypertext, networked computers and early graphical user interfaces[11].

Engelbart's On-Line System (figure 3.7) was a computer collaboration system developed at the Augmentation Research Center, part of the Stanford Research Institute. It was the first to use hypertext links, the mouse, screen windowing and information organized by relevance. Using a time-sharing computer, it supported 16 workstations composed of a raster-scan monitor, a three button mouse and a device called the chord keyset[12].

'The Journal' was one of the On-Line System's most revolutionary features. It was a hypertext-based groupware program, that is a predecessor of modern collaborative document creation server software like, for example, wikis[12].

The On-Line System was not, however, easy to learn. It used program modes, was based on a strict hierarchical structure, did not have a point-and-click interface and forced the user to learn difficult mnemonic codes[12].

Figure 3.7: The On-Line System

After the end of the On-Line System project and of the Augmentation Research Center, many of his researchers left for the Xerox PARC. With a different view from Engelbart's many of his younger colleagues saw the personal computer as the future, as opposed to time-sharing computers, and started work on computers that used the mouse and developed the modern Graphical User Interface.

3.2 Xerox

Xerox's PARC labs was the birthplace of the first personal computers, the desktop metaphor, and the modern Graphical User Interface.

Although they were revolutionary, the Xerox Alto and Star did not have the commercial impact of other personal computers of the time. They are very important not because of generalized use but because of their impact on the next generation of personal computers.

3.2.1 Xerox Alto

The Alto (figure 3.8) was a $32,000 personal computer that consisted of four parts: the graphics display, the keyboard, the mouse and the disk storage and processor box. Every one of these items was designed to be placed on a desk or tabletop, except the disk storage and processor box, which was the size of a small refrigerator and was best placed on the floor[15].

Graphically, the Alto used a display turned sideways, like a page of paper, that was able to show a 808 by 606 pixel image in black and white. Each pixel had only one bit of memory associated to it, so it could only be black or white. Shades of gray were possible in areas by the use of pixel 'textures'[15].

The mouse had three buttons, which were called red, yellow and blue, although they were all physically black. It worked by detecting motion of

Figure 3.8: The Xerox Alto

the ball bearings located on the bottom and transmitting this motion to the Alto[15].

As for the user interface (figure 3.9), it implemented many of the features of the modern Graphical User Interface, such as the division of the screen in multiple windows, selections and actions controlled via mouse clicks and gestures, and what you see is what you get (WYSIWYG) editing. To illustrate the impact of these features, here is a transcribed excerpt of a 1981 article from Byte magazine:

"A file may be deleted simply by touching the file name with the cursor, then touching the Delete spot on the screen with the cursor. As the cursor enters a new window, it may change shape, perhaps appearing as an arrow in one window and a paintbrush in another."[15]

When referring to the 'Draw' graphics software (figure 3.10), the same article states:

"Curves can be drawn by moving the cursor directly, or by selecting several points and allowing Draw to mathematically fit a curve to those points. (...) Since this is very similar to the techniques used by artists and calligraphers, quite a bit of artistic expression is possible. An object can be duplicated, rotated, stretched, or shrunk, by means of a small set of commands and mouse gestures."[15]

The Xerox Alto was never sold in the marketplace, but several thousand were made and used by various universities. It paved the way for the Xerox Star, released as a commercial product.

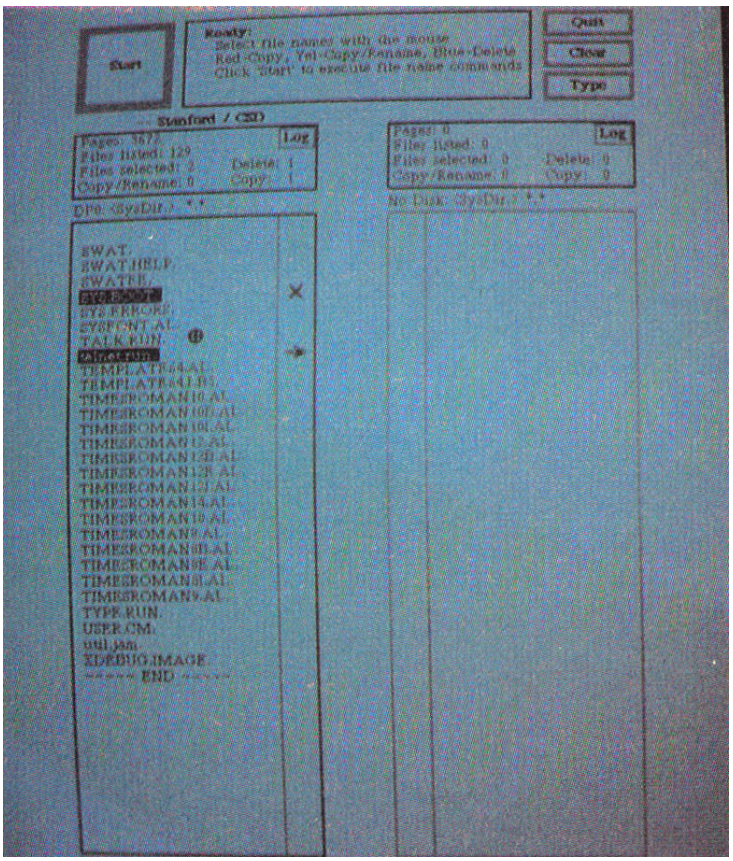

Figure 3.9: The Alto's user interface

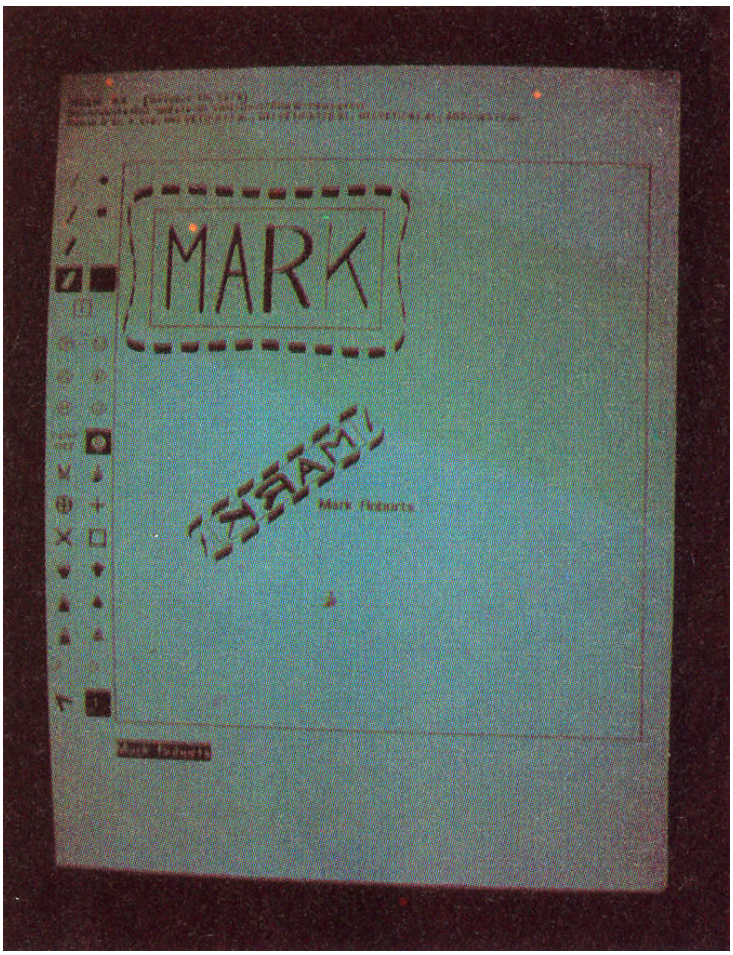

Figure 3.10: The Alto's 'Draw' graphics software

3.2.2 Xerox 8010 (Star)

The Star (figure 3.11) followed the work done on the Alto and used many of the same features. It had, however, a major addition to the user interface: the physical office metaphor. It is important to follow and understand the rationale behind this innovation.

Figure 3.11: The Xerox Star

The Star was developed in a way that was rare at the time: it started from the users and their conceptual model. Jonathan Seybold wrote:

"Most system design efforts start with hardware specifications, follow this with a set of functional specifications for the software, then try to figure out a logical user interface and command structure. The Star project started the other way around: the paramount concern was to define a conceptual model of how the user would relate to the system. Hardware and software followed from this."[20]

The methodology used to design the Star gave much importance to the user interface. The designers found that work done on user interfaces at the time was very ad-hoc and proposed to adopt a more rigorous approach. They defined a design step that was omitted at the time: task analysis. Their methodology report reads:

"The idea behind this phase of design is to build up a new task environment for the user, in which he can work to accomplish the same goals as before, surrounded now by a different set of objects, and employing new methods."[22]

Work was therefore done to understand the user's tasks and create an appropriate graphical user interface (figure 3.12). The Star had a clear design objective and a clear user base:

"Star was designed as an office automation system. The idea was that professionals in a business or organization would have workstations on their desks and would use them to produce, retrieve, distribute, and organize documentation, presentations, memos, and reports. All of the workstations in an organization would be connected via Ethernet and would share access to file servers, printers, etc." [21]

With focus on users that wanted to get work done and who were not interested in computers, the Star had a fixed set of applications that were adapted to user's tasks. Users didn't have to worry about installing applications or even about starting the right application for a determinate task or file. Operating system, software and applications were not visible, leaving the users focused on their work.

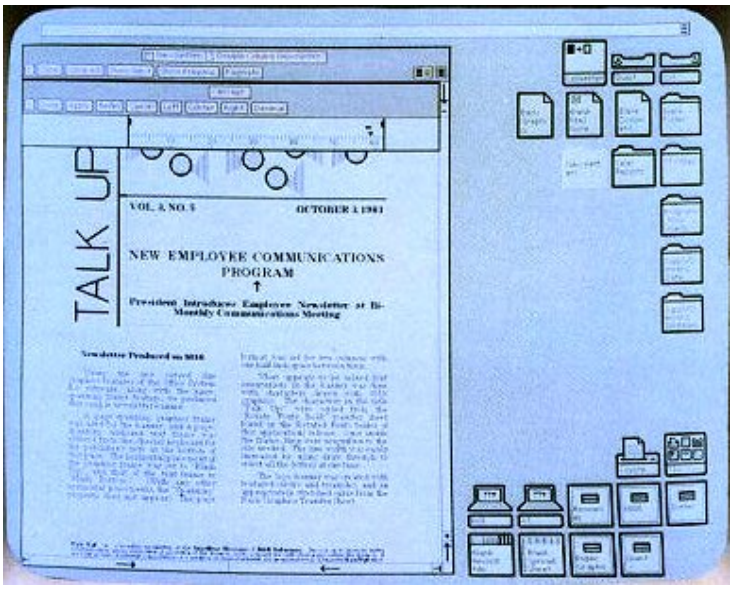

Figure 3.12: The Xerox Star's user interface

The physical office (or desktop) metaphor was, therefore, an answer to the problem: how can we make the use of computers a better experience for business professionals, who are interested in accomplishing known tasks in a more productive way. Here is the Star designers' description of the physical office metaphor:

"Every user's initial view of Star is the Desktop, which resembles the top of an office desk, together with surrounding furniture and equipment. It represents your working environment, where your current projects and accessible resources reside. On the screen are displayed pictures of familiar

office objects, such as documents, folders, file drawers, in-baskets, and out-baskets. These objects are displayed as small pictures or icons, ..."

"You can open an icon to deal with what it represents. This enables you to read documents, inspect the contents of folders and file drawers, see what mail you have received, etc. When opened, an icon expands into a larger form called a window, which displays the icon's contents. Windows are the principal mechanism for displaying and manipulating information."[17]

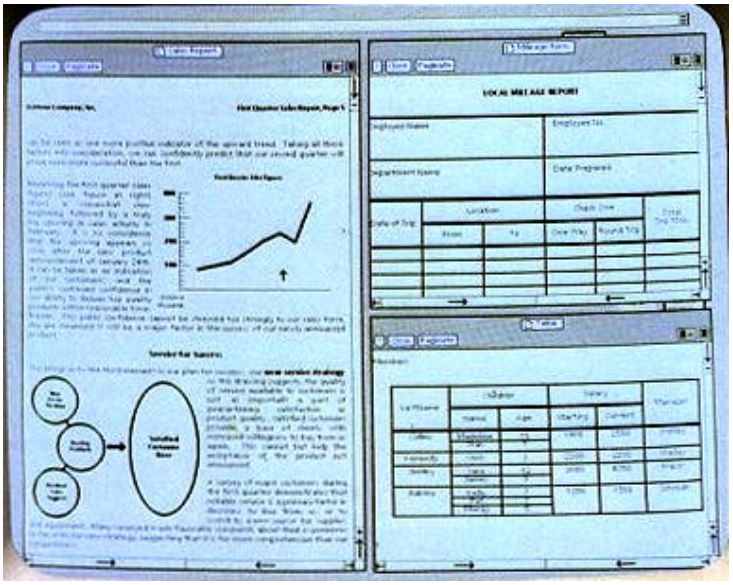

Figure 3.13: Multiple types of document opened in the Star

This metaphor was to influence the design of most of the following graphical user interfaces, despite the Star's commercial failure: only about 25,000 were sold.

The importance of the physical office metaphor at this point in the development of graphical user interfaces is major, although it is important to study the way it was used and altered in other systems, like the Apple Lisa and Macintosh or Microsoft Windows.

3.3 Apple and Microsoft

After the success of the Apple II and the failure of the Apple III, two distinct projects would decide the future of Apple: the Lisa and the Macintosh. Both projects were being worked on at roughly the same time, had different teams and different objectives, and resulted in similar graphical user interfaces.

There are however very important differences between the two, which will be discussed below.

Apple was very influenced by the work done at Xerox, by way of various visits to the PARC, possible because Xerox had bought minority shares in Apple and permitted access.

Microsoft had been very successful with its partnership with IBM, as its MS-DOS operating system was included with every IBM PC sold. With pressure from graphical operating systems like the MacOS for the Apple Macintosh, and even software that run on top of MS-DOS like DESQview and GEM that offered windows and graphical user interfaces, Microsoft developed Windows.

3.3.1 The Apple Lisa

Apple's Lisa (figure 3.14) had very similar objectives to the Xerox Star. It was designed to be used in the office by secretaries, managers and professionals, and its objective was to aid these users to be more efficient in office tasks. Engineering and marketing at Apple made ease-of-use the main goal for the Lisa. Lisa's user interface would have to be easy to manipulate, intuitive and friendly.[25]

Figure 3.14: The Apple Lisa

The evolution of Lisa's user interface is described in the Interactions article 'Inventing the Lisa User Interface'[25], written by members of Lisa's development team. Their work started as a user interface with almost no graphic capabilities that made use of soft function keys and cursor keys and ended four years later with the desktop model, which was a possibility originally rejected during the development process. Apple's developers considered the 12 inch display too small to display documents and icons at the same

time, and also considered that locating documents in nested folders was too complex. Mimicking the office filing systems was also considered to be just a rehashing of a system that already had many problems.[25]

The final user interface (figure 3.15) for the Lisa was, nevertheless, desktop metaphor based. It was similar to the Xerox Smalltalk (an internal Xerox development system) and Xerox Star's user interface but introduced important innovations like drop-down menus, the trash can, drag-and-drop, desktop menu and the clipboard.[26]

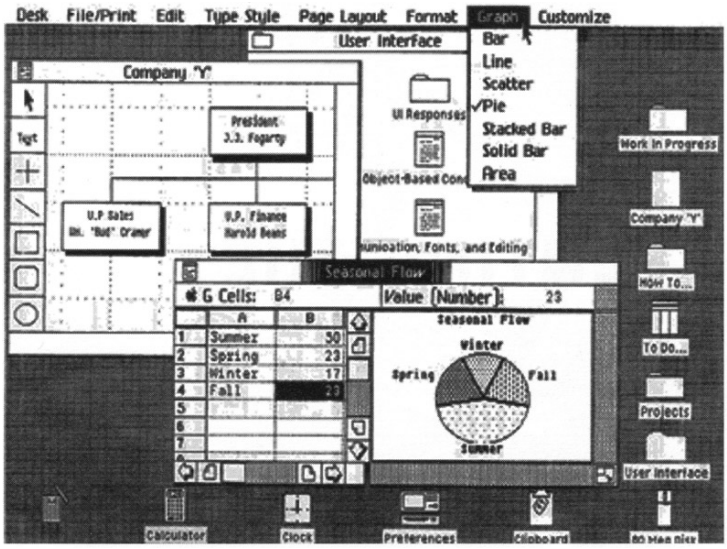

Figure 3.15: Lisa's user interface

Although graphically similar to the Macintosh and Windows user interface (and therefore almost all of the modern operating system's graphical user interfaces), there are very important differences: a document centric model, persistent edits, save and revert, and context saving.[26]

Document centric model Lisa's user interface was based on a document centric model: documents were represented by an icon or a window and resided in a determinate place, a hard disk or diskette; applications, processes and disk files were hidden from the user.[26]

There were no 'Open...' or 'New' commands in any application's 'File' menu. Work on a document started by opening it by selecting its icon. New documents were created from stationary pads. When documents were opened, the document icon was replaced by a shadow, creating the perception

that icons and windows were different representations of the same object, the document, and not separate entities.[26]

There was also no 'Quit' command in applications. Lisa managed applications, starting or ending them depending on the needs of the documents that were being opened or closed. There were no windows related to running applications if there were no opened document that used that application. Also, all document windows were independent, and selecting one document window did not bring forward all document windows of the associated application.[26]

Lisa icons were not individual disk files. Some icons, like folders, had no related disk files and others, like applications, had many. Icon number was reduced and the user's confusion was minimized, as Lisa never revealed temporary files, application files or system files.[26]

Persistent edits, save and revert While using Lisa, all edits of a document were persistent. The user could turn off the computer (or the Lisa could crash) and nothing would be lost. There were two version of a document: a saved version and a working version. The working version became the saved version if the user expressly saved the document or when he put away the document. There was a 'Revert' command that allowed to bring back the saved version and discard the working version.[26]

Context saving When the Lisa powered off, the work context was preserved. On boot, all window positions, icon positions and opened documents were the same has on shutdown. The same happened when a disk was ejected and reinserted: all positions and documents that belonged to the disk were preserved and restored.[26]

3.3.2 The Apple Macintosh

The Apple Lisa was not a successful computer. It was very expensive, being about twice the price of a normally equipped IBM PC. The Macintosh (figure 3.16), a much less expensive computer was soon after released. This would be the sales hit that Apple needed.[24]

The Macintosh was the first big selling personal computer with a graphical user interface, and as such it brought the graphical user interface to the mainstream. This graphical user interface was not, however, a descendant from the Lisa's. It was similar in appearance, but had a major difference: it wasn't a document centric system. Its desktop metaphor was, in result of this, not as clear as in the Lisa or the Xerox Star.

Figure 3.16: The Apple Macintosh

In the Macintosh user interface (figure 3.17), icons represent files on disk, folders are disk directories and the user can navigate the file structures freely. Applications are executed by clicking on the application icon located in the file structure. To start work on a new document you have to find the corresponding application and execute it this way. Clicking on existing files will, however, open the right application if the system recognizes the file.

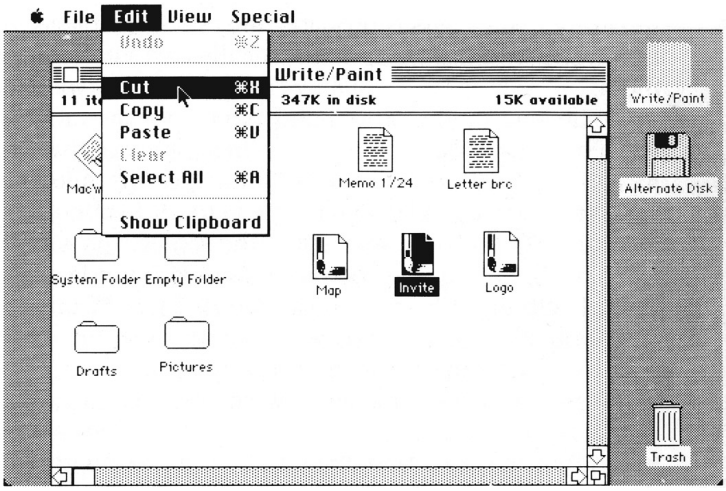

Figure 3.17: The Macintosh's user interface

It is important to note that the Macintosh could only execute one application at a time (the Lisa could run multiple applications) and didn't have a hard disk. To run an application the user normally had to insert the disk that held the application's executable file.

