On User Interfaces for Educational Multimedia Applications

Panagiotis Takis Metaxas[1]

Computer Science Department

Wellesley College

Abstract

This paper addresses two questions related to educational multimedia applications:
- What lessons should we learn from today's Graphical User Interfaces (GUIs)?, and
- What the user interfaces should look like in the future?

We address the first question by drawing on our experience from two multimedia projects, that deal with multimedia proceedings and multimedia lectures. For the second question, we criticize the directions that today's research for 3-D and VR GUIs is going, and we argue that they may not necessarily be on the way to the optimum user interface, which is none, and we will describe an alternative path.

1. Introduction

Multimedia applications for educational purposes have proliferated during the past 10 years. Some of them have been successful, others have not. (Here, by successful we mean the degree that have helped students learn the material that is presented through the applications. We are interested in addressing the issue of how the design of user interface has affected their success and what are the lessons we should learn from them. Clearly, this issue cannot be addressed completely in a single paper, however, we can speak for our own experience and for the experience of some of our colleagues. In particular, this paper addresses two questions:

- What have we learned from designing Graphical User Interfaces (GUIs) for multimedia educational purposes, and

- What the user interfaces should look like in the future?

We address the first question by discussing first the history of GUIs, and then examining in some detail two case studies from our own experience in multimedia educational applications. The first project implemented interactive electronic proceedings while the second used

[1] Email address: pmetaxas@wellesley.edu, URL: http://www.wellesley.edu/CS/pmetaxas.html Supported by NSF grant INT-9418518 for US-Australia Cooperative Research. Part of this work was done while the author was with the Dartmouth's DEVLab and MIT's Laboratory for Computer Science.

multimedia to deliver educational lectures. From these two projects, as well as from projects that other colleagues have undertaken, we derive a set of guidelines that we believe would be useful to everyone working on multimedia applications today.

To answer the second question, first we examine briefly the main directions that user interfaces seem to explore today, namely three-dimensional and virtual reality user interfaces (3-D UIs and VR UIs). We argue, however, that this is not the direction that we may want to go; briefly, the reason is that it is more difficult to manage a messy-room (the canonical metaphor that corresponds to 3-D and VR UIs) than a messy-desktop (the accepted metaphor of 2-D window-based GUIs). Our thesis is that a more natural and efficient user interface is one that involves *sound* in the human-to-computer direction, and the extensive knowledge of *book typesetting* in the computer-to-human direction.

2. A Short History of Graphical User Interfaces

In the early times of computer history, processing was done in a *batch* form. Users had to stand in line and leave their jobs outside the operator's room. The *operator* , a person responsible to feed the computer with data and code was essentially the only person that communicated directly with the computer. Even though it may seem that the operator was the user interface, the fact is that in these early days *there was no notion* of user interface. Communication with the computer was so tedious, both in terms of language and interface, that the average person could not be trusted in this role.

Things changes slightly (even though at the time it seemed as a huge change) when more advanced operating systems made time-sharing possible. Now, the users were able to communicate directly with the computer, through the *command-line interface*. Users, for example, would type unpronounceable lines of the form

```
        cp A:drctns.txt D:idntknw.txt
```

and the machine might either perform the task successfully and give *no response*, or indicate that *something* went wrong and quit. The command line was providing modes of communication that were rather difficult to distinguish, and the user would have to memorize unpronounceable words and cryptic operating system responses.

The *modal interface* was ugly, error-prone and inconvenient, partially responsible for attaching the "geek" characterization to programmers. Yet, it has still to depart from our world because computer users – contrary to popular belief – resists any change to their environments, even when the benefits of the change are obvious.

Things became more rational around 1972, when the famous XEROX PARC team came up with the first *graphical user interface (GUI)* for the XEROX Altos. This interface employed the desktop metaphor on which papers and other documents are laid out. Since the screen of the computer terminals had a rather small area available for this desktop, users were seeing only small windows of the whole document.

Despite its strikingly superior idea of the graphical over the modal interfaces, it took them about 12 years to become known, mainly through Apple's operating system that employed them in the first release of the Macintosh 128K in 1984. Their popularity grew slowly, however, partly due to of a wide-spread macho feeling among programmers that associates power with intimidation.