If we compare this to the Lisa or even the Xerox Star, we can see a definite change on the model of interaction. It can be argued that the Macintosh does not implement a desktop (or physical office) metaphor, but uses similar graphic elements to present, in a friendlier way, the computer's file structure. In essence, the model of interaction has more in common with the command line interface of MS-DOS and the Apple II than with the document centric model of the Star and the Lisa.

3.3.3 Microsoft Windows

With the success of the Apple Macintosh, and with the appearance of the DESQview and GEM graphical user interfaces, Microsoft started work on Windows, a graphical user interface that ran on top of MS-DOS. Windows 1.0 (figure 3.18) and Windows 2.0 were notably inferior to other GUIs and Microsoft didn't make a real impact and until 1990, with the release of Windows 3.0 (figure 3.19).

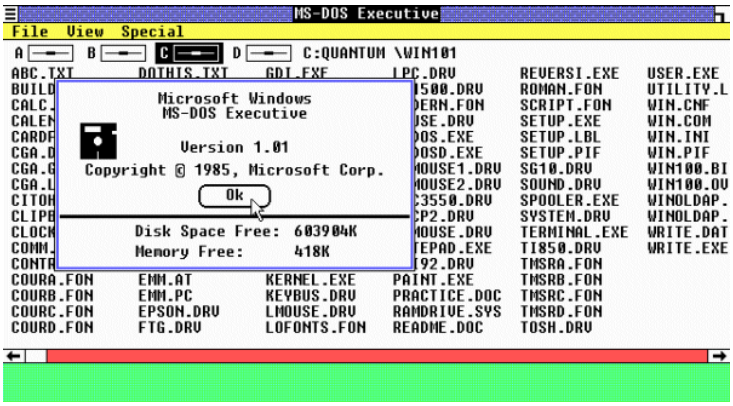

Figure 3.18: Microsoft Windows 1.0

On Windows release version 2.03 Apple sued Microsoft over a previous agreement related to the user interface of Windows 1.0. Apple lost the case because of a technicality and later signed a private settlement with Microsoft.

Microsoft Windows 3.0 is very similar in use to the Mac OS, and only by Windows 95 were there additions in the user interface, such as the start menu and the taskbar. The taskbar was a response to the rise in computer processing power that permitted many applications to run at the same time, and the start menu was introduced as a way to organize and give faster access to the rising number of applications installed in each computer.

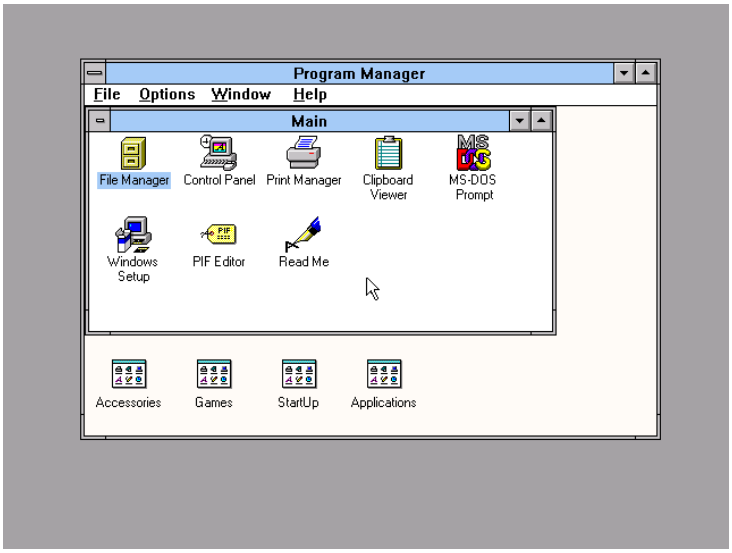

Figure 3.19: Microsoft Windows 3.1

3.4 Sun, NeXT, IBM, and Be Inc.

After the Macintosh and Windows, many other companies developed graphical user interfaces for their systems. We will focus here on Sun's NeWS, NeXT's NeXTSTEP, IBM's OS/2 and Be Inc's BeOS. Although all use the file centric model of the Apple Macintosh, each has unique features.

3.4.1 Sun NeWS

The Network extensible Window System was a windowing system design for client-server use. It permitted (like X Windows) to use a graphical user interface on a client terminal connected via network to a server that effectively received, processed and replied to user actions.[33]

NeWS used a modified Postscript (normally used in printers) to run in cooperative multitasking mode, and was programmed using a complete Object Oriented programming style.[33]

NeWS was not very successful, mainly due to competition from X Windows, that, unlike NeWS, had a freely distributed source code under an MIT License.[33]

3.4.2 NeXTSTEP

NeXTSTEP was the operating system for the NeXT computer (figure 3.20), a product of NeXT Computer Inc. The company was founded by Steve Jobs, co-founder of Apple, after he was forced to leave Apple in 1985.

Figure 3.20: The NeXT Computer

NeXTSTEP was a very advanced operating system for the time: it was object-oriented, multitasking, was based on UNIX and used Postscript as its windowing engine.[34]

As for the graphical user interface (figure 3.21), it introduced two new elements, the Dock and the Shelf. The dock was a placeholder for frequently used programs, each represented by an icon. Icons could be added or removed from the dock by the user.[35] The Shelf was used to hold commonly used files, directories and also programs. It could be used to hold references to files during file system operations.[35]

The NeXT computer was used mainly in Universities, and it was using a NeXT that Tim Berners-Lee built the first web browser, WorldWideWeb.

NeXT Computer Inc. was latter bought by Apple and the current Macintosh operating systems, MacOS X, has many things in common with NeXTSTEP.

3.4.3 IBM OS/2

IBM's OS/2 (figure 3.22) was an operating system developed by both IBM and Microsoft up to OS/2 version 1.3. IBM continued to work on OS/2 up to version Warp 4 and Microsoft developed Windows NT.

The user interface was initially very similar to Windows 3.0, but by version Warp 4 many elements were altered. The most important innovation was the warp bar, located at the top of the screen. This was close in use to the Windows 95 start menu and taskbar, as it permitted access to frequently used programs, files, and other system objects.[37]

IBM has since stopped development of OS/2 and it is only used today in certain applications, or in companies where IBM had a strong following.

Figure 3.21: The NeXT computer's user interface

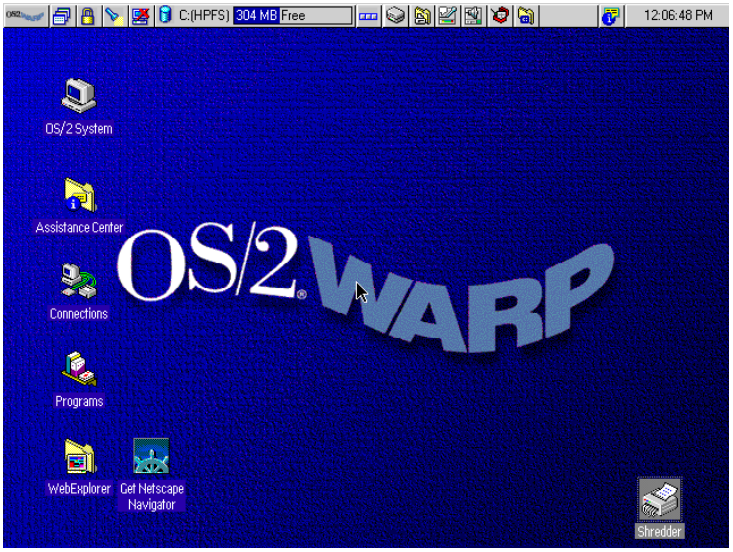

Figure 3.22: IBM's OS/2 Warp user interface

3.4.4 Be Inc BeOS

The BeOS (figure 3.23) was originally the operating system for the BeBox hardware, but was latter ported to PowerPC and x86 processors in an attempt to widen the user base. BeOS was a media-centric operating systems, and it excelled in audio, video and other multimedia applications. It had a small but strong user base, and it is still used at present by hardcore followers who continue to develop and share applications and patches.

Figure 3.23: The BeOS user interface

As for the user interface, the BeOS had more in common with Windows than with MacOS, but used some unique user interface elements: workspaces and deskbar. The deskbar was similar to Windows start menu and taskbar but was arranged vertically. Workspaces permitted the user to have multiple desktops, each with its background and screen resolution.

3.5 UNIX and GNU/Linux

In the UNIX world, the implementation of a graphical user interface was faced with a problem. Most UNIX computers were used through networks, with terminals connecting to larger, remote machines.

3.5.1 X Window System

Developed at MIT, the X Window System solved the problem by separating the X server from the X client, permitting the separation of the input and display functions of all applications to be separated from the rest of the application, allowing it to be available to a large number of remote users.[39]

The X Window System does not, however, contain specifications about the user interface, and a window manager application is necessary to control the placement and appearance of application windows.

Even the window manager cannot control each application's appearance, and it is common to have multiple applications with very distinct graphical features. Users normally prefer to use the X Window System with a desktop environment, which includes a window manager, multiple applications and consistent interface. OpenLook, MOTIF, and later the CDE (Common Desktop Environment) are early implementations of desktop environments in the X Window System.

3.5.2 Open Look and Open Windows, MOTIF and the CDE

Open Look was Sun Microsystems' specification for UNIX user interfaces. It was a user level specification, which could be implemented with various technologies. X Window System and NeWS toolkit versions were developed.[42]

Open Look was the basis for OpenWindows (figure 3.24), a desktop environment that was distinguished by welcoming the user with a blank screen. Applications were launched by right-clicking on the workspace and by selecting an application from an hierarchical pop up menu. Minimized applications were displayed as icons on the bottom of the screen.

The file manager displays the file structures as folders and files as icons, the current directory path as a sequence of folders and, when running, displays a wastebasket icon on the screen. Dragging a file to the workspace opens the file, and does not copy the file to the desktop or create a link.

MOTIF was a competitor to Open Look but was not a specification, it was a widget toolkit implemented over the X Window System.[41]

MOTIF is the basis for the Common Desktop Environment, or CDE (figure 3.25), which is in many ways similar to OpenWindows, but has a major addition: the tool bar located at the bottom of the screen. This tool bar presents icons with an associated option menu accessed by an arrow placed on top of each icon. This menu can be torn off and floating windows are created.

In the CDE, if a file is moved to the workplace, a shortcut to the file

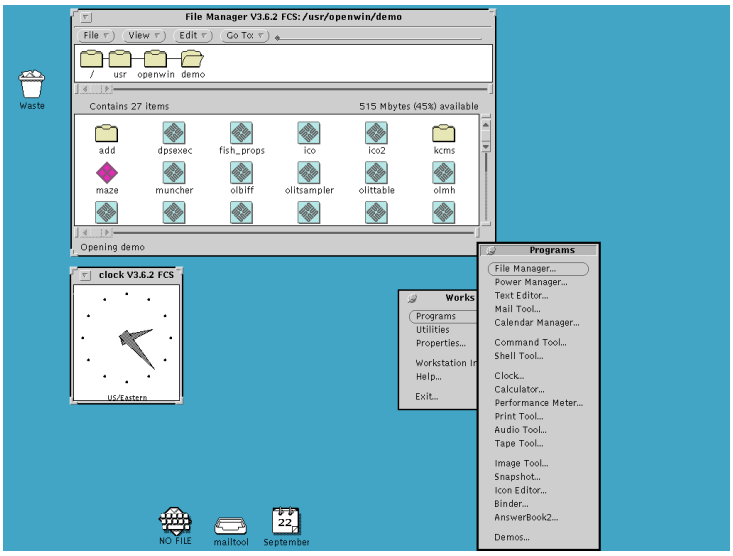

Figure 3.24: Sun's OpenWindows

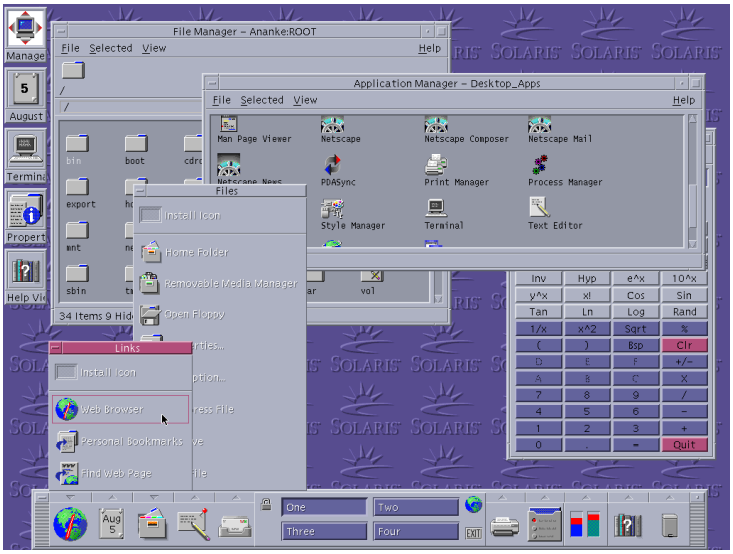

Figure 3.25: The Common Desktop Environment (CDE)

is created, unlike OpenWindows. The right-click application menu and file manager are similar to OpenWindows'.

Open Look and MOTIF were dethroned by the GTK and Qt toolkits, that are the base for current X Window System desktop environments GNOME and KDE.

3.6 Current OS GUIs

The main personal computer operating systems of today are evolutions of MacOS, Windows and UNIX. Although the graphical user interfaces of these operating systems are in many ways similar to those already discussed above, many changes have since been implemented. We will focus on describing the interaction model now implemented in each user interface, and the different approaches to file management and navigation, application management and navigation, and user tasks awareness.

3.6.1 Apple MacOS X

MacOS X is the current operating system for all Apple computers, from 12" laptops to servers (in this case MacOS X Server). The core of this operating system is BSD UNIX, and is not a continuation of the older MacOS system. Apple chose to adopt a proven operating system as a base to its own, like NeXT had done with its computer. Many technologies from NeXTSTEP are indeed used on MacOS X and others have been updated or served as inspiration for new ones.

The graphical user interface is (figure 3.26), however, an evolution of the old MacOS. It is of note the importance of this option for Apple, as it was critical to maintain its user base during this radical alteration of operating system. Apple managed to build a system that is extremely complex and, by designing a somewhat familiar user interface, conserve its user base and even convert UNIX users not pleased with current UNIX graphical user interfaces.

MacOS X maintains the classic MacOS model of a file centric user interface. It is not document centric like the Apple Lisa or the Xerox Star, although it is task aware in a sense, as will be mentioned below.

Interaction with MacOS X is mainly started through the Dock and through the Finder, the first being an applications management element and the later a file system navigator. It is also possible to use the desktop in the classic way, holding links to applications, files and folders.

Figure 3.26: Apple's MacOS X

The Dock MacOS X's Dock (figure 3.27) is an evolution of NeXTSTEP's Dock, although with a much improved look and with many additions. The Dock holds links to applications, files or folders, running applications, minimized windows and the wastebasket.

Figure 3.27: MacOS X's Dock

The user is free to add and remove icons from the Dock, normally creating a list of the most used applications, files and folders. To add an icon, the user needs only to drag the application, file or folder to the Dock. To remove an icon, the user drags the icon out of the Dock (it disappears in a puff of smoke).

The Finder The Finder is a file system navigator with three different views: icon view, list view and column view.

Icon view (figure 3.28) implements the classic folder navigation with files and folders presented by icons. The folder structure can be navigated by clicking in these icons.

Figure 3.28: MacOS X's Finder in Icon view

List view (figure 3.29) shows the file system as a list, where the folders are identified by an arrow. This arrow can be clicked and the contents of the folder are presented in an indented list.

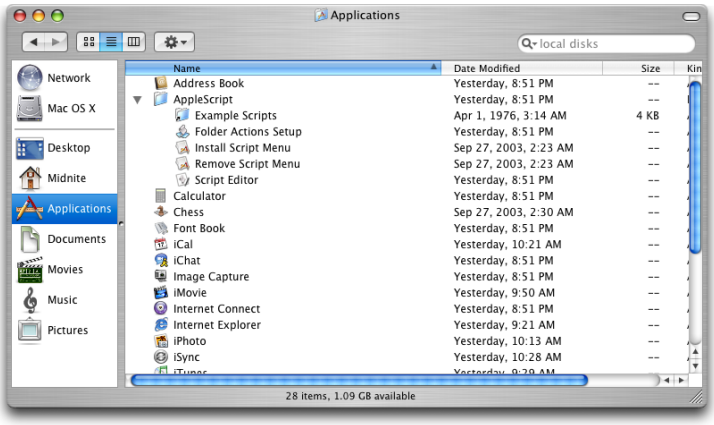

Figure 3.29: MacOS X's Finder in List view

Column view (figure 3.30) presents the file system as a sequence of columns which are navigated vertically. When a folder is selected, the current column is shifted to the left and the new column occupies its space. When a file is selected, a preview is presented at the rightmost column.

Task awareness using applications MacOS X comes installed on Apple computers accompanied with many applications. We will refer here iPhoto,

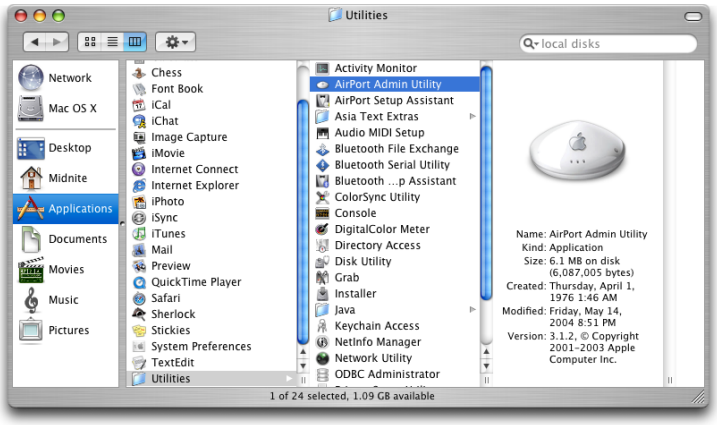

Figure 3.30: MacOS X's Finder in Column view

iMovie and iTunes (figure 3.31), which are designed to present the user with a complete interface for determinate tasks.

These applications eliminate the need for the user to directly manage photo, movie and music files, presenting an organization more suited to each kind of information. Each application also gives the user access to all major tasks, like photo manipulation, movie editing and music listening.

3.6.2 Microsoft Windows XP

Windows XP (figure 3.32) is currently the most used personal computer operating system, and as such, also the most used graphical user interface. Windows XP is an evolution of Windows NT (and not of Windows 95) so it is a totally graphic operating system.

Windows users saw its user interface evolve through the years with numerous updates but some were more important: the start menu, the taskbar, the integration of Internet Explorer and folder tasks.

The most common ways to initiate interaction with Windows XP are the start menu and the file browser. It is also possible to place links to files and applications on the desktop, and use those links to start interaction.

The start menu The current version of the start menu (figure 3.33) includes many different elements that work in different ways.

On the top left there are 'pinned' applications, normally the web browser and the email application, although the user can add other applications to these.