Today, there is no doubt that the *messy-desktop* interface — which so closely resembles real life's desktops — has been extremely successful. As a measure of its success, we mention the fact that 23 years after its introduction, a software company stirred lots of excitement – and revenues – by releasing an operating system with such a graphical user interface. One has to admit that an idea that excites people so many years after it was first presented, must be a *really* good idea, indeed!

GUIs helped the proliferation of users because computers were not intimidating any more. Even kids that could not read or write, could use a computer, something unthinkable before. People realized that *usability* was much more important than functionality – over some minimum threshold of functionality, of course. Ed Tufte's 1988 aphorism "Today the competition is at the User Interface" [1] still holds true.

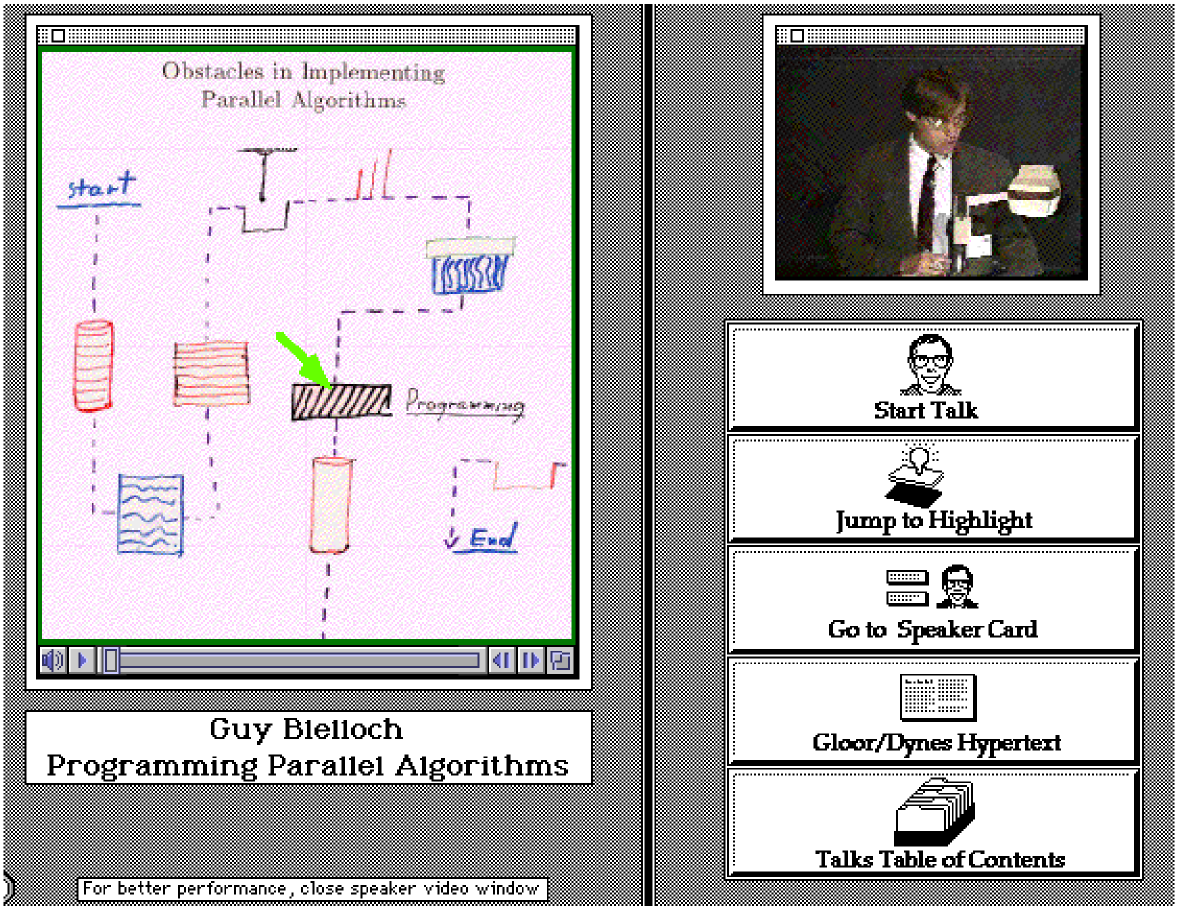
Unfortunately, the field of user interface design has not advanced significantly since. Many people believe that the advancement will come from real 3-D and visual reality interfaces. However, as we will argue in a later section, the incorporation of sound into a well-designed 2-D panel would serve the user better.

In the next three sections we describe briefly the experience we gained from two educational multimedia applications that we have built with colleagues, and will derive some guidelines from our own and other colleagues' experiences.

3. Case Study 1: Electronic Conference Proceedings

Academic conferences are a long-standing and effective form of multimedia communication. Conference participants can transmit and receive information through sight and sound, that is, by viewing individuals, text, and graphics, and by hearing the spoken word. This same-time, same-place communication is sufficiently valuable to justify large investments in time and travel funds. Printed conference proceedings, a particular kind of printed books, are attempts to recapture the value of a live conference, but they are limited by both their delivery medium and by the significant differences from the conference presentation. We addressed this problem in the CD-ROM multimedia proceedings[2] of the DAGS'92 conference[3], that delivers text, graphic, audio, and video information as an integrated whole, with extensive provisions for random access and hypermedia linking. (For details on this production, see [4].)

This program uses two basic screens to communicate with the reader of the proceedings, the talk screen (Figure 1) and the hypertext screen (Figure 2). The figure below shows a typical shot of the talk screen, which is used to help the reader follow dynamically the talk, and more efficiently than if he/she were present at the time of the presentation. The movie on the left is made out of the transparencies that the speaker used, while the one on the right is a short loop of the speaker speaking. The buttons start and stop the transparencies talk, give direct access to particularly interesting points of the talk, give information about the speaker, jump to the hypertext of the paper that is associated with the talk, and go to the list of all talks that appear in the CD-ROM.



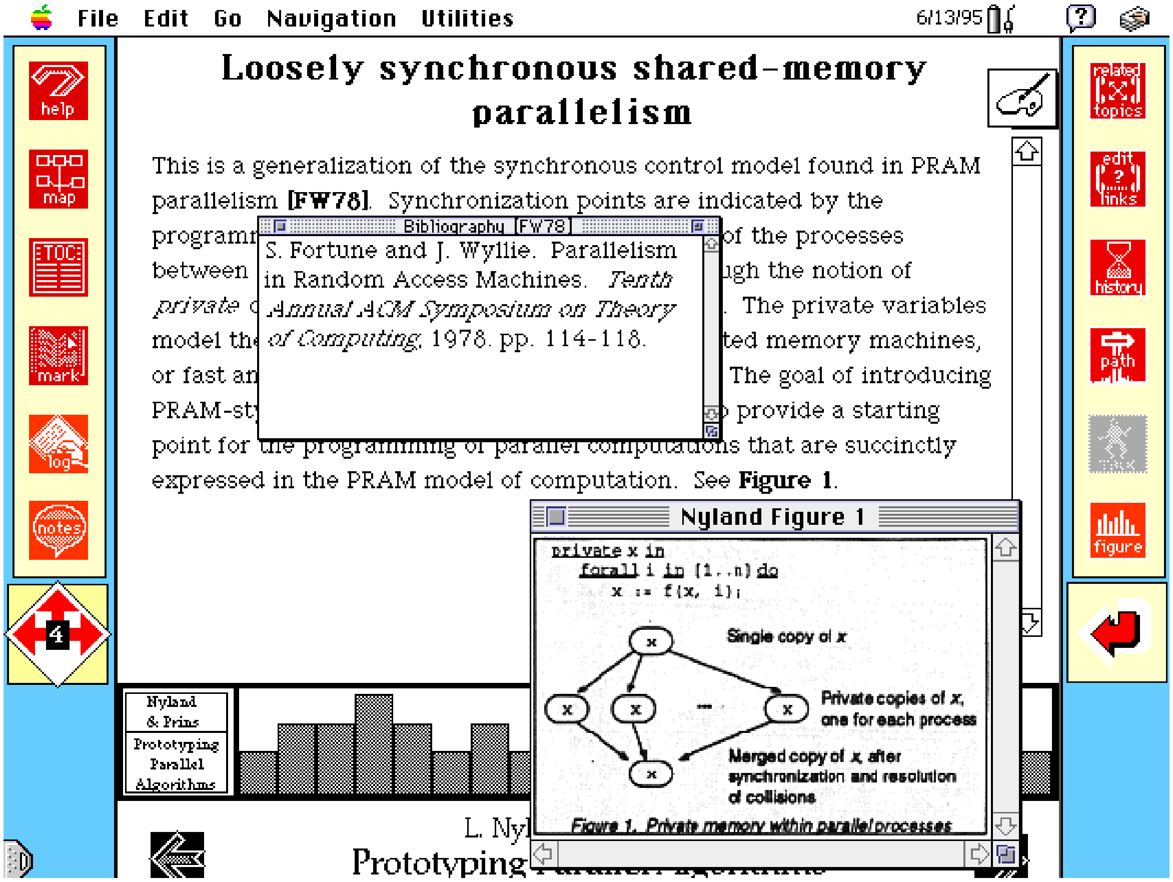
Guy Blelloch
Programming Parallel Algorithms

For better performance, close speaker video window

Fig. 1: The talk screen of [2]. Only 36% of the total screen area contains useful information.