Figure 3.31: Apple's iMovie, iPhoto and iTunes

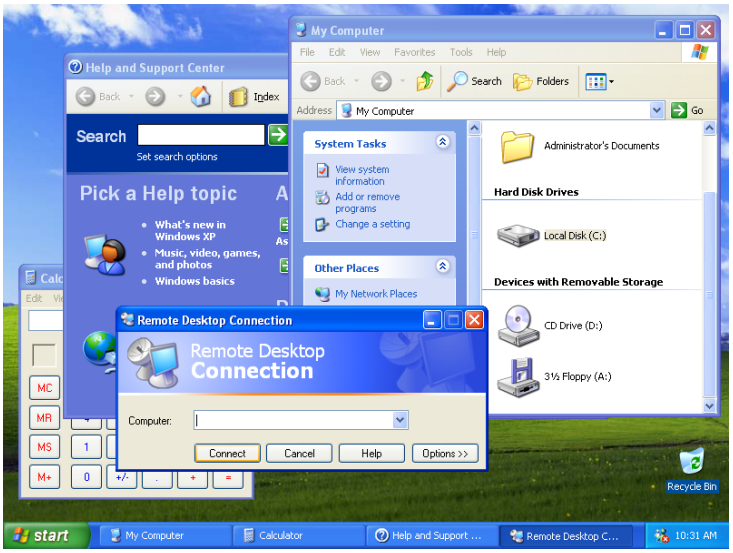

Figure 3.32: Microsoft's Windows XP

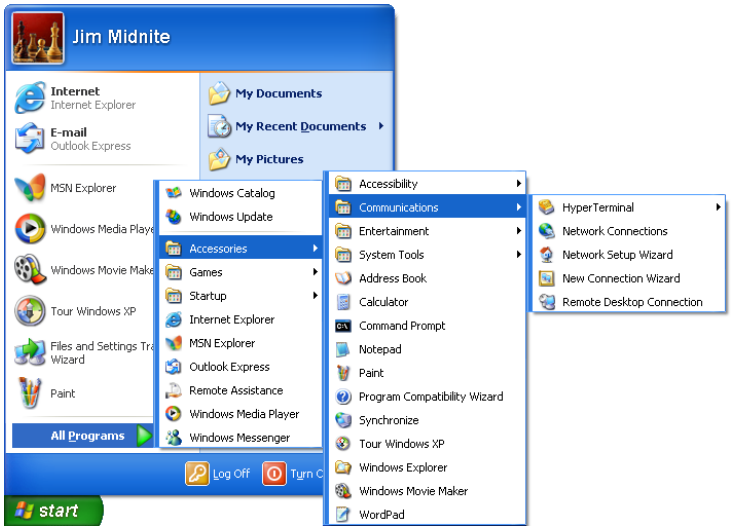

Figure 3.33: Windows XP's start menu

Below is a list the most used applications, that is continuously updated by the system, which allows the user to remove applications but not to add them.

At the right, there are links to important folders like 'My Documents', 'My Pictures' and 'My Computer', as well as links to the Control Panel, Printer panel and other applications.

There is also the ´All Programs' menu, which opens a hierarchical structure of links to applications or files. This structure is user editable, and applications normally add their own links when they are installed.

The file browser Windows XP's file browser (figure 3.34) has various display options (thumbnails, tiles, icons, list and details), but the interaction is always similar to normal folder navigation.

There is a 'Folders' button that shows a tree view of the folders at left, with the contents of each folder presented in the main view. This option is equivalent to the 'Windows Explorer' application.

Task awareness using folder tasks While navigating the file structure, Windows XP presents on the left a list of tasks. These tasks change depending of the current folder and current selected icon.

For example, while browsing the 'My Pictures' folder, there are links to 'View as slide show', 'Order prints on line', 'Print pictures' and 'Copy all items to CD'. Normal file and folder tasks are 'Make a new folder', 'Publish

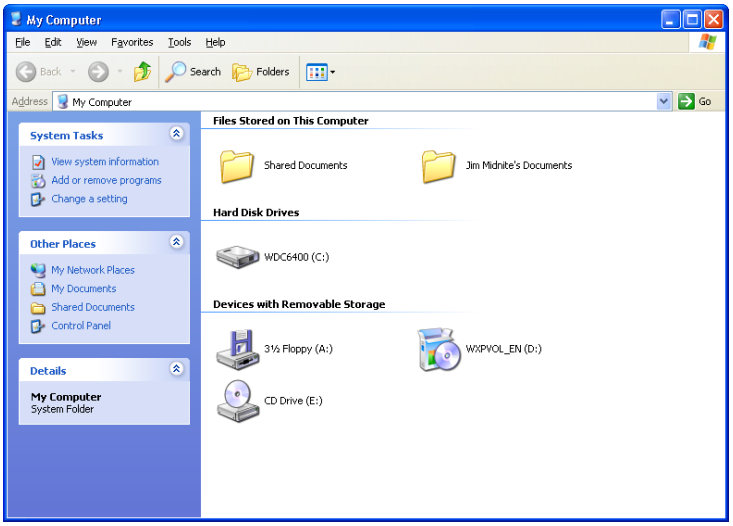

Figure 3.34: Windows XP's file browser

this folder to the Web', 'Share this folder', 'Rename this file', 'Move this file' and 'Delete this file', among others.

3.6.3 GNOME and KDE

GNOME and KDE are the most commonly used Linux desktop environments, and one of them is chosen as the default in most Linux distributions. Both GNOME (figure 3.35) and KDE (figure 3.36) are in most ways similar to Windows XP and MacOS X, from which they get most of the inspiration for the user interface. It is also possible to perceive many influences from older UNIX desktop environments like the CDE and also from NeXTSTEP.

As both of these desktop environments are very configurable, the user has many options to choose from, and by consequence, the typical desktop varies greatly from user to user, as does the default appearance, due to changes in almost every new release.

As for the file browser, both GNOME and KDE have different applications for this job. GNOME uses Nautilus (figure 3.37) and KDE uses Konqueror (figure 3.38). Again with influences from other graphical user interfaces, Nautilus and Konqueror try to present the most commonly used options in Windows XP and MacOS X. Nautilus is the most slim-lined of the two, as Konqueror opts to follow Windows' lead and is itself a full web browser. Konqueror is capable of presenting full web pages, and is also capable of presenting previews of documents while navigating the file system.

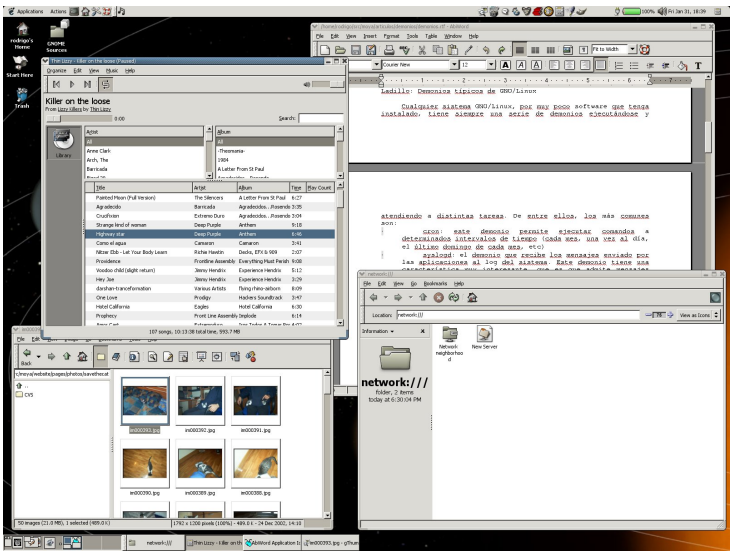

Figure 3.35: The GNOME desktop environment

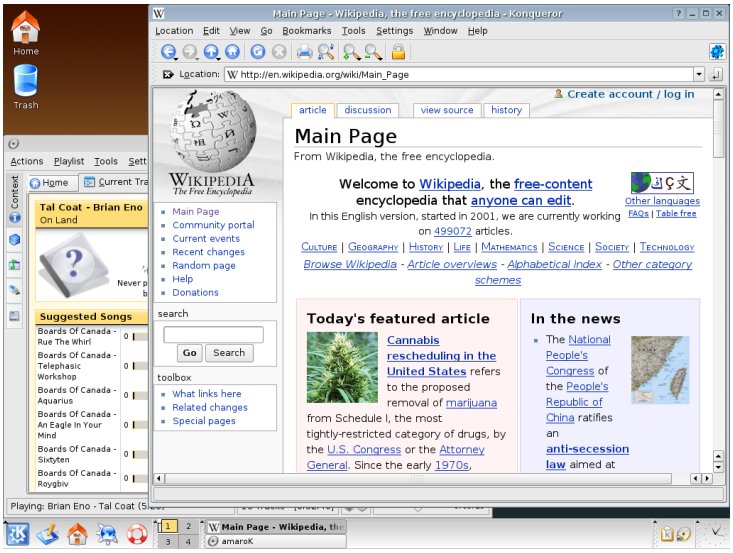

Figure 3.36: The KDE desktop environment

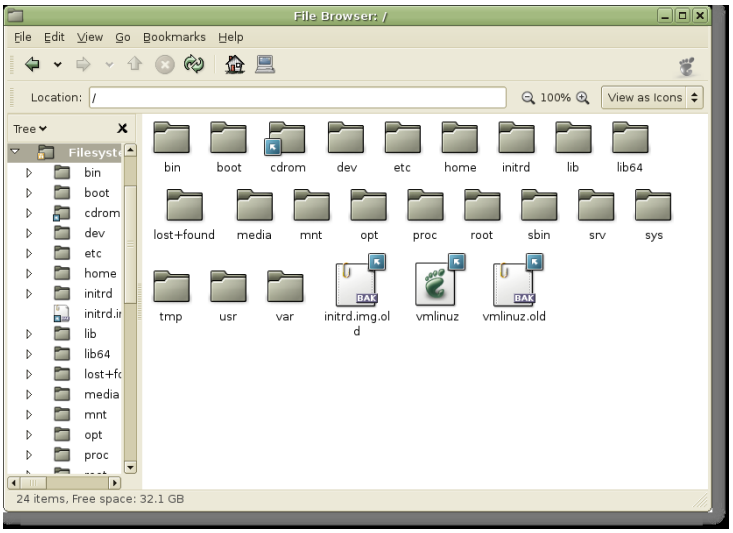

Figure 3.37: GNOME's Nautilus

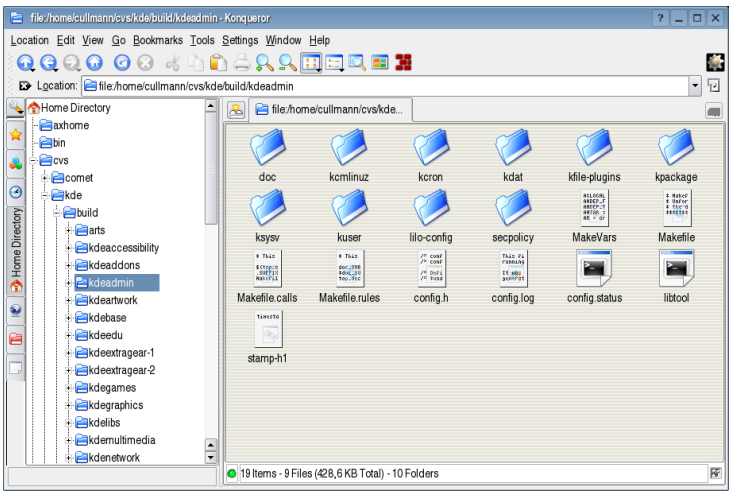

Figure 3.38: KDE's Konqueror

3.7 The World Wide Web

A great alteration in computer use has been taking place in the last decade: the World Wide Web. With rising user numbers all over the world, the Web is now almost indispensable to any personal computer user.

The alteration of habits and attitude toward computer use brought by the dissemination of the Web should not be discarded, and should be reflected also in the way new graphical user interfaces are designed. We will briefly look at the history of the Web and describe new technologies and leading Web-based applications.

3.7.1 Hypertext and the Web

The seeds of hypertext and the Web can be identified as early as Paul Otlet's scholar's workstation and Vannevar Bush's memex. Douglas Engelbart's On-Line System was one of the first projects to implement hypertext. Apple's hypermedia application called Hypercard (released in 1987) was a hit and did much to make hypertext popular.[44]

The Web itself was created by Tim Berners-Lee (figure 3.39) and Robert Cailliau at CERN, initially as a project named ENQUIRE. Tim Berners-Lee made a formal proposal for the World Wide Web on November 12, 1990. Tim Berners-Lee's breakthrough was to join hypertext and the Internet but, as none of these technical communities wanted to start work on his World Wide Web, Tim Berners-Lee wrote a Web server, a Web browser and editor during Christmas holidays of 1990.[45]

Figure 3.39: Sir Tim Berners-Lee

On April 30, 1993, CERN gave the World Wide Web to the world, making it free for anyone.[45]

3.7.2 Ajax

The classic web model is closely based on hypertext, with web pages presented to the user as he clicks on links and requests them. While this model is perfect for hypermedia, it is limiting for high interactivity applications.

Ajax (or Asynchronous JavaScript and XML) is a web development technique that uses a combination of XHTML and CSS, the Document Object Model, XML and XSLT, the XMLHttpRequest object and JavaScript to eliminate the problems with classic web interaction.[46]

As Ajax allows for contact with the web server without stopping interaction with the user (figure 3.40), many interaction techniques that could not be used or were too time consuming to use can now be implemented.

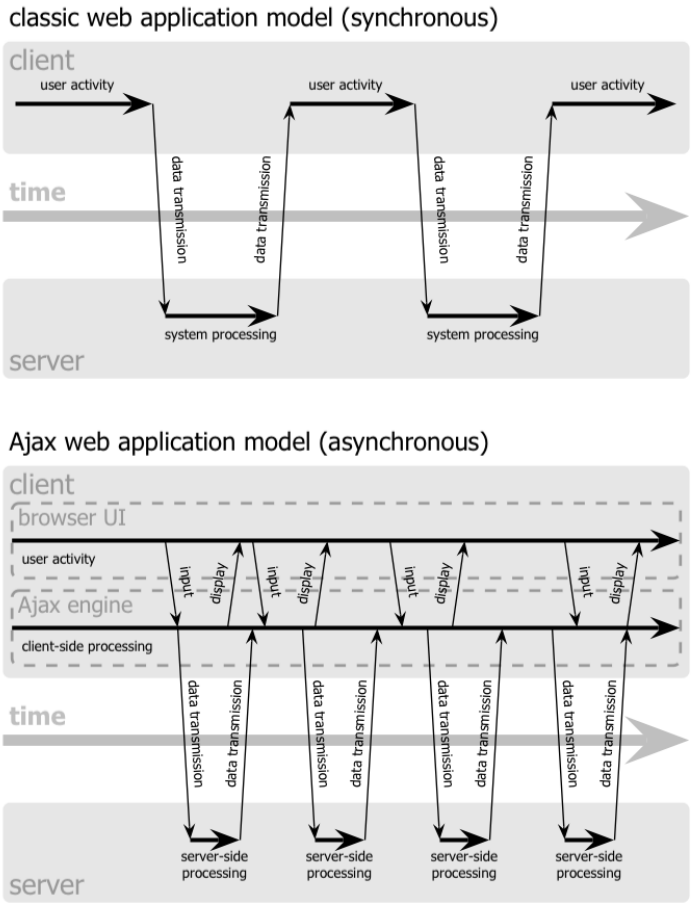

Figure 3.40: The classic Web application model and the Ajax application model

One of the first high-profile uses of Ajax was done by Google, by implementing its 'Google Suggest' feature: while the user is keying in the search phrase, Google contacts the server and presents, without stopping interaction with the user, a drop-down list of the most searched phrases that begin with what the user has already typed in.

3.7.3 Web Applications

There are numerous examples of successful web applications. Here we will focus on some recent applications that are innovative and represent new approaches not only to the Web, but also to user interface design in general.

Gmail (www.gmail.com)

Gmail (figure 3.41) is a web mail application that relies on Ajax, Web standards, a carefully designed user interface and a powerful search system to give the user a very strong alternative to email applications. Compared to earlier web mail applications, Gmail presents a much more effective user interface that is clearly designed for the web, and not based on desktop email applications.

Figure 3.41: Google's Gmail

Google Maps (maps.google.com)

Google Maps (figure 3.42) is another great example of simple interface design from Google, and is probably also the most complex use of Ajax to date. Google Maps is a complete Map application that also responds to Google-type searches. This application does not reload while the user interacts, and presents maps, satellite images using Ajax to load images from the server.

Figure 3.42: Google Maps

Flickr (www.flickr.com)

One of the first photo-sharing web applications, Flickr (figure 3.43) presents the user with an easy to use interface for uploading, managing and documenting photos. Flickr also uses Ajax to make the interaction and photo presentation a better experience. There is also a comments feature that allows viewers to comment on user's photos.

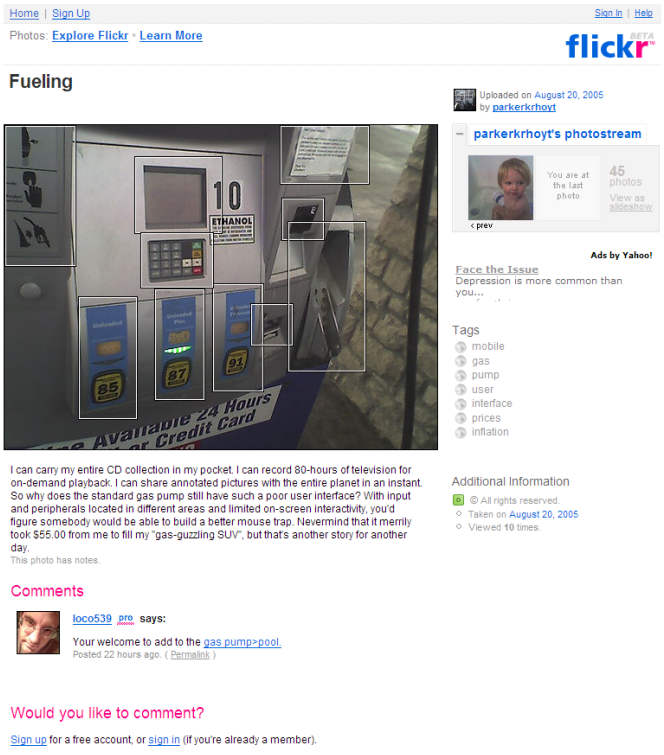

Figure 3.43: Flickr

Basecamp (www.basecamp.com)

A good example of how a Web application can bring a new perspective on old problems, Basecamp (figure 3.44) is a project management application that has its focus on communication. A clean and simple user interface, combining standard Web navigation with Ajax powered interactivity, proves that an application as complex as project management can be translated to the Web.

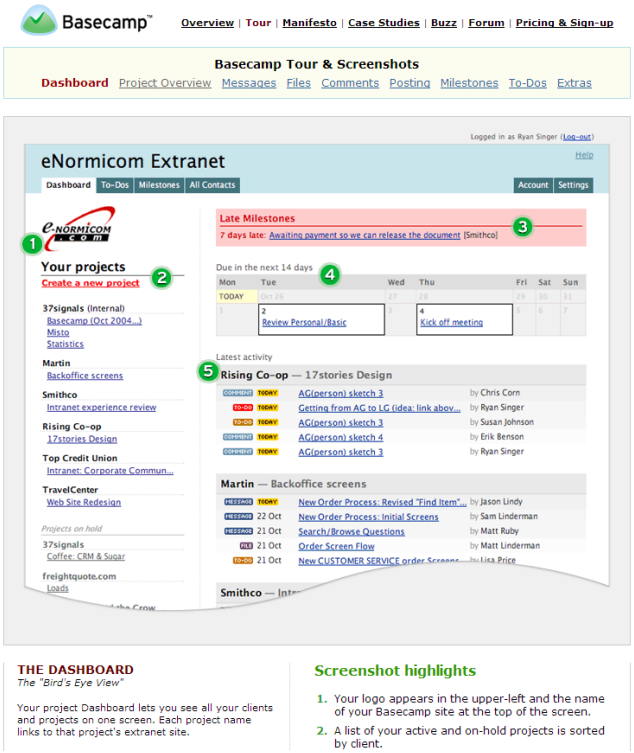

Figure 3.44: Basecamp

Ta-da List (www.tadalist.com)

From the makers of Basecamp, Ta-da List (figure 3.45) is a simple but effective Web application that manages to do lists. This Web application is even simpler to use than many conventional to do applications. This is another example of the power of a simple user interface based on standard Web navigation and Ajax interactivity.