Figure 2 represents a typical shot of the hypertext that is associated with the talk. It contains two palettes, on the left and right of the screen, with tools that give considerable power to the reader to access the contents of the paper in many different ways. Floating windows show references and figures on demand.

During the design of the system, it was believed that giving as much power as possible to the reader/user was a highly desirable attribute of a successful system. "The readers could use the system as naively as they would use paper proceedings, or as advanced as a full-fledged multimedia system", was the prevailing view of the designers. Obviously, designing such a system was not an easy task, taking the better part of a man-year production time, but at the time was considered worth the effort.



Loosely synchronous shared-memory parallelism

This is a generalization of the synchronous control model found in PRAM parallelism **[FW78]**. Synchronization points are indicated by the programm of the processes between ugh the notion of *private* c . The private variables model th ted memory machines, or fast an The goal of introducing PRAM-sty provide a starting point for the programming of parallel computations that are succinctly expressed in the PRAM model of computation. See **Figure 1**.

Bibliography [FW78]

S. Fortune and J. Wyllie. Parallelism in Random Access Machines. *Tenth Annual ACM Symposium on Theory of Computing*, 1978. pp. 114-118.

Nyland Figure 1

```
private x in
  forall i in [1..n] do
    x := f(x, i);
```

x
Single copy of x
x
x
...
x
Private copies of x, one for each process
x
Merged copy of x, after synchronization and resolution of collisions

Figure 1. Private memory within parallel processes

Fig. 2: The hypertext screen of [2]. The administrative palettes on the sides give power to the user to access the hypertext in many different ways.

What it was not realized, however, was that:

- A complicated interface, no matter how powerful, is *intimidating* to the readers. Unsophisticated readers would not use it, while most of the readers would just use only the 3–4 of its features that seem natural to them ignoring the rest, and

- A baroque interface takes away valuable computer screen area to accommodate administrative debris. As a consequence, one is forced to use scrolling fields, lowering significantly the already low (because of the mathematically-rich contents) readability.