Figure 3.45: TaDaList

Blogs

The success of Web logs (blogs) is based on a simple fact: everyone likes to share its opinion. The Web was envisioned as a way to promote exchange of information between scientists, and eventually became another media for corporations. Blogs (like wikis) offered the opportunities for common people to be heard, removing the technological barrier that prevented ideas to be published on the Web. Simple user interfaces that manage and publish information was all that was needed for another Web revolution. Blogs are normally used as opinion sites (figure 3.46), news sites or personal diaries.

Wikis

Wikis follow the same philosophy as blogs, allowing non-technically minded users to publish information on the Web. Wikis have, however, a big difference from blogs: they allow anyone to edit already posted information. Wikis are mainly used for collaborative writing (figure 3.47), presenting easy access to editing pages and using simple user interfaces.

3.8 Overview

As was shown in this chapter, the interest in building a system that simplifies information work has been alive since the 30's. Paul Otlet and Vannevar Bush defined machines with the objective of easing the overflow of information that began to be evident at the time. These machines, however, were never built.

Digital computers provided the technology to implement many of the ideas initiated by these pioneers, but it was some time before the general

Figure 3.46: Gizmodo, a popular gadget Blog

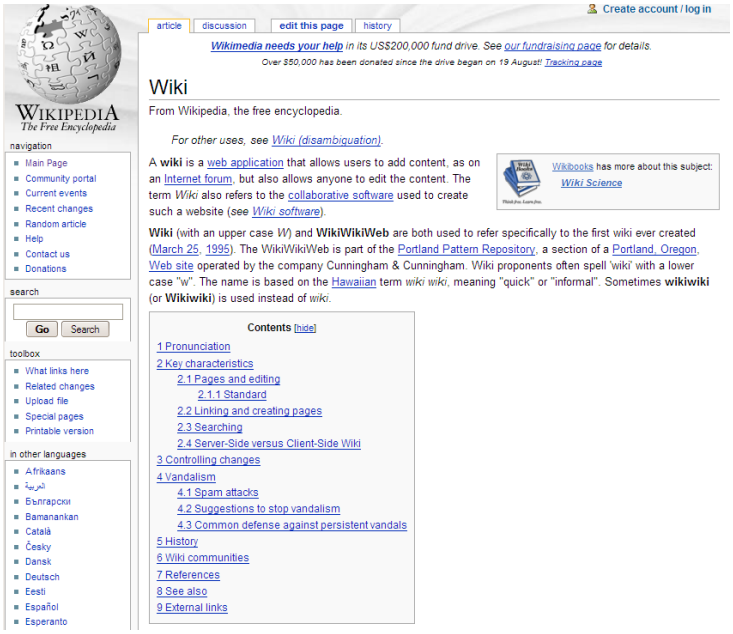

Figure 3.47: Wikipedia, a very successful wiki encyclopedia

public could benefit from them. Although many personal computers were born before it, the Apple Mac was arguably the first to offer a user interface that could be understood and used by non-technical minded users. The revolution of the Graphical User Interface had begun.

We have, however, to realize that the Mac was crippled technically and therefore also in terms of its user interface. As can be seen by analyzing both Xerox's Star[21] and Apple's own Lisa [25], many good ideas, concepts and models were discarded in building the Mac. In fact, the Mac's desktop metaphor design, the basis of almost all future graphical user interfaces, was notably inferior to the Lisa's.

Nonetheless, the file-centric desktop metaphor of the Mac became therefore the model for all future Apple operating systems, Microsoft Windows, and UNIX desktop environments. Almost all evolution in operating system user interfaces was done on new elements designed to lessen the impact of modern computer user necessities on a fragile user interface model.

The next revolution to hit the computer world did not come from a computer or software company: it came from the academic community. The Web had a phenomenal impact on the World and changed the way we look at our personal computers. In terms of user interface design, the Web opened the computer user's mind to a different way of interacting.

Hypertext and hypermedia are radically different from classic application user interfaces and had relatively small impact before the Web. Now, however, most computer user are familiarized with the hypermedia model and accept it, as well as other models that combine both hypermedia and application style elements. We can safely say, judging by the success of Web based applications like Google, Gmail or Flickr, that navigating information structures and task based user interfaces is becoming a normal skill for computer users.

In the next chapter a proposal is presented that incorporates both modern Web interaction models and lessons learned from earlier user interface designs, with the objective of creating a more adequate computer working environment for the modern computer user.

Chapter 4

The task aware user interface proposal

The basis for this proposal is the intention of creating an operating system user interface that eliminates problems in current user interfaces. A choice was made to center the user interface model on the user tasks, in the form of an hierarchical structure of tasks, created and maintained by the user. Each task is the home for file, folder, application, hyperlink and annotation associations.

The model of interaction and the specifications for this proposal are now presented.

4.1 Abstract Model

A possible solution, implementing a task aware interaction model, was defined. This model is user centered, task aware and includes annotation facilities. It aims to:

Make the UI reflect the user's tasks To achieve fast navigation and recognition of the UI's structure, the system should reflect the user's own understanding and model of its tasks, and not a predefined structure.

A common idea behind the work of Paul Otlet, Vannevar Bush and Douglas Engelbart was the definition of a system which sought to augment the capabilities of the user in handling complex tasks and complex information. To achieve this, the system has to *talk* to the user in terms he can understand and in actions he is familiar with.

One way to accomplish this is to limit the tasks performed by the system to a well defined subset. This was the option taken in both the Xerox Star

and the Apple Lisa, devoted to standard office use. The document centric approach and invisible applications suited this environment well, as the user needed only to think about the documents he was using, as all office related tasks were available transparently.

Current personal computers users expect, however, an open-ended approach to system's capabilities. They expect to be able to install new applications which can aid them in new kinds of tasks. All major modern operating system user interfaces fail in handling this new capabilities, and do not present them in an organized and coherent way. It is the user that must struggle with the operating system to do his work.

The system has therefore to be able to learn each user's individual task model, and present it during normal use.

Use the task structure to group distinct elements The user defined task structure should accommodate files, directories, links and applications, to be associated to each defined task. This way, relevant elements will be in close proximity, grouped according to its relevance to each task, and not distributed through distinct structures.

To accomplish common tasks, it is normal for the user to open different files, navigate directories, follow links and open various applications. Although most modern operating system user interfaces include facilities to accomplish a similar goal (for example, creating a folder for each task and creating symbolic links to files, other folders, applications and web links), they are not normally used, mainly due to the complexity of the process.

The benefits of having all the elements you need to accomplish a task at close reach, and the comfort of a user interface that makes it simple to manage these elements, make this feature indispensable to the proposed system.

Induce the user to organize, annotate and comment To help the user document its interaction with the computer, a fast and unobtrusive way to add notes, comments and work logs should be offered. The user should be induced into commenting his own work, in order to aid in user interface structure recognition and personal organization.

Even in early developments, the importance of annotating the user's work was noted: Paul Otlet's description of his scholar's workstation included the capabilities of writing information and creating relationships between documents; Vannevar Bush's memex was envisioned with the possibility of adding new information to the system, either in the form of new entries or annotations to already existing entries; Douglas Engelbart's On-Line System contained 'The Journal', into which each user inserted information.

Again, it is possible for the user to add information to current operating system's user interfaces, but the process is not straightforward. An integrated approach is needed, where annotations are easily made while working on a task.

4.2 Specifications

Based on this model, the proposal took shape with the following specifications:

Navigation through an editable task structure The user views the computer system as a hierarchical structure of tasks and subtasks. Each task has a name, a short description and an annotation area. Subtasks can be added to each task, and can be deleted, moved or edited while browsing. There is not a dedicated edit mode or browse mode: the task structure is meant to be in constant evolution to respond to changes in the user's vision and approach. The implementation of this is described in sections 5.3 and 5.4.

Association of files, directories, applications and links to each task Each task has associated files and directories, applications and links. These are presented with names and descriptions which can also be edited, added, deleted or moved while browsing. Files and directories can be opened, applications can be run and links can be followed directly by clicking on the appropriate action. Sections 5.5, 5.6 and 5.7 show how these actions can be executed in the prototype.

Annotation, comments and logs associated with each task Annotations are added to each task by way of a simple text field, which can be edited, and will recognize the name of subtasks, files, directories, applications and links that are associated with the task, providing direct links to open files, run applications and follow links. This annotation area can be used by the user as a notepad for ideas, to do lists, extended descriptions or work log. Its use enriches the task structure with related content, helps users remember past and future actions and builds appropriation, much like a personal diary or workbook. The annotation facilities of the prototype are described in section 5.8.

Descriptions associated with each file, applications and links To aid in the recognition of each element associated with a task, these can be

named and described. This extra information will be important to distinguish files, directories or applications with the same name, or multiple versions of documents. This, in conjunction with task annotation, gives the user the possibility to easily add information while working, without having to use additional files or information management applications. The description facilities are presented in sections 5.5, 5.6 and 5.7.

4.3 Design and Implementation

The proposed user interface has been designed to be used as a layer above current operating system user interfaces. It does not offer alternatives to physical file and folder management, system administration and other OS facilities. It was also designed to be run on various operating systems like Windows, MacOS and UNIX.

Although the proposed user interface was idealized as browser-based, there is no reason for it not to be implemented as an application using custom UI objects. It seems, however, a good idea to use existing technologies and applications that are already familiar to the user.

The technological design of the proposed user interface lies manly on the inclusion of a layer of services available to the user, which allow for all the functionalities described above: managing an editable task structure, association of files, directories, applications and links to tasks, association of annotations, comments and logs to tasks and association of descriptions to files, applications and links.

This services layer uses a local database to facilitate the storage of data, as well as allowing for searches and querying.

Interaction with the user is made via browser, which communicates with a local webserver that in turn uses the layer of services mentioned above. A custom webserver is therefore needed, as it is necessary for it to have an integration with the operating system normally not available in common webservers.

4.4 User reaction

When this model was explained to the interviewed users, a very positive reaction was obtained. All of them were open to the idea of a different way to use their computers, and were genuinely interested in something they though could help them organize their computers and work better and faster.

One user said, “I think this form of organization would make it easier to

work. If for a task I had direct access to all related files, applications, links and also my notes, it would certainly be faster to work. I think it makes good sense."

"I see lots of problems in my day to day use of the computer," another user said, " from your description of this different way of working, I think it solves some of these problems. If it saves me from browsing for files and for applications and even lets me make notes, and it presents these things to me associated to a task, I think it would help me a great deal."

About the annotation facilities, one user said: "That would help me to maintain a history of my work, directly associated to it. I think this makes a lot of sense. As a to do list also... It would help me, definitely."

About the description of objects, another user said: "I now use very long filenames so I know what is in each file. I use things like 'copy of file where I corrected sentence this or that.doc' and so on, and it is not practical. You can't see long filenames very well on the folders... As my other option is to make a note on paper about what is in the files, I continue to use very long filenames. If I could write a long description and associate with the files, it would make things a lot better!"

4.5 Overview

The model presented gave the necessary leads for the specification of a user interface that proposes to take a new approach focused on the user's tasks. The proposal presented in this chapter resulted, therefore, in the implementation of a prototype, which will described in the next chapter.

Chapter 5

The prototype

In order to get user reaction on the proposed solution for a task aware operating system user interface, a prototype was developed. This prototype follows closely the proposed solution presented, and is completely functional, permitting user installation and normal daily use.

A choice was made to include only the basic functionalities and minimize operating system integration. This way, the work on the prototype was focused on the main ideas behind the proposed solution, and therefore testing was also focused on these main ideas.

5.1 Architecture and global overview

The main entity in this user interface is the task, and the main navigation structure is the task hierarchy. All user navigation is made throughout this structure, which effectively substitutes the folder structure, the application launcher structure and the browser favorites structure. The user only has to manage this one hierarchy, with the added bonus of all information needed to accomplish a task being associated to it.

The user interface prototype has one main screen, which represents a task, on which five areas are distinguishable: the subtask area, the annotation area, the files and folders area, the applications area and the links area.

Navigation throughout the task structure is made using the subtask area and the breadcrumb links above the task name. By clicking on a subtask, the selected subtask is made the current task. Information about this task is then presented in the same manner as its parent task. The breadcrumb navigation is updated to provide a link to the task's ancestors.

All actions, like creating a new subtask, associating a file or editing annotations are done in this screen, and are accessed by the action button

located inside each distinct area. New windows are only opened for action confirmations or to browse for files or other tasks.

In this way, a simple interface is presented to the user, using a common Web interaction style, that is easily perceived if the user has had contact with Web applications like web mail or photo sharing, and is easily learned by users that are not very familiar with these.

All descriptions, be it of tasks, files or applications and all annotations are presented to the user in a way that can be read if needed, but does not obfuscate more important information like task or file names.

5.2 Implementation of the prototype

The diagram bellow presents a block view of the prototype, showing the various objects of the system, communication paths and technologies used.

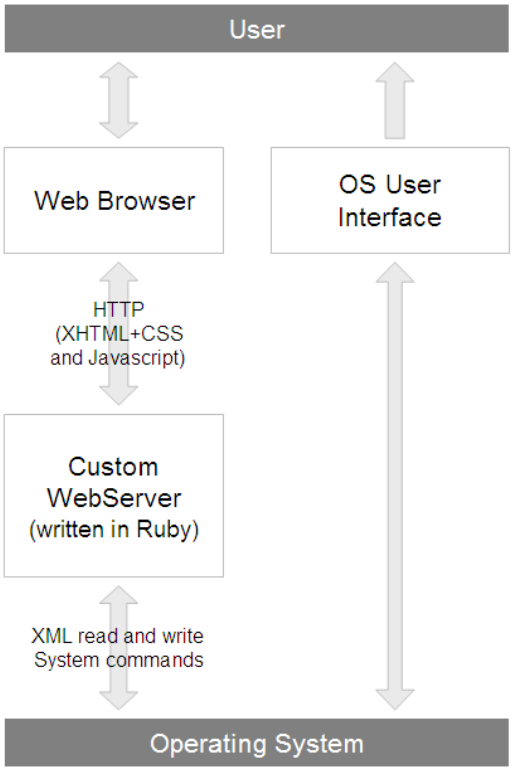

Figure 5.1: A block view of the prototype

The prototype UI is presented via a web browser, and is coherent with normal website browsing. Users follow links to navigate tasks, open files,

run applications and browse to links; actions (editing, deleting, moving and creating) are presented as labeled buttons. The UI uses client-side XHTML, CSS and JavaScript technology and communicates with a custom web server, written in Ruby, installed locally in the user's computer, allowing direct communication with the operating system. Data is stored as XML files, which are continuously updated while the system is used. The use of the browser-based interface can be seen on the following pages.

These technological options were made to achieve four main objectives: fast development time, portability, use of standard web technologies and use of open source programming languages. The use of Ajax (discussed in section 3.7.2) was not necessary, as the local custom web server responds fast enough to not represent an obstacle to interaction. If this was not the case, Ajax could be used to speed up the systems' response.

A detailed description of the prototype follows. User reactions, obtained throughout user testing and post-test interview, are also presented.

5.3 The task and task structure

The foundation of the user's navigation is the task structure. Starting with the 'Home' task (figure 5.2), it is possible to navigate through this structure by following the subtask links, positioned at the left. Subtasks of the present task are represented by name and description on this link list.

For each task, its associated annotations are presented at the center, as well as its associated files and folders, applications and links. Each of these groups is identified with a color for easy recognition.

By following one of the subtask links, all information for the selected subtask is presented, including possible subtasks (figure 5.3).

As the navigation along the task structure continues, links of all predecessors of the current task are presented, building a breadcrumb navigation aid. This normally permits fast navigation to the last viewed tasks, as well aiding in visualizing the current position in the task structure.

5.4 Creation of tasks

While viewing a task, it is possible to immediately add a new subtask by clicking on the 'New' action, located above the subtask list (figure 5.4).

The user is prompted for a name and description, and after clicking on the 'Submit new subtask' action, the newly created subtask is presented in the subtask list, along with existing subtasks (figure 5.5). To view or edit

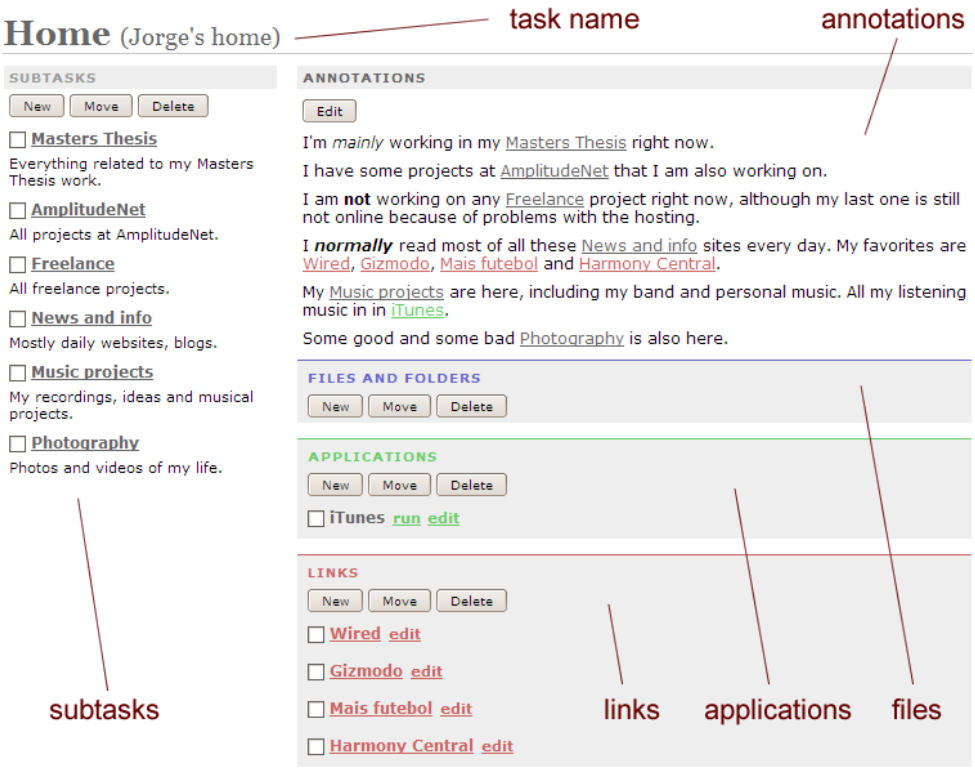

Figure 5.2: The 'Home' task

Figure 5.3: The 'MastersThesis' subtask

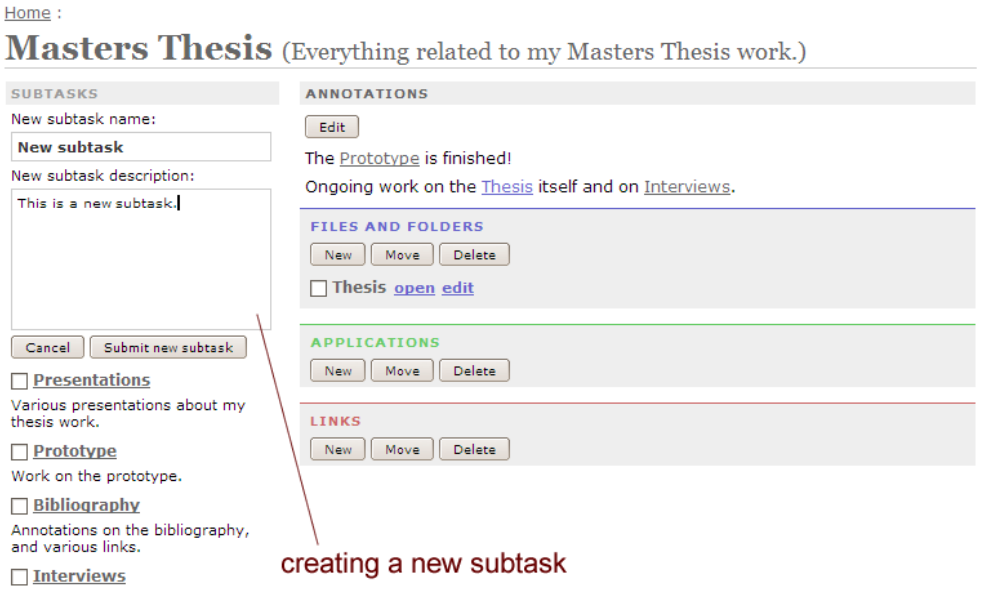

Figure 5.4: Prompt for new associated task attributes

the subtask, it is necessary to navigate to it, after which it will be possible to associate annotations, files and folders, applications and links.

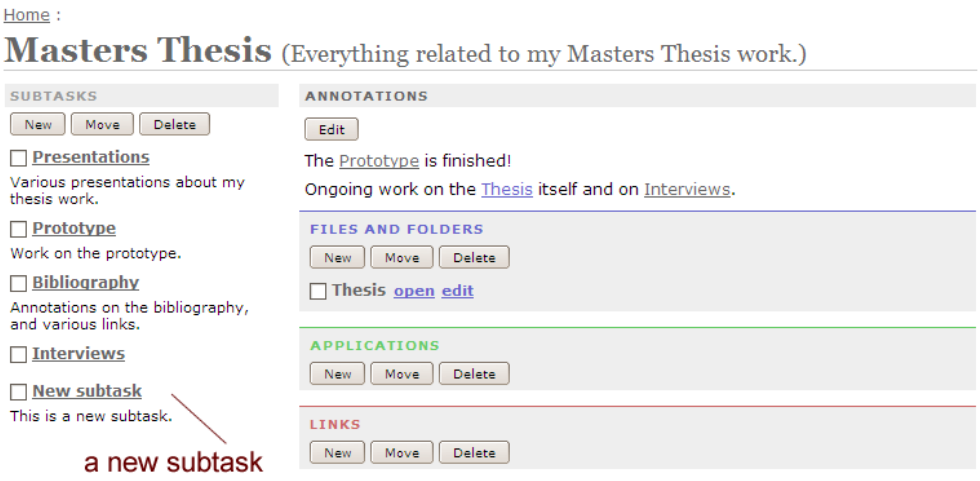

Figure 5.5: New task association inserted

Moving subtasks to another task is done by selecting the subtasks to move, using the boxes on the left of each subtask, and clicking on the 'Move' action (figure 5.6).

Deleting subtasks is done in a similar way, selecting the subtasks to delete and clicking on the 'Delete' action (figure 5.7).

By associating closely the creating, moving and deleting actions with the task structure navigation, we create an environment where alterations to this structure are easy to accomplish. The objective is not to serve a static structure, but to present it and at the same time permit easy evolution, so it can follow changes in the user's own mental model of its tasks.

5.5 Associating files and folders to a task

Each task has a group of associated files and folders that can be opened by clicking on the 'Open' link associated with each file or folder. This click instructs the underlying operating system and opens the file or folder, normally with the correct application.

To create a new association of a file or folder, the 'New' action, which is located above the file and folder list, is used (figure 5.8).

A prompt for a new file or folder name, description and link appears. To fill the file or folder link field, the user can click on the 'Browse' action, which

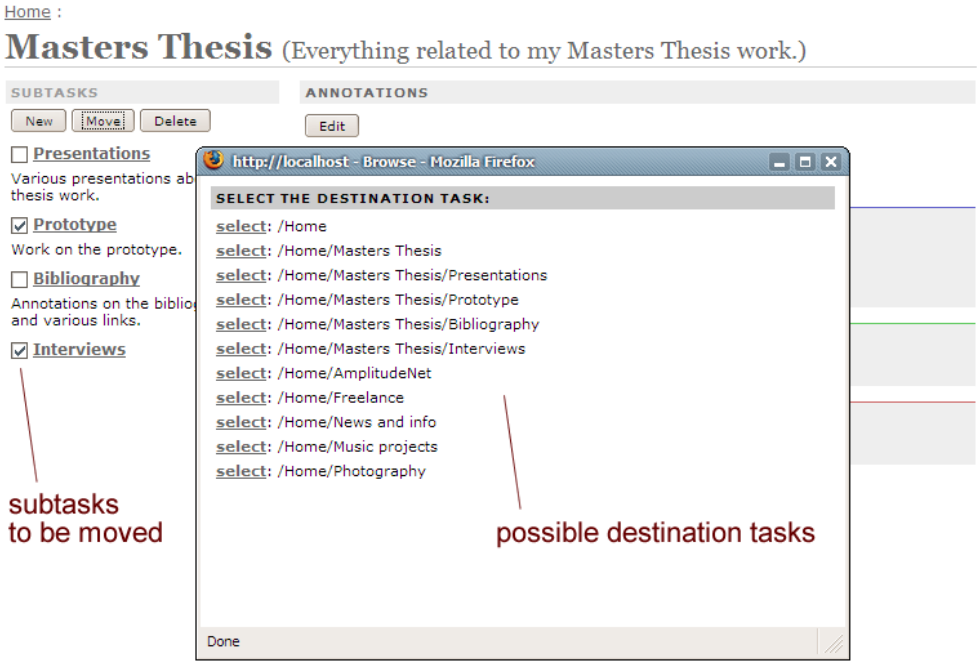

Figure 5.6: Moving a subtask association

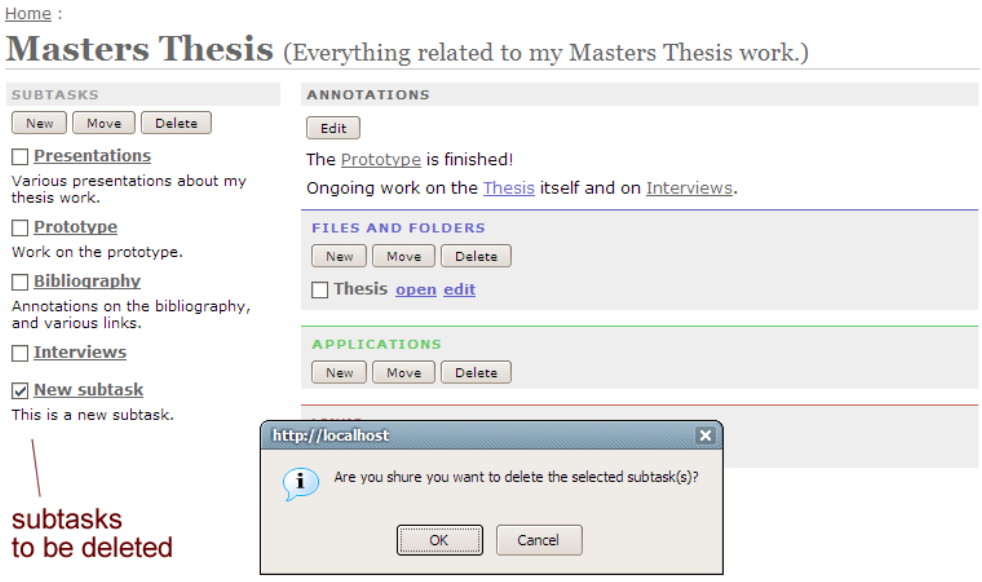

Figure 5.7: Deleting a subtask association

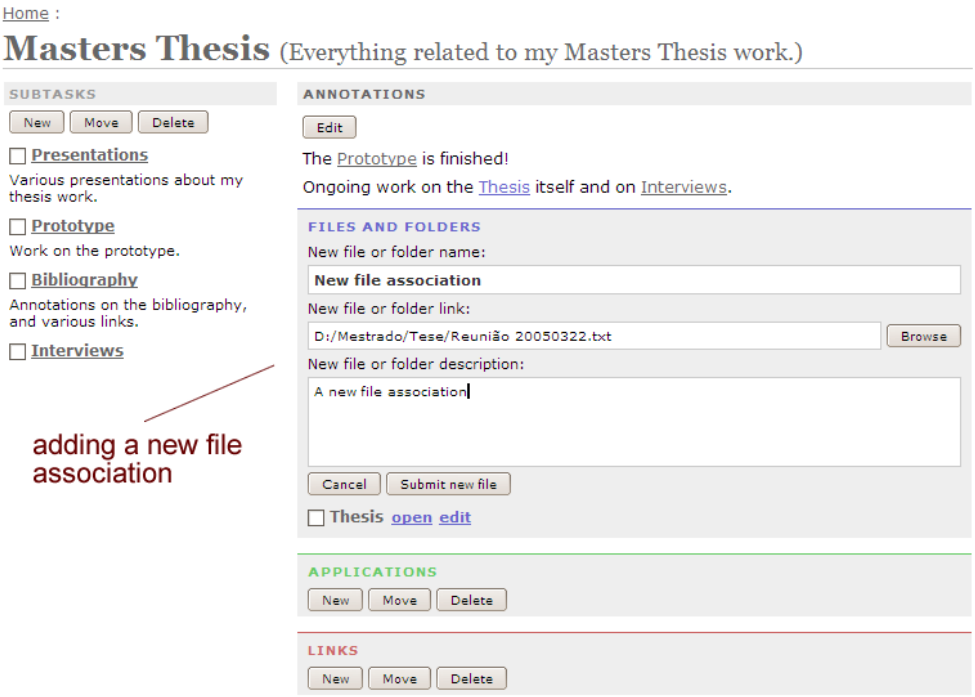

Figure 5.8: Prompt for new associated file attributes

permits a file system browse. By clicking in the 'Submit new file' action, the new file or folder association is added to the list (figure 5.9).

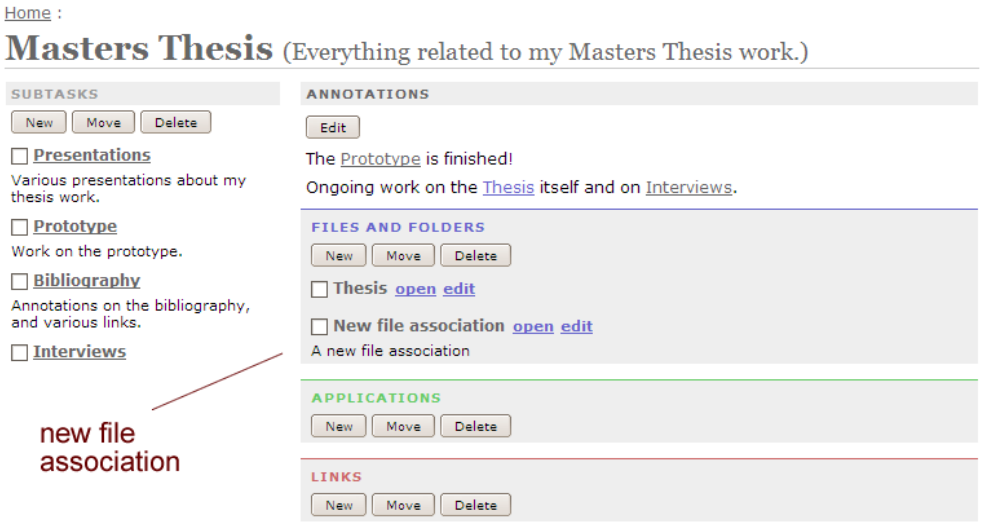

Figure 5.9: New file association inserted

As in the case of the subtasks, moving files or folders to another task is done by selecting the file or folder to move, using the boxes on the left of each entry listed, and clicking on the 'Move' action (figure 5.10). As only links to files and folders are created, moving has no effect on the file system.

To delete one or more file or folder, it is necessary to select the entries to delete and click on the 'Delete' action (figure 5.11). As with the moving action, the delete action has no effect on the file system, and this action is not destructible at that level.

5.6 Associating applications to a task

Applications are also associated with each task, and can be run by clicking on the 'run' link, presented next to each application listed. The run command calls on the operating system to run the selected application.

To associate an application with a task, the 'New' action is used (figures 5.12 and 5.13).

This action prompts for a name, description and application link. As with the creation of a new file association, the application link can be selected by

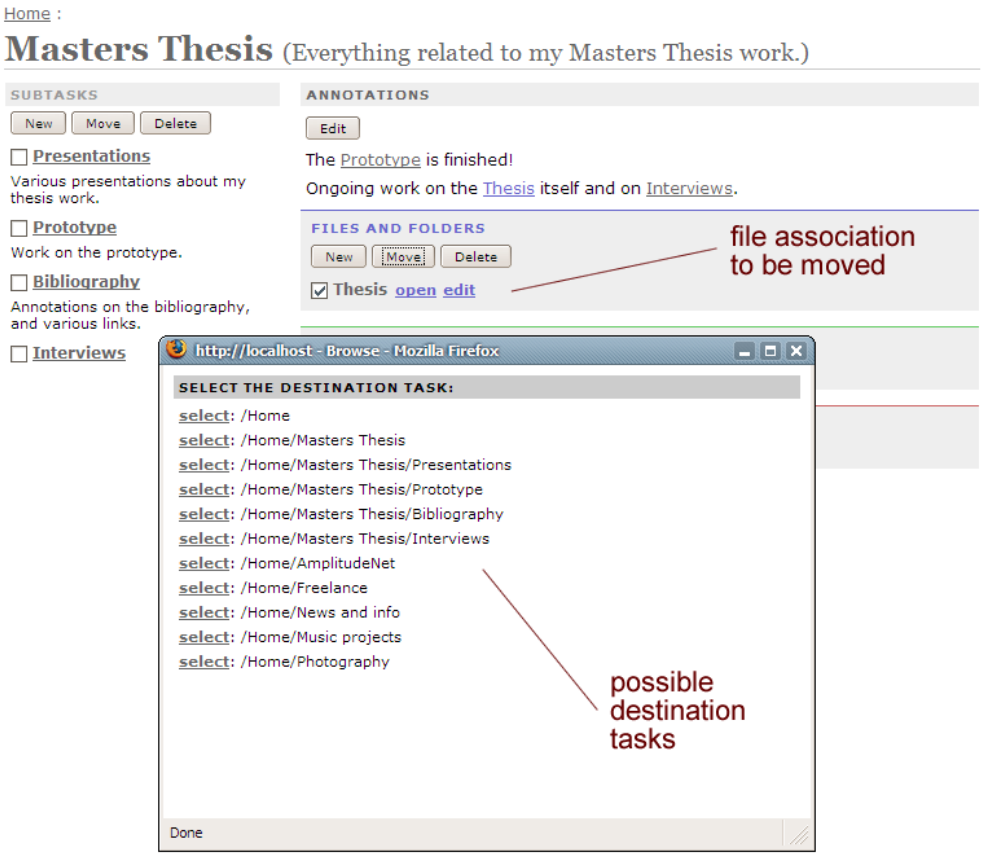

Figure 5.10: Moving a file association

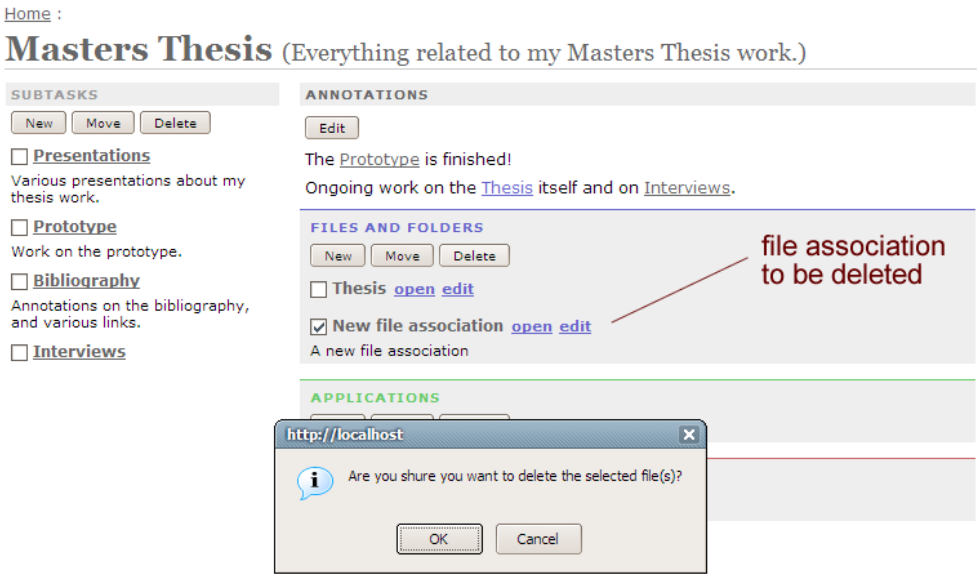

Figure 5.11: Deleting a file association

browsing the file system for a file. This file must be executable and will be called when the 'run' link is clicked.

As usual, moving application links to another task is done by selecting the entries to move, using the boxes on the left of each entry, and clicking on the 'Move' action (figure 5.14).

Deleting application links is done by selecting the entries to delete and clicking on the 'Delete' action (figure 5.15).

5.7 Associating links to a task

Associated Web links are listed in each task and can be followed by clicking on it's name. As the prototype runs on a Web browser, the Web page is immediately showed in a new browser window.

To associate links to a task, the 'New' action is used (figures 5.16 and 5.17).

As on other associations, the user is now prompted for a name and description. An URL is also asked, and will be used as the link URL when the user clicks the name of the association.

Moving links to another task is done by selecting the links to move, using the select boxes, and clicking on the 'Move' action (figure 5.18).