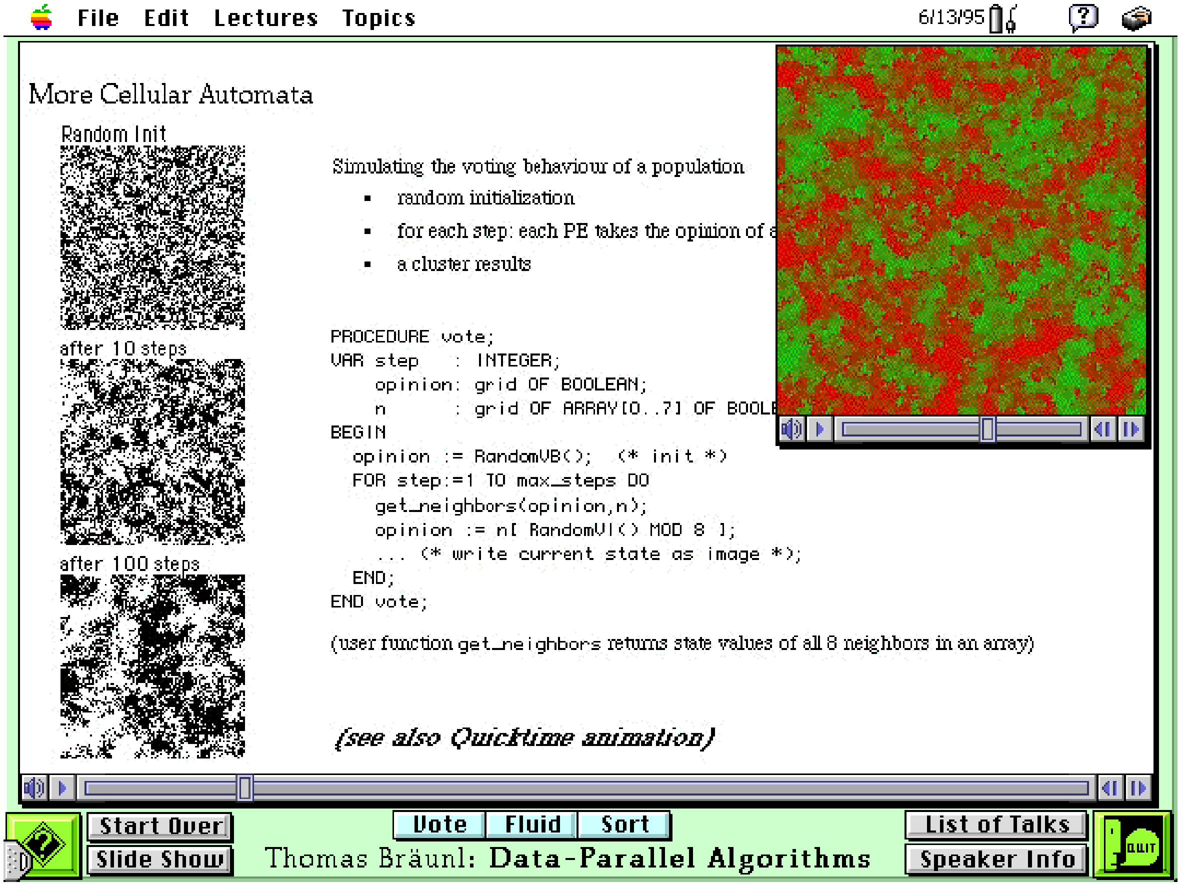
In other words, *simplicity of use* is what one should first care about, not power at all costs. Keeping it simple pays off not only in terms of production time, production cost and robustness, but it also — and that was the surprising result — increases usability.

These observations concerned mainly the hypertext user interface. The talk screen, on the other hand, was used significantly more and the only thing that readers were complaining about it was that the transparencies were not large enough to see details in some cases. Indeed, only 36% of the screen area contains useful information, even less than the hypertext screen which is 60%. Again, the problem is administrative debris getting in the way of useful information. When we designed the next CD-ROM [5] we took into account these observations.

4. Case Study 2: Multimedia Lectures

This multimedia interface is designed as an example of how to use interactive video in education and training, an area in which multimedia applications will have a great impact. Because of the size and composition of the production team (five untrained students working part time) , we were also interested in effective ways of production.

The CD-ROM [5] includes a number of introductory lectures for data-parallel computing. Its goal is to help newcomers in the area get a quick understanding of the issues related to data-parallel computing. It provides eight digitized talks on several introductory themes. The reader can follow dynamically a talk given by experts in the area, and see several animations associated with it. In this sense it gives more to the attendee than the actual attendance of the talk.



More Cellular Automata

Simulating the voting behaviour of a population

- random initialization
- for each step: each PE takes the opinion of a
- a cluster results

```
PROCEDURE vote;
VAR step    : INTEGER;
    opinion: grid OF BOOLEAN;
    n       : grid OF ARRAY[0..7] OF BOOLE
BEGIN
  opinion := RandomVB();  (* init *)
  FOR step:=1 TO max_steps DO
    get_neighbors(opinion,n);
    opinion := n[ RandomVI() MOD 8 ];
    ... (* write current state as image *);
  END;
END vote;
```

(user function get_neighbors returns state values of all 8 neighbors in an array)

(see also Quicktime animation)