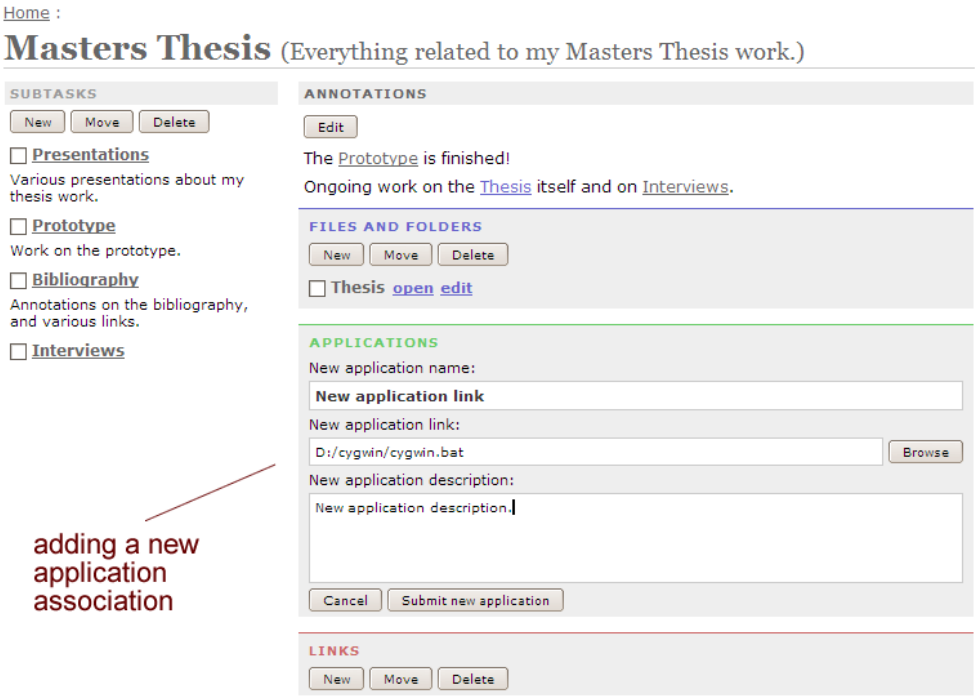

Figure 5.12: Prompt for new associated application attributes

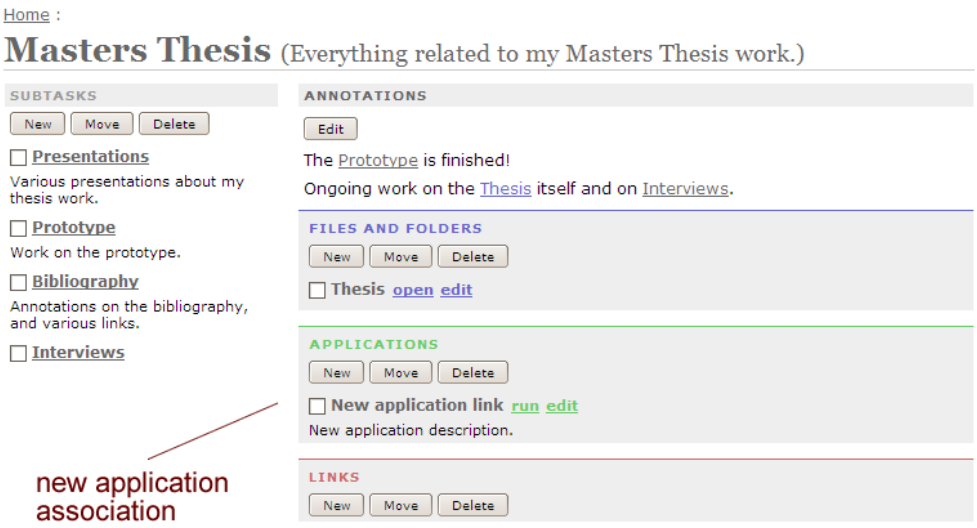

Figure 5.13: New application association inserted

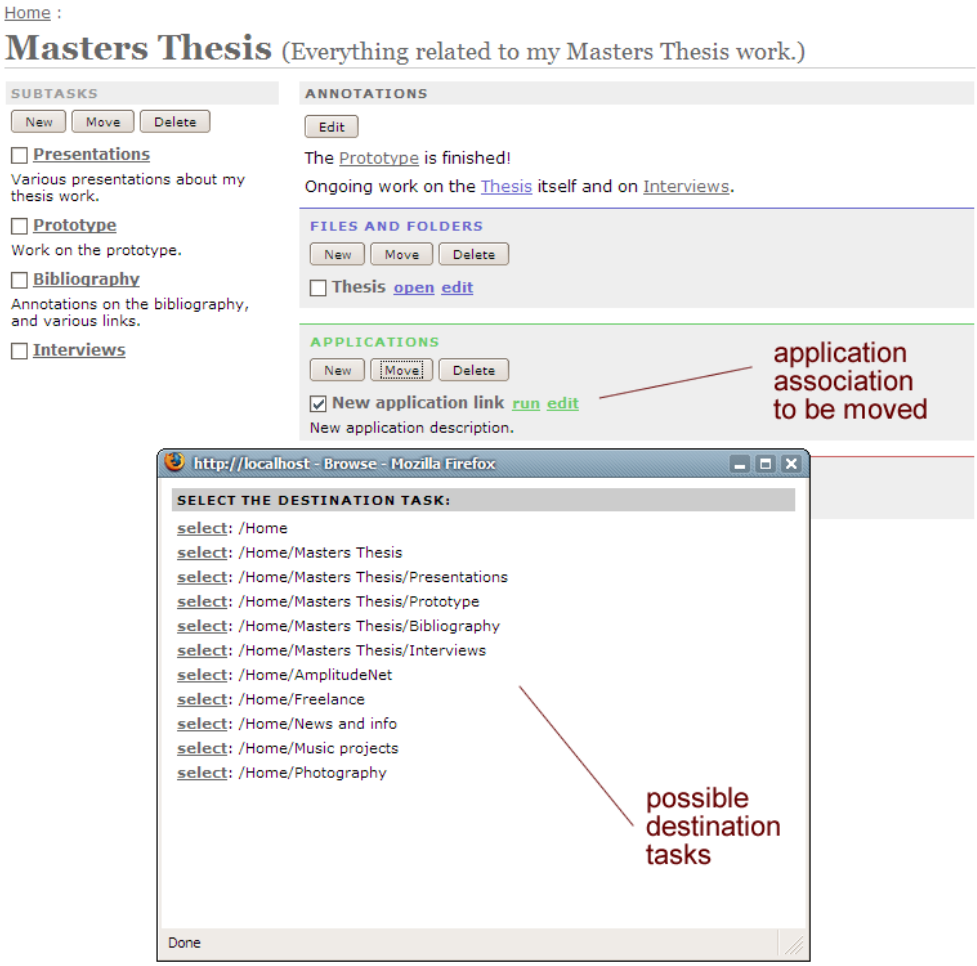

Figure 5.14: Moving an application association

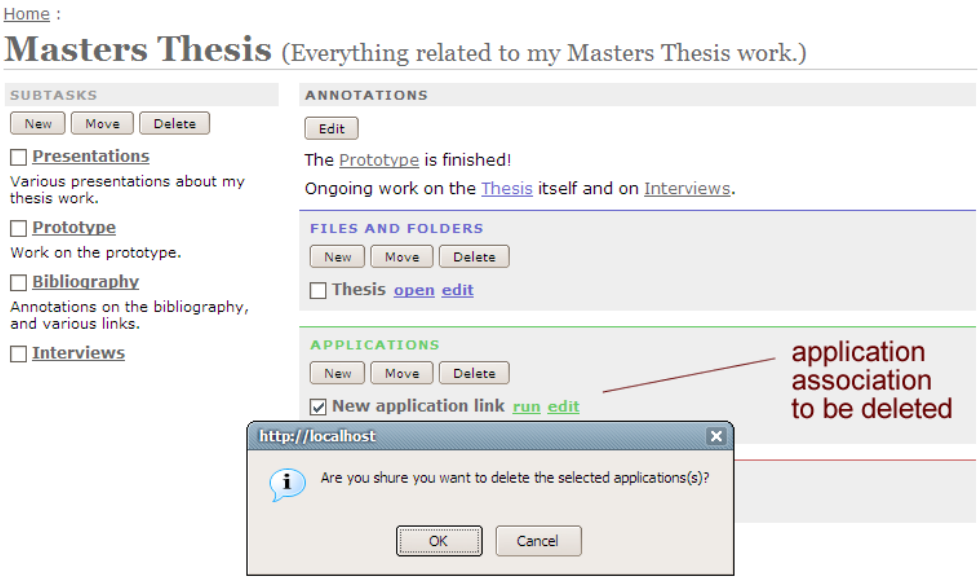

Figure 5.15: Deleting an application association

Deleting the links is done in a similar way, selecting the entries to delete and clicking on the 'Delete' action (figure 5.19).

5.8 Annotations

Annotations are associated to each task, and are edited by clicking on the task 'Edit' action in the annotation area (figure 5.20). This action also permits to alter the task name and task description, which is defined when creating a new task.

Task annotations can include inline links to subtasks, files and folders, applications and links associated to the task. These links can be inserted by writing the name of the element inside square brackets. Emphasis on the annotation text can also be made by surrounding text with asterisks. A single asterisk results in italic text, double asterisks result in bold text, and triple asterisks result in bold italic text.

After submitting the edit, the annotation text is presented in the annotation area of the task.

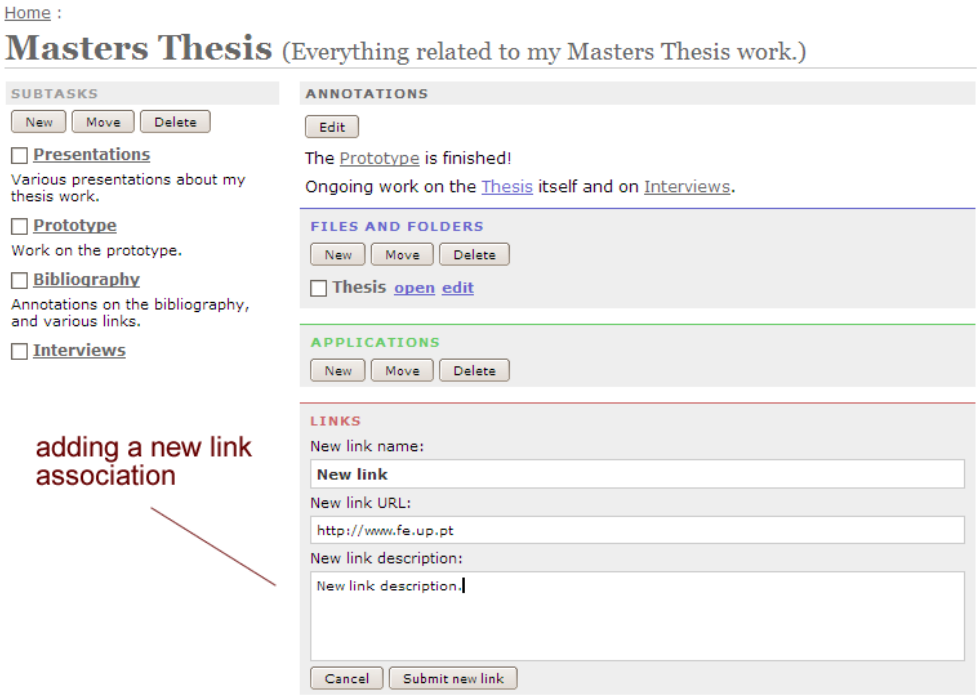

Figure 5.16: Prompt for new link attributes

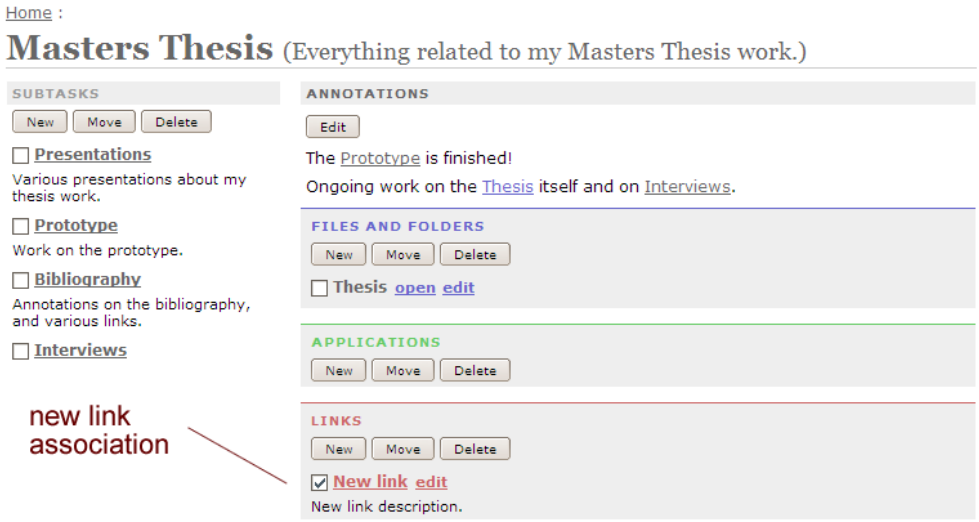

Figure 5.17: New link inserted

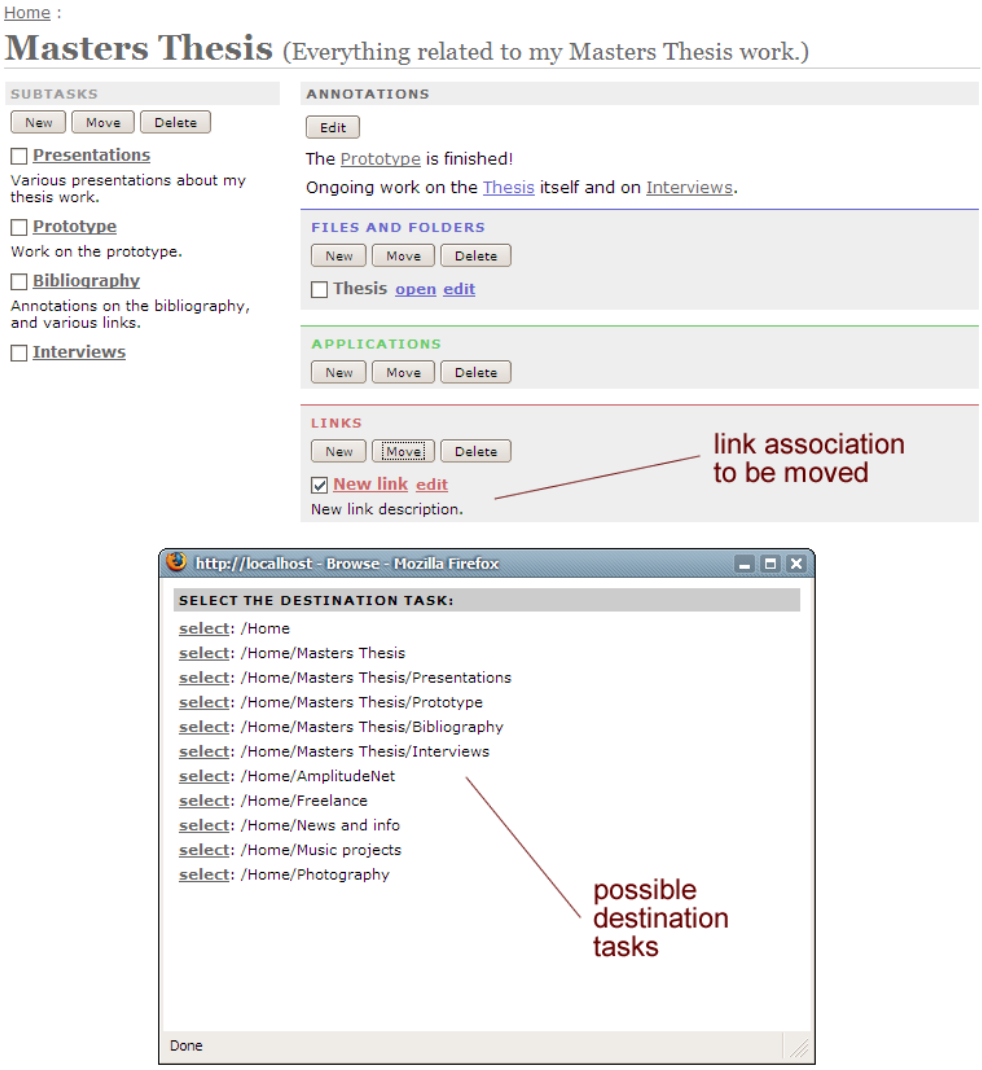

Figure 5.18: Moving a link

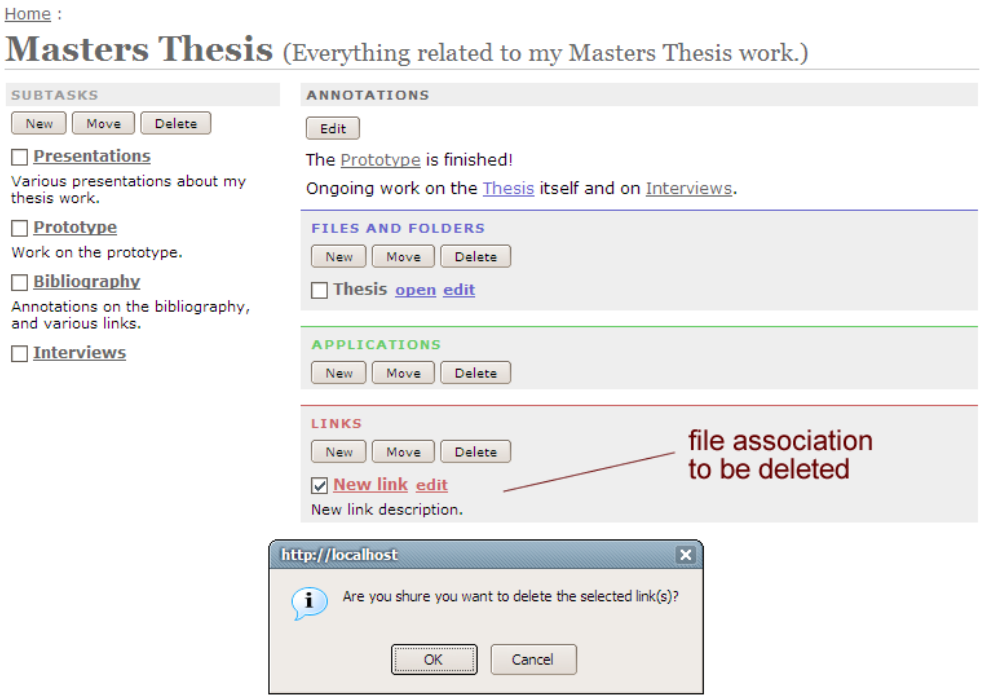

Figure 5.19: Deleting a link

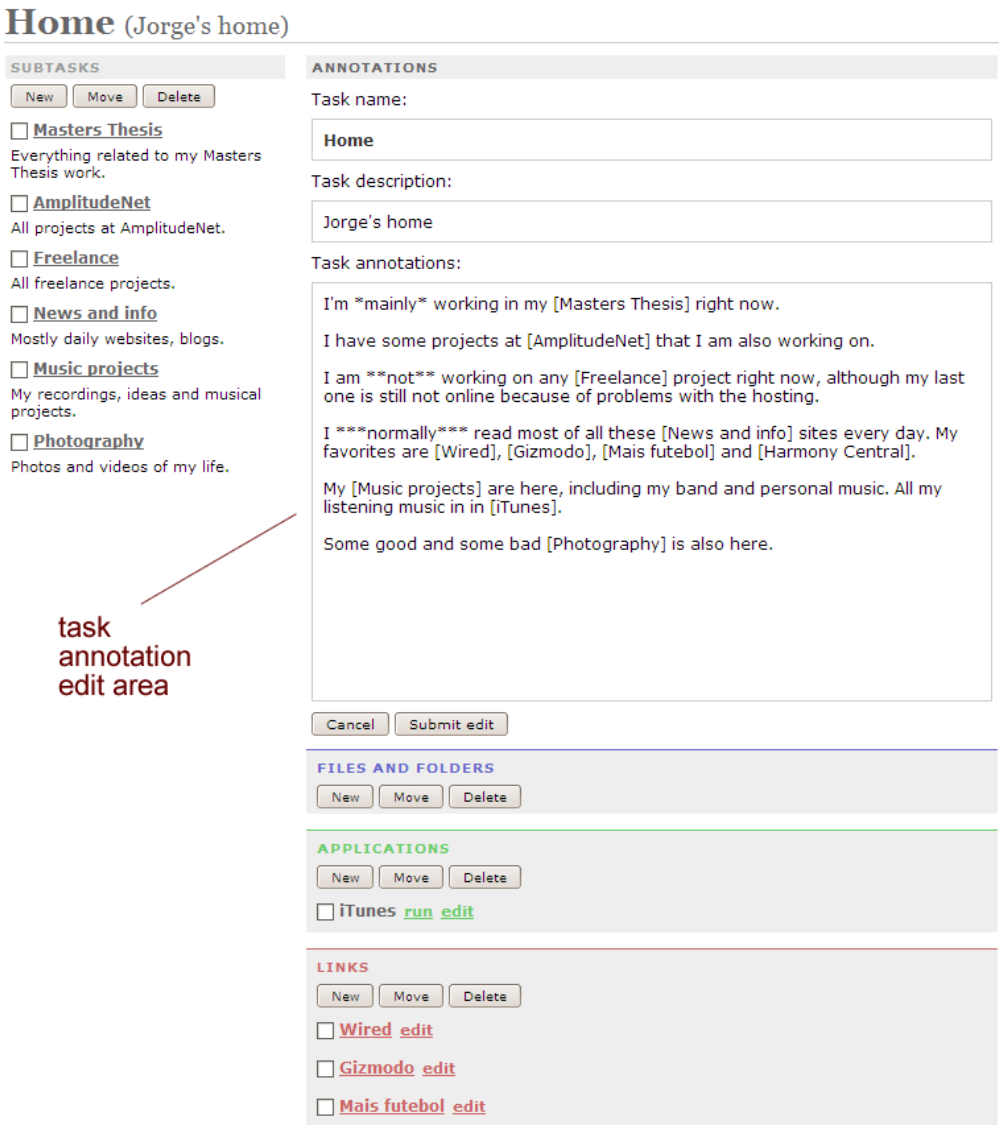

Figure 5.20: Editing annotations and task attributes

5.9 An example of use

To better exemplify the use of the proposed system, a typical use case is presented. The author's work on the present thesis is documented, starting at the 'Home' task and working within the 'Thesis' task.

The 'Home' task contains the following subtasks: 'Masters Thesis', 'AmplitudeNet', 'Freelance', 'News and info', 'Music Projects' and 'Photography'. These are the *big tasks* the author chose to classify his computer use.