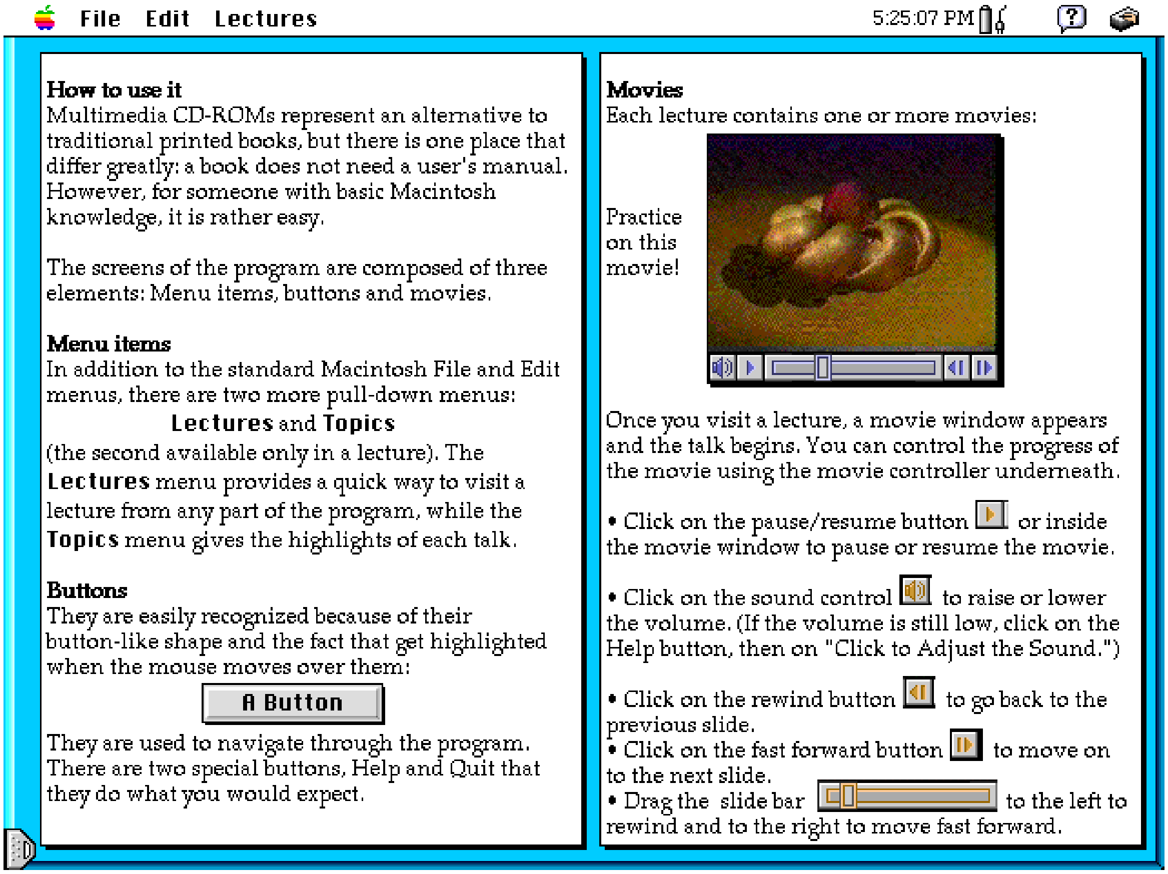
Thomas Bräunl: **Data-Parallel Algorithms**

Fig. 3: A typical talk screen of [5] Most of the screen space is devoted to presenting information.

The functionality we provided included only the basic operations that we found readers appreciate the most. As one can see in Figure 3, we use 80% of the screen for the interactive movie, while we provide ways to start and stop the talk, see a slide-show of the most important transparencies (controlled by the reader), get a list of the available talks and get information about the speakers. Moreover, we provided access to relevant animations that the can help clarify the presentation and specialized help, tuned to the particular screen that the reader sees.

We thought very carefully and decided to strip down the previous system's unnecessary complications. *Narrowing* the system was a rather brave decision: reusing the old system would have zero design (but not production) cost; moreover, one tends to only *add* things every time a new version of a product is released. However, we believe that this was a correct decision, since it allowed us to come up with the final product in a much shorter amount of time and put the effort saved in the quality of the contents. After all, any (multimedia or not) product is only as good as its contents!

Particular attention was paid for on-line help. We provide two kinds of help: A one-screen tutorial which explains how each of the objects of the interface works (Figure 4); and a specialized help screen to be consulted by the viewer when he/she has questions *during* the viewing of a talk.



How to use it
Multimedia CD-ROMs represent an alternative to traditional printed books, but there is one place that differ greatly: a book does not need a user's manual. However, for someone with basic Macintosh knowledge, it is rather easy.

The screens of the program are composed of three elements: Menu items, buttons and movies.

Menu items
In addition to the standard Macintosh File and Edit menus, there are two more pull-down menus:
Lectures and **Topics**
(the second available only in a lecture). The **Lectures** menu provides a quick way to visit a lecture from any part of the program, while the **Topics** menu gives the highlights of each talk.

Buttons
They are easily recognized because of their button-like shape and the fact that get highlighted when the mouse moves over them:

A Button

They are used to navigate through the program. There are two special buttons, Help and Quit that they do what you would expect.

Movies
Each lecture contains one or more movies:

Practice on this movie!

Once you visit a lecture, a movie window appears and the talk begins. You can control the progress of the movie using the movie controller underneath.

• Click on the pause/resume button ▶ or inside the movie window to pause or resume the movie.

• Click on the sound control 🔊 to raise or lower the volume. (If the volume is still low, click on the Help button, then on "Click to Adjust the Sound.")

• Click on the rewind button ◀ to go back to the previous slide.
• Click on the fast forward button ▶ to move on to the next slide.
• Drag the slide bar to the left to rewind and to the right to move fast forward.

Fig. 4: The help tutorial of [5].

A big difference between the printed and the electronic book is that the latter can be a *living* document, in the sense that it can be updated in short, regular time periods. Doing the same for printed books is much more expensive for both the producer and the "consumer". We wanted our product to be such a document, but burning a CD-ROM does not allow too much space for "life". During the last three years, the development of the World-Wide Web has changed the rules of the game. We have taken advantage of this technology by providing continuous support via a home page. In this site, readers can find updates of the software included in the CD-ROM, related publications, direct contact links to the contributors, pointers to related resources on the internet, on-line help, etc.

This web page can be accessed at URL: http://www.wellesley.edu/CS/lidpc.html

5. Guidelines

Among the lessons learnt, not only from our own experience but also from studying other multimedia user interfaces, are the ones we mention below. Because of lack of space, we list them without much elaboration:

- Investigate how users use the system and make it more efficient. This involves releasing a beta version to a number of representative users, and including their input, probably redesigning the interface. Provide shortcuts for all the common queries.
- Scrolling is evil – avoid it at (almost) all costs. After all, scrolling is inherently sequential and introduces from the window what hypertext kicked out of the door: instant access to relevant information.
- Screen space is valuable and very expensive – use it wisely. Do not force the material presented to fit around the administrative debris; it is the other way around!
- Fancy slide transitions (zooming in/out, page turning, fading in/out, etc.) are visually nice but expensive in terms of resources (both production and minimal system replay requirements). Use them only if you can afford them.
- Smooth visual transition from node to node and view of the history of the actions taken preserves reader's sanity.
- Content editing is *very* important – a book is only as good as its contents. Expect to spend most of the production time on it. (And even if you did not expect it, do not complain.)
- Content editing is *very* difficult – be prepare to put a lot of effort into it. You can only trust it to experts of the material.
- The previous two guidelines *do not suggest* that the design of the user interface should be ignored. Intuitive interface is essential to usability — people ignore (if not hate) manuals. (Figure 5 shows an example of the main screen ("The Browser") of a multimedia CD-ROM accompanying an excellent book, with a confusing user interface.)
- People like scanning visually the transparencies/contents, it gives them context and increases their interest in the material.
- Video/audio synchronization is important — and this is rather obvious — but you do not need experts for this.
- Running two video movies simultaneously from a CD-ROM is not here yet. If you need to use it, make sure you specify carefully the minimal requirements of the system.
- Keep what works well, damp what does not, no matter how much effort you put into developing. And, if you are not sure what works well, ask about the experiences of others — "don't get it original, get it right!"
- Always offer specialized help – there is no substitute for it when users really need it.
- The decisions behind the design of a user interface should *not* be the result of compromising or of union of opinions. Democracy should be left to other domains.
- Let the graphic artists do the artistic part — focus on functionality and quality of the material.