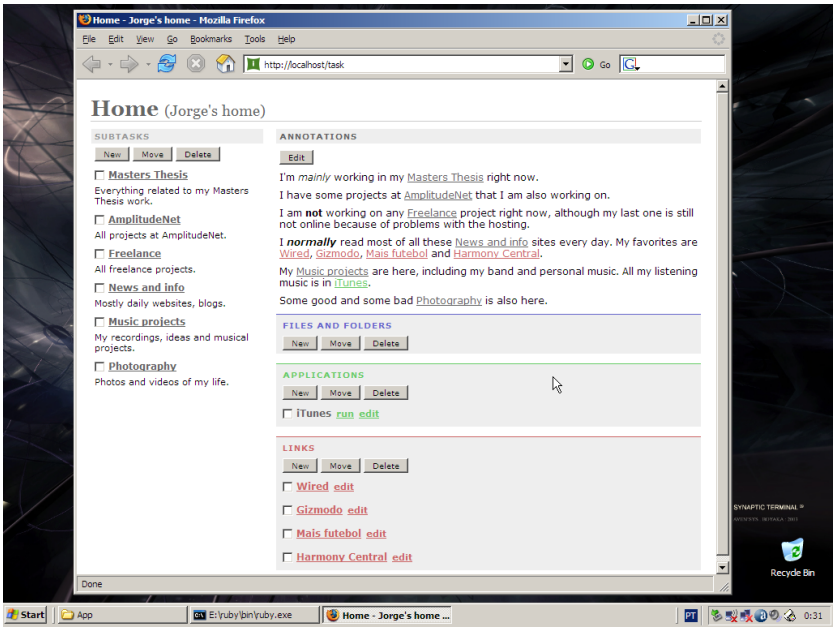

Figure 5.21: Starting work

No files or folders were associated with the 'Home' task, as none are important enough to be in this position. One application, iTunes, is, as it is in use almost all the time, and the author chose to place it in the very first task to have quick access to it. Some links to web sites are also placed in the 'Home' task as they are consulted every morning, when the computer is first used.

The annotations in the 'Home' task serve as a description of the author's computer use, and are used by the author to document what he is currently working on and what is the state of these tasks. It is also very useful when another person uses his computer: it is much easier to find important information following this text.

Clicking the 'Masters Thesis' subtask, the following subtasks are listed:

'Presentations', 'Prototype', 'Bibliography', 'Interviews' and 'Tests' (figure 5.22). These subtasks are part of the 'Masters Thesis' task, and represent either the *smaller tasks* that have to be accomplished to finish the *bigger task* or simply ongoing tasks that serve mainly as an information repository.

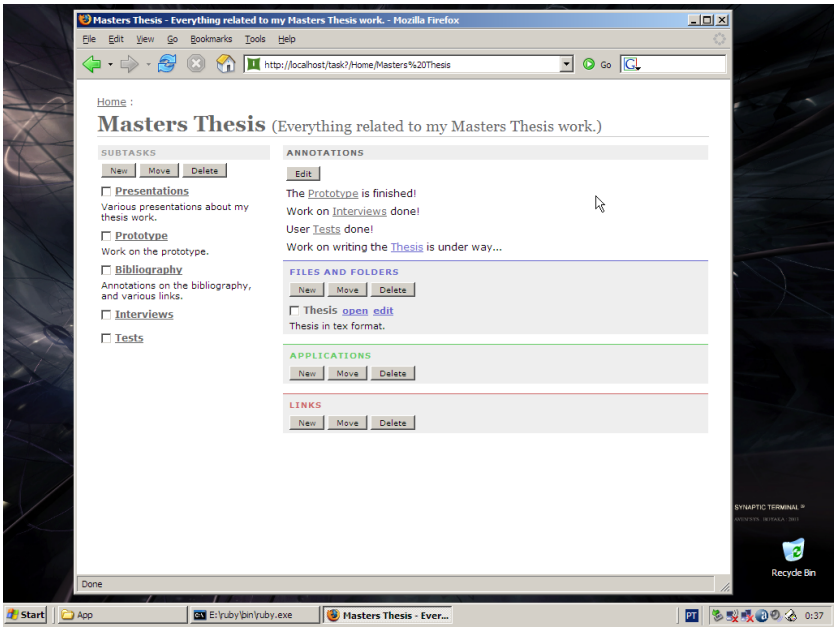

Figure 5.22: The thesis task

The annotations associated with this task are used by the author to document the evolution of his work. This task is now almost completed, only waiting for the completion of writing the Thesis itself.

The text for the thesis is associated to this task in the form of a file in TeX format (figure 5.23). Clicking on the open link next to its name makes the file open with the application that the operating system associates with this file type. In this case, the GVIM application opens the tese.tex file.

The author can then work on this file, and can also add info to the 'Masters Thesis' task, witch is still accessible in the background.

While working, the author maintains some links associated to this task (that can later be moved to the 'Bibliography' task) that are important to the current state of the task. Here, a link to a web page about Paul Otlet, with a description, is listed (figure 5.24).

Clicking on this link opens the web page, enabling fast interaction and information retrieval (figure 5.25).

As normally happens while browsing a web page, links are discovered to

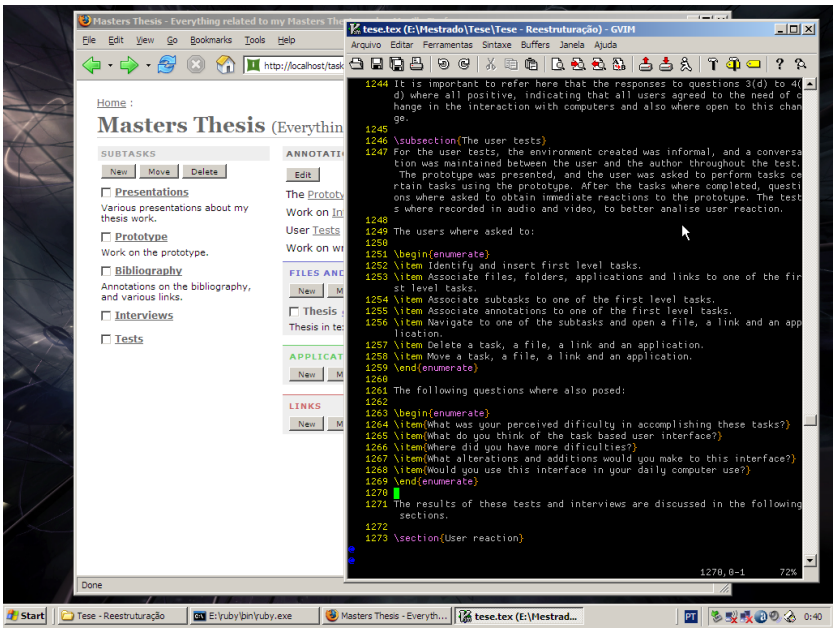

Figure 5.24: Following a link

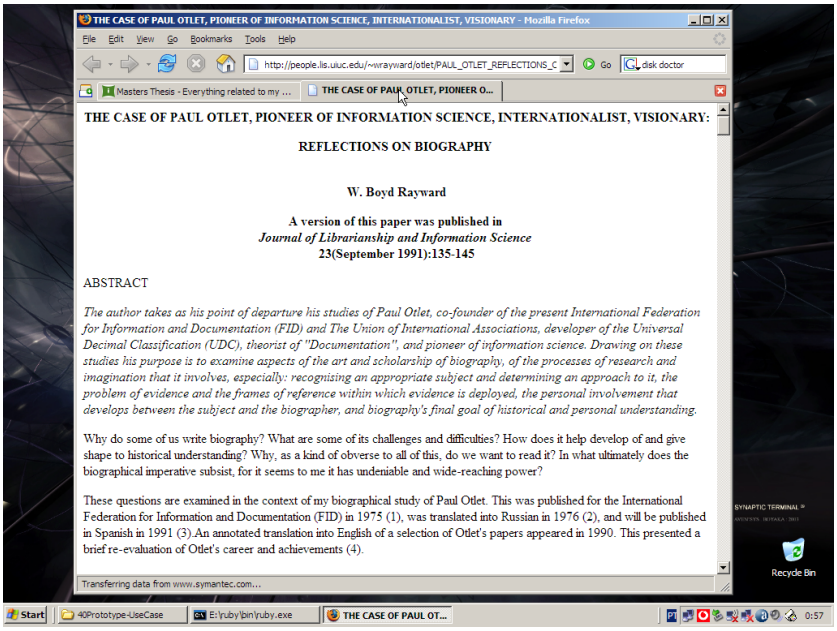

Figure 5.25: Viewing the web page

other relevant pages. In this case, another text present in the same web site is discovered (figure 5.26).

After reading the introduction of this new web page, the author chooses not to read all of it now, and associate this link to the 'Masters Thesis' task, so he can read all of it later.

To associate this new link, the author copies and pastes the name of the web page, its URL and description to the system (figure 5.27).

The result is the creation of a new entry on the 'Links' section of the 'Masters Thesis' task (figure 5.28). This new link will remind the author that he has not read the web page yet. After the author has analyzed this web page, he will move it to the 'Bibliography' subtask, if the information is important, or delete it if it is not.

The author chooses also to place a new annotation in the 'Masters Thesis' task, as he finds this new info to be of importance. He opens the annotation area and inserts the text, 'Just found new info on Otlet: [Visions of Xanadu]!' (figure 5.29).

This annotation gives even more visibility to the new link, and also serves as a log of the work the author has done. When he opens his computer to work on the thesis, he will see that the last thing of relevance he has done was finding this new information.

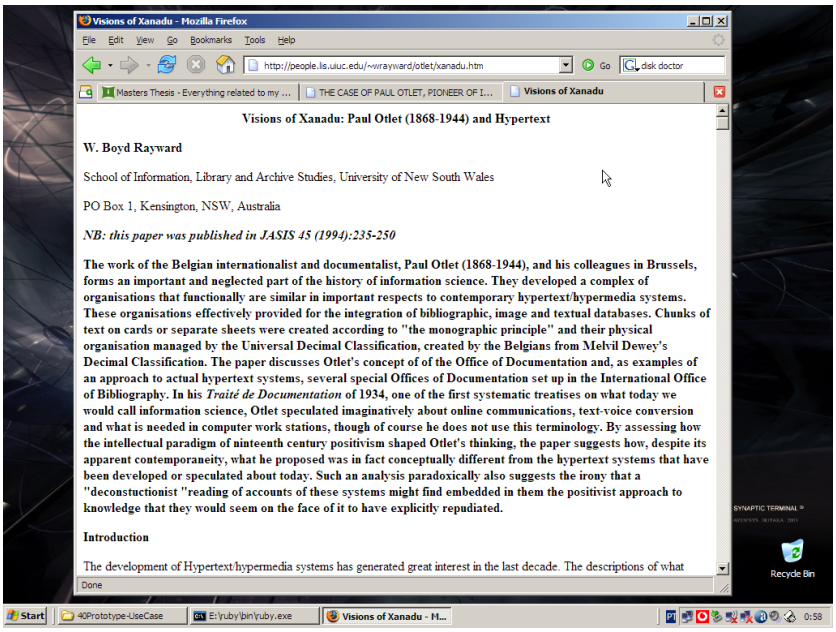

Figure 5.26: Finding a new link of interest

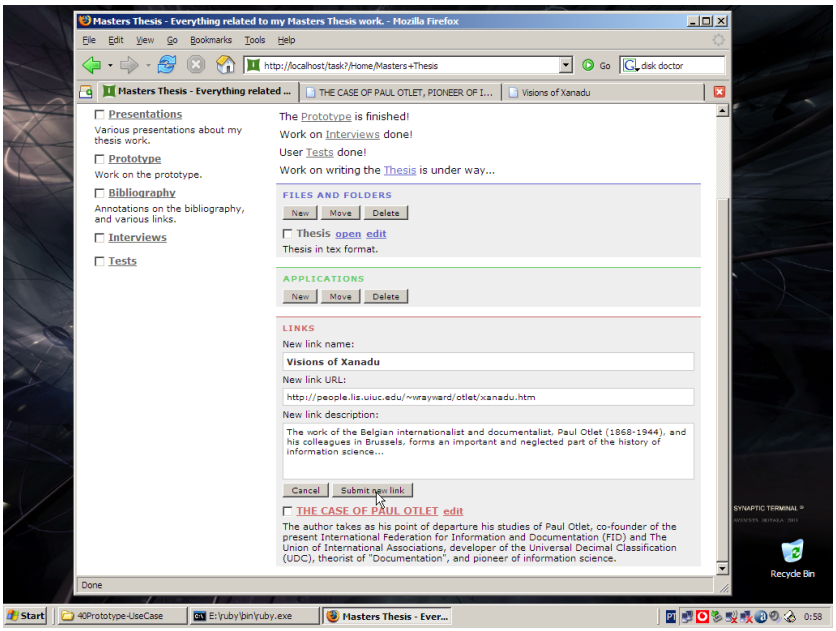

Figure 5.27: Adding a new link to the thesis task

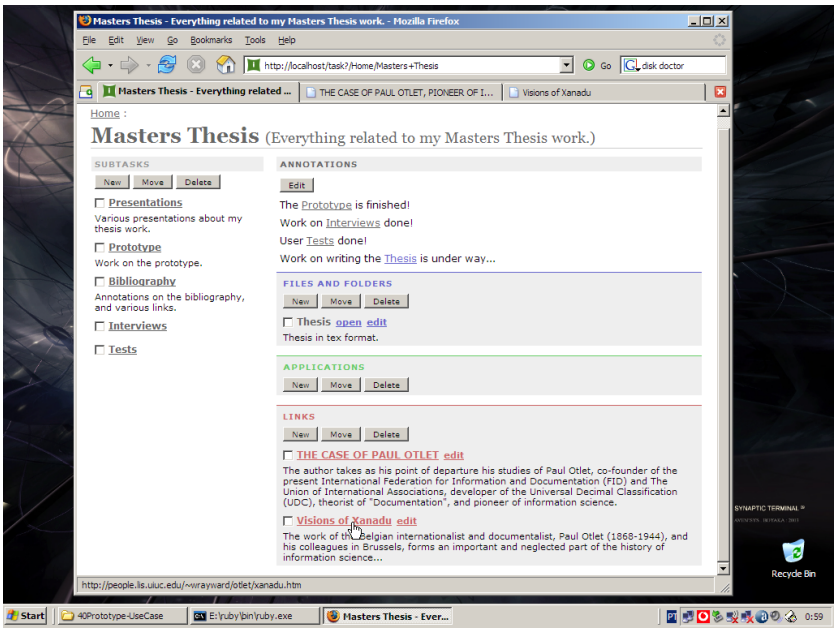

Figure 5.28: New link added

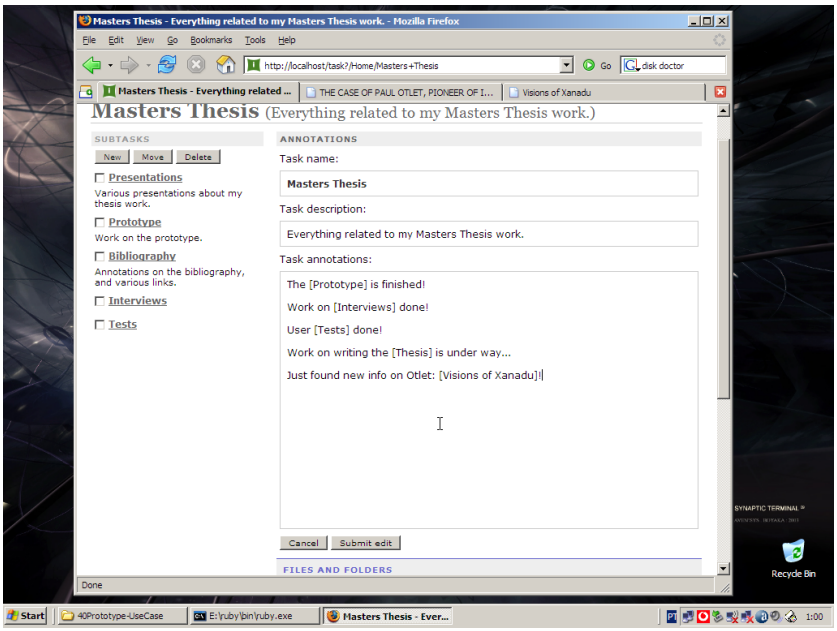

Figure 5.29: Adding a new comment to the thesis task

In the edited task (figure 5.30), this new text is inserted, and the name Visions of Xanadu is identified as an associated link to this task, creating an inline link to the web page, for easier access.

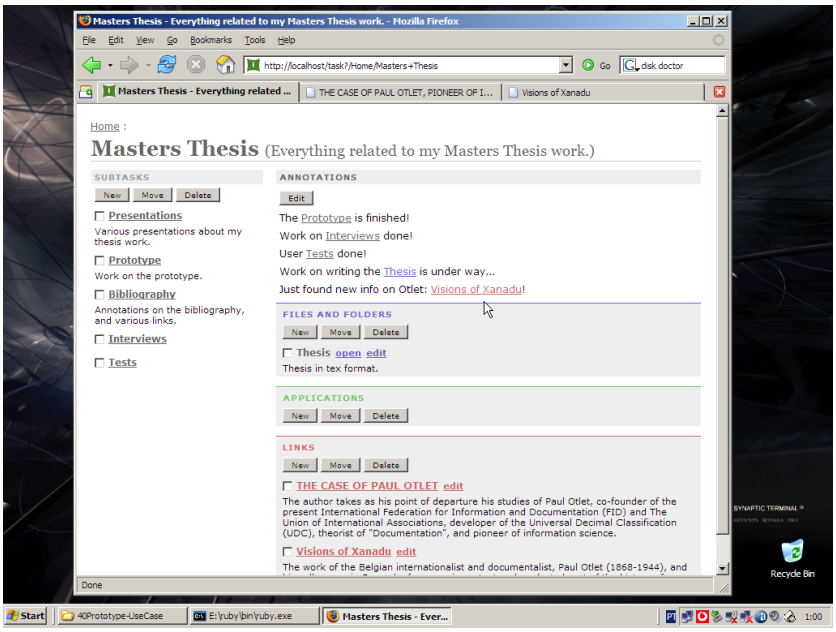

Figure 5.30: New comment added

As shown by this simple example, the daily computer work can be simplified, better documented and more structured with the use of the proposed system. Although many of its functionalities can be replicated in the Windows XP operating system the author uses, the proposed system integrates them in one simple to use interface, making it easy to keep track of work and simple to document it.

If the Windows XP operating system was used to replicate this use case, the author would have to:

- Create a folder structure to hold each task;
- Create links in the corresponding folder to associate the files he uses;
- Create links to web pages;
- Create a text file in each task to enable annotations.

Some functionalities are simply not available using the Windows folder structure:

- Associate descriptions to folders;
- Associate descriptions to files;
- Automatically create links on annotation text.

5.10 User interviews and tests

To have a sense of user reaction to the proposed system, six users were selected to participate on tests and interviews. These tests and interviews aimed to gather ideas and have a sounding board to validate the author's view of both the problems with current operating system user interfaces and the proposed solution.

All users were computer literate and worked with computers on a regular basis. This homogeneous group of users was chosen due to the low number of users available, and allowed a relatively rich insight into the most common computer user, which uses the computer at work or at home and uses common applications like word processing, spreadsheet and Internet access.

5.10.1 The interview

A first interview was conducted, before work was begun in the prototype, destined to get more information about the user and its computer use, as well as the openness to a change in the operating system user interface. These interviews were recorded in audio, to better analyze the responses. The questions posed were the following:

1. Computer Use
   (a) How much do you use computers?
   (b) Where do you normally use them?
   (c) What kind of computer do you use?
   (d) What are the tasks you normally use computers for?
   (e) What operating system do you normally use?
2. Operating System Use
   (a) Describe the way you organize your desktop.
   (b) Describe the way you organize your start menu or dock.
   (c) Describe the way you organize your folder structure.

(d) Describe the way you organize your bookmarks.

3. About Operating System User Interface Problems

   (a) What is your perception in relation the your computer's organization?

   (b) Do you feel in control of the information stored in your computer?

   (c) What do 'folders', 'files' and 'desktop' mean to you?

   (d) Do you feel the need to adjust your way of thinking when you start your work with the computer?

   (e) Do you need to navigate different structures (files, folders, start menu and bookmarks), when you work in one task?

   (f) Do you feel the need to take notes when you use the computer?

4. About A Change In Operating System User Interface

   (a) Would you accept a change in operating system user interface if it reflected in greater ease of use?

   (b) Would you find interesting to build and work within a task structure that reflected your computer use?

   (c) Do you think that this task structure should group files, folders, applications and bookmarks?

   (d) Would you find it useful if the system induced you to organize, comment and annotate your computer use?

The results of these interviews were very encouraging to the author, confirming the relevance and usefulness of this work. All the users noted, in varying levels, a feeling of uncertainty and lack of control while working with their computers.

It is important to refer here that the responses to questions 3(d) to 4(d) were all positive, indicating that all users agreed to the need of change in the interaction with computers and were also open to this change.

5.10.2 The user tests

For the user tests, the environment created was informal, and a conversation was maintained between the user and the author throughout the test. The prototype was presented, and the user was asked to perform tasks certain tasks using the prototype. After the tasks were completed, questions

were asked to obtain immediate reactions to the prototype. The tests were recorded in audio and video, to better analyze user reaction.

One decision made by the author was to give higher importance to casual conversation and the gathering of difficulties and suggestions than to timing the user's actions. Because of the low number of users available, this approach gives more information than more rigid testing. Has can be confirmed in the following section, these conversations were the genesis of some interesting insights into the user's rationale and also ideas for the evolution of the proposed system.

The users were asked to:

1. Identify and insert first level tasks.
2. Associate files, folders, applications and links to one of the first level tasks.
3. Associate subtasks to one of the first level tasks.
4. Associate annotations to one of the first level tasks.
5. Navigate to one of the subtasks and open a file, a link and an application.
6. Delete a task, a file, a link and an application.
7. Move a task, a file, a link and an application.

The following questions were also posed:

1. What was your perceived difficulty in accomplishing these tasks?
2. What do you think of the task based user interface?
3. Where did you have more difficulties?
4. What alterations and additions would you make to this interface?
5. Would you use this interface in your daily computer use?

The results of these tests and interviews are discussed in the following section.

5.11 User reaction

This prototype was used, as mentioned above, for user testing. Various actions were asked of the test users, always with the intent of simulating how they would use this user interface in their own computers. All common uses were tested, like creation of new tasks, moving subtasks around the task structure, associating files, applications, editing annotations, and many others.

The general response was very positive, with relatively few problems and errors. Generally, users quickly understood the idea behind the task structure, and easily created tasks and managed hierarchies.

After the tests, an interview was conducted to get a sense of what users thought of their first contact with this different way of working. Again, the response was very positive, as most of them were interested in using this user interface, even in this prototype form. Interesting suggestions were made, which are discussed in chapter 6.

Presented below are reflections on the test results, with user comment transcription, as well as a listing of total test times for all the test users.

5.11.1 Learning and using the interface

It was obvious when the usage tests recordings were reviewed that almost all of the users had no problem understanding the user interface. They responded quickly to their requested tasks and did not 'get stuck'. When asked, in the following interview, about the perceived difficulty of the user interface, the answers were: "It's real easy to use", and "It was very similar to my web mail application", for example.

Only one user had more problems. She was not accustomed to web applications and felt a little disoriented in the beginning of the test. In a few minutes, however, she had experimented with the interface and was able to perform almost all of the actions quickly. She said: "It was easy after I got the hang of it. I looks like some Websites I know... I should be easier for people that use the Web more than I do."

5.11.2 The task structure

After using the prototype, most of the users were sure that the task aware model was very useful. Most of them were already interested from the original interviews, and here there was a chance to see if their interest was maintained after working with the prototype.

Their reactions were: “After seeing this work, I am even more convinced. I didn’t know how your idea would be transported to the computer, but now I think this way of working brings many benefits.” Another user said: “It forces you to keep things better organized and could make it easier to work with the computer.”

One user had some reservations: “It seems like a good idea, what I don’t know is if it would stand up to prolonged use, or in complex situations. You should do longer tests, like a few months, so I could see if it can respond well.”

5.11.3 Difficulties

Like any good prototype, this one had some problems. Some were implementation bugs, which were progressively solved, and others were usability problems. These were not very severe, but need future attention. Here are the problems encountered:

Breadcrumb not visible enough Most users did not see the breadcrumb navigation area, and had some problems navigating up the task hierarchy. Almost all used the back button of the browser, and only one searched the interface until he found the breadcrumb links.

Further testing on this user interface element should be done, to allow for a decision. Should it be repositioned, highlighted or just be bigger in size? There is no clear answer now, but it should be relatively easy to find a solution to this problem by conducting further tests on different subjects.

The submit buttons The submit button posed some problems. One user did not perceive the need to click on it to execute an insert action and two other users did not see it in the screen, as it appeared ‘bellow the fold’ and they needed to scroll the screen to execute the action.

Perhaps the ‘Submit’ label is not clear and a different word should be used. ‘OK’ was suggested by the user, as she thought it was easier to understand. As for the position of the button, a duplication of it above the forms, as well as its current position bellow it, should rectify this problem.

The move action The move action represented some problems to two users. The prototype redirects the user to the final position of the moved object after the actual move action. It is not standard behavior, and is possibly best to change it.

One of these two users also thought the pop-up window that allows selection of the final position was not very easy to read. He thought that, if there were many tasks, it would be even more difficult. In this case, a better solution should be found.

Following associated links Files and applications are opened by clicking in the 'open' or 'run' links next to their names. The link was treated a different way, being the link name the hyperlink itself. This was confusing to almost all of the users. All expected to see a 'follow link' link or a 'URL' link, but tried the existing 'name of link' link.

It is clear that the coherence should be maintained and there should be a 'follow link' link, a 'URL' link or some other link, so that the associated link can have the same behavior as the file and the application.

Finding the subtask of a task One of the users did not understand, at first, what were the relationships between the current task, the subtasks and the associated files, applications and links. Only after some navigation did it become clear.

Perhaps there needs to be a revision of font sizes, background colors or relative positions for a better perception of the task structure and associated elements. To this end, more user testing would be necessary, so other options could be tested and their effectiveness confronted.

5.11.4 Daily use

As a final question in the interviews after the tests, the users were asked if they would use this interface in their computers. Some answers were: "Yes, it would make my life easier", "yes, my computer would get more organized" and "yes, it would make computer use simpler for me."

Only one user was not very sure but he said: "For me, I don't know. But for most of the users in my office, it would be very good. They struggle with the computer and this way, they would clearly see their tasks and the files. I think it would help them."

5.11.5 User test times

The user test were timed, averaging around 20 minutes each. These times varied not only because of the user, but also due to length of the conversations that happened during the tests. Nevertheless, it is interesting to analyze the results.

Total test times for the users were:

- User 1: 20 minutes
- User 2: 15 minutes
- User 3: 15 minutes
- User 4: 13 minutes
- User 5: 24 minutes
- User 6: 20 minutes

As already explained above, the tests were done in a conversational manner between the author and the users, and because of this some tests ran longer than others. As one of the author's objective was to gather opinions and reactions as the tests progressed, conversations started mid-test about specific difficulties or suggestions.

A pattern is, however, apparent: the faster users were those who used the web more. These users took almost no time to understand the way the proposed system worked and had very little problems in performing the actions.

5.11.6 Overview

The overall user response to this prototype was very favorable. There was a general feeling of effortlessness in user testing, and the task model of the user interface was well received and understood by the users. They were specially interested in the possibility of grouping files, applications and links to a central element, the task, and with the annotating and description facilities.

The user testing was also a source of ideas for additions to the proposed user interface. These ideas are presented on the next chapter, along with other future developments based on this work.

Chapter 6

Future developments

Much work has yet to be done, building on the presented model and prototype, to achieve the goal of having a good alternative to current operating system user interfaces. From more user testing to complete integration into one operating system, there are many possible developments ahead. Some of them are now described.

6.1 Long duration and more rigorous tests

User involvement and testing has been very important in this work, and is indispensable in the future. One type of test that seems essential at this stage are more rigorous and longer duration tests.

One can only have a genuinely clear view of user acceptance when this user interface proposal is placed in the role it was designed for. So, more tests should be conducted, allowing test users to work in a daily basis with a prototype. Careful selection of user groups and user types should be made, allowing a better a better view of different necessities, and periodical interviews would then give insight into each test user's experience and opinion.

6.2 Search

As all the task structure data, including annotation and descriptions is gathered and recorded by the prototype, it is relatively straightforward to implement a search function. This functionality could be very important, specially in large task structures, making access easier to tasks and other elements placed in deep positions of this structure.

The development of search engines is not to be taken lightly[48], as can be verified by poor examples in many web sites. Care has to be taken in

the design of the search user interface and in the implementation of search algorithms. Information relevance, ranking of results, and the definition of what can be searched and what should not be searched are some of the questions that have to be studied to implement a good search engine.

6.3 Integration of email

One of the ideas suggested in the user interviews is integration of email. It could be possible to associate certain email threads, contacts or subjects to a task. By using algorithms similar to those used currently to find spam, email messages can be automatically associated to a determinate task.

This development appears to be of some complexity, but its benefits seem to be able to compensate the effort.

6.4 Integration of task management

Another one of the ideas from the interviews is the integration of task management. After seeing the annotation facilities of the prototype, it seemed to one of the users that the interface could be further improved by adding management facilities for the tasks.

The user could optionally assign to each task start dates, due dates and status, forming a simple task management scheme that in many cases could avoid the use of a separate application. It is not intended to transform this user interface into a complex project management program, but to give some management facilities for everyday tasks or simple projects.

6.5 Version management

File version management is possible to integrate in this user interface in a relatively transparent way. A file associated with a task has a name chosen by the user, as well as a description, that are completely independent of the real file name, as identified by the file system. Various versions of a file, corresponding to different files in the file system, can be grouped under this user chosen file name, thus forming the base for an integrated version management system.

Surely many implementation and design problems will surface, but it seems possible to add useful version management facilities in a way that would help and not make the user interface too complex.

6.6 Backup

Information backup is a usually a problem for the common user. With information spread throughout the file system, paper notes spread on the (real) desktop, it is difficult to be sure that the required information is safely stored.

A backup facility can be integrated in the proposed system, making use of the task structure, which would allow the user to backup all the system's information or just some tasks. This backup facility would make a copy of the selected tasks, and all information associated to it, including subtasks and associated files or folders. Even if the files are scattered, this action would group them all in one backup file. The restore facility would read this file and populate the task structure, and place the files in their original place in the file system.

6.7 Solutions for work groups

Although intended for personal use, the proposed user interface could be adapted for work groups. Two different solutions are possible: one is to use a centralized task structure, allowing access to multiple users, and the other is to use one private task structure for each user and allow all users to view a specified section of them.

Each of these solutions has its use and is better suited for distinct situations. It seems clear, however, that both could be very useful in a work group context.

6.8 Integration with an operating system

The obvious conclusion of this work would be to integrate this user interface proposal with an operating system, substituting the common desktop metaphor based user interface. Although this a complex and time consuming effort, it is nonetheless the author's goal.

A prime candidate for this work would be Linux, which is currently served by the KDE and GNOME desktop environments, both derived from MacOS and Windows. It is interesting to note that there are few desktop environments that don't use the desktop metaphor, and those who do not use it focus on the command line, and are intended for UNIX-savvy users.

This is an opportunity to build an alternative user interface, intended for general use and designed to answer current computer user's needs, using stable and powerful UNIX backbone.

Chapter 7

Conclusions

Based on the observation of users and their difficulties using current operating systems, this work proposed an alternative model of interaction, with the intention of starting work on a simpler, more usable user interface which responds to current computer users' needs.

The model proposed is based on the user's own view of the tasks for which the computer is used. He is responsible for identifying these tasks and associating information to them. Files, applications, hyperlinks and annotations are all elements that can be associated to tasks, and do not have their own separate structures.

This model works as a layer above the current operating system, serving as a task aware user interface that uses the operating system facilities to open files, applications, hyperlinks and annotations.

During the historical research for this work, it became evident that the evolution of the user interface was influenced by many factors not directly related to the quality of the user interface itself, such as hardware prices, marketing and corporate decisions.

The latest computer revolution, the World Wide Web, is based on open source technologies and this opened the door to many excellent professionals, allowing them to offer users new interaction experiences and raise expectations as to what users expects from daily computer use.

It is at this time that we look at current operating system user interfaces and see that they have not changed significantly in twenty or thirty years. Many great ideas have been forgotten over the years and almost all of us use daily a user interface model that was not considered the best at the time it was first presented.

Because of this, the theoretical options behind the proposed model are mainly based on the ideas of user interface pioneers and early graphical user

interfaces, and the technical options are based on recent web technologies.

Through years of informal conversations, the problems of current operating system user interfaces have become more and more evident to the author, especially as users become more accustomed to browsing the Web or working with intranets.

The interviews carried out during this work served to confirm some of these problems and showed that users are open to a change in the way they work with their computers. It is our task to offer this change to them.

The developed prototype showed, in user testing, that the proposed model can be easily understood by common users, and that a browser-based user interface following this model can be successfully used as a communication interface between user and computer.

The author chose to conduct user interviews and tests with an informal and conversational approach, not to prove that the proposed system was better than the operating system user interface they used, but as a way to gather the maximum amount of informations as he confronted the users with his ideas and the proposed system prototype.

The comprehensive description of the proposed system presented in this work, as well as the use case, aims to show the benefits of using a task aware user interface as an alternative to common operating system graphical user interfaces. Although it is possible to mimic some of the basic features of the proposed system on current operating systems based on folder structures, the complexity and overhead of it means almost no user choses to do so.

Much more work has to be done, especially with users, so that the ideas presented in this work can be offered to the public. The integration of this model with an operating system, replacing current user interfaces, would be the culmination of this work. With various open source operating systems available presently, this job seems a lot easier than it would have been some years ago.

Bibliography

[1] Wikipedia, *History of the graphical user interface*. Wikipedia, http://www.wikipedia.com, August 20, 2005.

[2] Wikipedia, *Paul Otlet*. Wikipedia, http://www.wikipedia.com, May 3, 2005.

[3] Alex Wright, *Forgotten Forefather: Paul Otlet*. Boxes and Arrows, http://www.boxesandarrows.com, November 10, 2003.

[4] W. Boyd Rayward, *The Case of Paul Otlet, Pioneer of Information Science, Internationalist, Visionary: Reflections on Biography*. Journal of Librarianship and Information Science 23, September 1991.

[5] Wikipedia, *Vannevar Bush*. Wikipedia, http://www.wikipedia.com, August 14, 2005.

[6] Wikipedia, *Memex*. Wikipedia, http://www.wikipedia.com, August 10, 2005.

[7] Vannevar Bush, *As We May Think*. The Atlantic Monthly, July 1945.

[8] Wikipedia, *Ivan Sutherland*. Wikipedia, http://www.wikipedia.com, July 10, 2005.

[9] Wikipedia, *Sketchpad*. Wikipedia, http://www.wikipedia.com, May 24, 2005.

[10] Ivan Sutherland, *Sketchpad: A man-machine graphical communication system*. Proceedings of the AFIPS Spring Joint Computer Conference, Detroit, Michigan, May 21-23, 1963.

[11] Wikipedia, *Douglas Engelbart*. Wikipedia, http://www.wikipedia.com, August 1, 2005.

[12] Wikipedia, *NLS (computer system)*. Wikipedia, http://www.wikipedia.com, August 14, 2005.

[13] Douglas Engelbart, *Augmenting Human Intellect: A Conceptual Framework*. Stanford Research Institute Summary Report, October 1962.

[14] Wikipedia, *Xerox Alto*. Wikipedia, http://www.wikipedia.com, August 5, 2005.

[15] Thomas A. Wadlow, *The Xerox Alto Computer*. Byte, issue 9/1981, September 1981.

[16] Wikipedia, *Xerox Star*. Wikipedia, http://www.wikipedia.com, July 15, 2005.

[17] David Canfield Smith, Charles Irby, Ralph Kimball, Bill Verplank and Eric Harslem, *Designing the Star User Interface*. Byte, issue 4/1982, April, 1982.

[18] David Canfield Smith, Charles Irby, Ralph Kimball and Eric Harslem, *The Star user interface: an overview*. Proceeding of the AFIPS National Computer Conference, 1982.

[19] Eric Harslem and LeRoy E. Nelson, *A retrospective on the development of Star*. Proceedings of the 6th International Conference on Software Engineering, 1982.

[20] Jonathan Seybold, *Xerox's Star*. The Seybold Report, Media, PA: Seybold Publications, vol. 10, no. 16, 1981.

[21] Jeff Johnson, Teresa L. Roberts, William Verplank, David C. Smith, Charles Irby, Marian Beard and Kevin Mackey, *The Xerox 'Star': A Retrospective*. IEEE Computer, September 1989.

[22] Charles Irby, Linda Bergsteinsson, Thomas Moran, William Newman, and Larry Tesler. *A Methodology for User Interface Design*. Systems Development Division, Xerox Corporation, January 1977.

[23] Wikipedia, *Apple Lisa*. Wikipedia, http://www.wikipedia.com, July 29, 2005.

[24] Peter Nulty, *Apple's bid to stay in the big time*. Fortune, February 7, 1983.

[25] Roderick Perkins, Dan Smith Keller and Frank Ludolph, *Inventing the Lisa User Interface*. Interactions, issue 1/1997, 1997.

[26] Frank Ludolph and Roderick Perkins, *The Lisa User Interface*. CHI '98 Conference Summary, 1998.

[27] Wikipedia, *Apple Macintosh*. Wikipedia, www.wikipedia.com, August 15, 2005.

[28] Lon Poole, *A tour of the Mac desktop*. Macworld magazine, issue 1/1984, 1984.

[29] Mark Jennings, *Evaluating the Macintosh Finder* Byte, issue 12/1984, December 1984.

[30] Wikipedia, *Microsoft Windows*. Wikipedia, www.wikipedia.com, August 14, 2005.

[31] Phil Lemmons, *Microsoft Windows*. Byte, issue 12/1983, December 1983.

[32] John Dvorak, *Sorting out fact from fiction in the Apple-Microsoft lawsuit*. PC Magazine, May 31, 1988.

[33] Wikipedia, *NeWS*. Wikipedia, www.wikipedia.com, July 28, 2005.

[34] Wikipedia, *NEXTSTEP*. Wikipedia, www.wikipedia.com, July 26, 2005.

[35] Wikipedia, *Dock (computing)*. Wikipedia, www.wikipedia.com, July 20, 2005.

[36] Wikipedia, *Shelf (computing)*. Wikipedia, www.wikipedia.com, November 13, 2004.

[37] Wikipedia, *OS/2*. Wikipedia, www.wikipedia.com, August 20, 2005.

[38] Nathan Lineback, *OS/2*. GUI Gallery, http://toastytech.com/guis/indexos2.html.

[39] Wikipedia, *X Window System*. Wikipedia, www.wikipedia.com, August 16, 2005.

[40] Dick Pountain, *The X Window System*. Byte, issue 1/1989, January 1989.

[41] Wikipedia, *Motif (widget toolkit)*. Wikipedia, www.wikipedia.com, August 19, 2005.

[42] Wikipedia, *OPEN LOOK*. Wikipedia, www.wikipedia.com, June 9, 2005.

[43] Nathan Lineback, *Solaris 8 Common Desktop Environment and the OpenWindows environment*. GUI Gallery, http://toastytech.com/guis/sol.html.

[44] Wikipedia, *Hypertext*. Wikipedia, www.wikipedia.com, August 5, 2005.

[45] Wikipedia, *World Wide Web*. Wikipedia, www.wikipedia.com, August 21, 2005.

[46] Jesse James Garrett, *Ajax: A New Approach to Web Applications*. Adaptive Path, www.adaptivepath.com, February 18, 2005.

[47] Jakob Nielsen, *Usability Engineering*. Morgan Kaufmann, 1993.

[48] Louis Rosenfeld and Peter Morville, *Information Architecture for the World Wide Web, 2nd Edition*. O'Reilly, August 2002.

[49] P.N. Johnson-Laird, Vittorio Girotto and Paolo Legrenzi, *Mental models: a gentle guide for outsiders*. April 7, 1998.