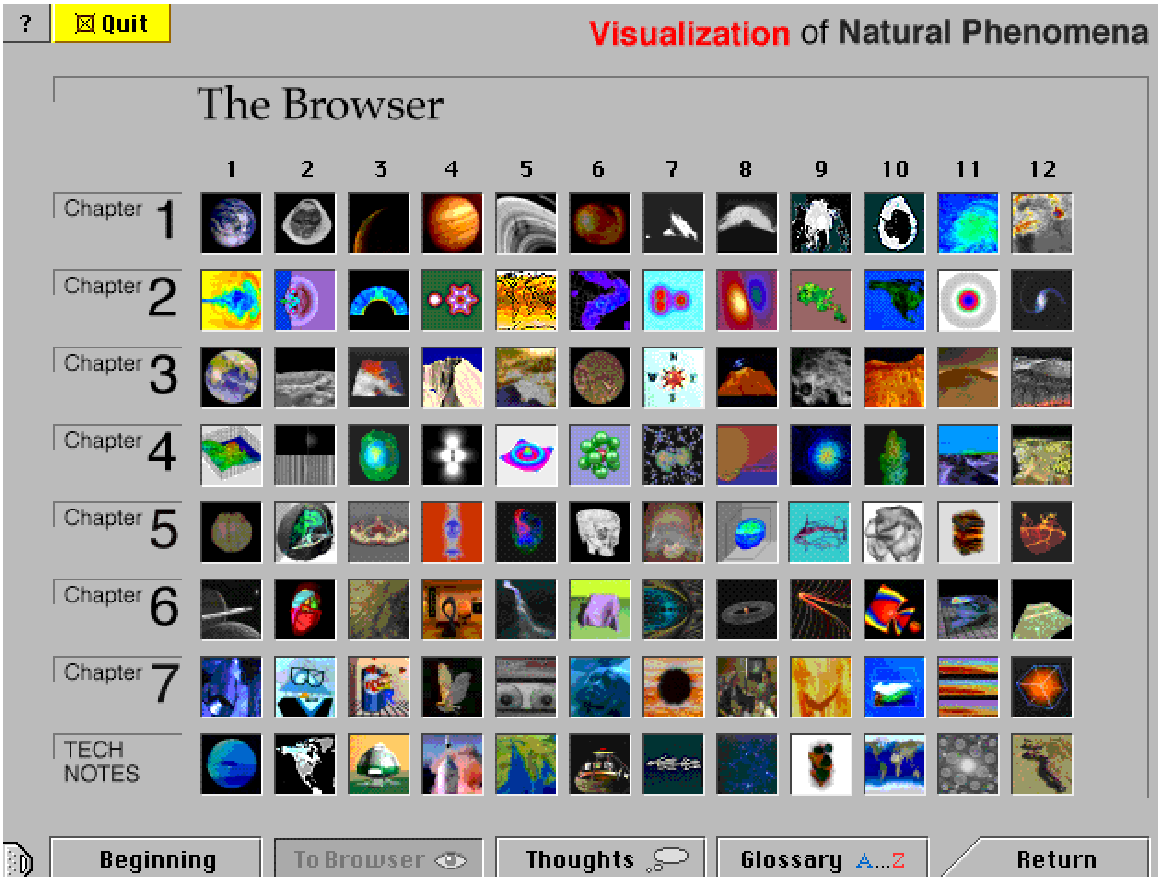


Fig. 5: The main menu of the multimedia application accompanying [6]
is not a match for the excellent contents of the book

6. The User Interfaces of the Future: Sound is Essential

One cannot deny that he GUIs of the '70s have been extremely successful. As we have mentioned, a measure of their success is the fact that have made computers accessible to virtually everyone — even to corporate managers. Another is, that they were able to create excitement many years after their introduction.

However, *this success makes surpassing them, much harder*. Indeed, this is the experience we have seen so far with 3-D GUIs, the obvious extension of 2-D GUIs: Visualizing a 3-D interface, organizing the information inside, navigating through it, programming it, supporting it — in short, everything — is much harder in 3-D. However, there have been a number of 3-D and pseudo-3-D interfaces have been developed so far. Below we mention some of the more promising of them:

- XEROX PARC's file system visualizer that allows the user to view the files in three different 3-D ways. It is arguably a useful way of visualizing the file system.

- MagicCap of MagicLink, Sony's palmtop computer/information organizer. It is a crude and not real 3D interface, but it demonstrates good functionality.

- QuickTime VR, Apple's approach to MT Architecture Machine Group's "Aspen Movie Map" project. The idea is simple: A movie is created from thousands of pictures taken from different angles. The viewer can turn and zoom the camera, but cannot look around objects. Unfortunately, creating VR movies is still much harder than regular QuickTime movies.

- VRML (Virtual Reality Modeling Language) that is designed to work on the Web, essentially a 3D scene description language plus hyperlinks.

But is this the direction we should be going to? 3-D and VR interfaces consume a large part of the processor's time, with diminishing returns. Adding texture color and movability in 3-D scenes is still a challenge to today's powerful uni-processor machines. It will eventually come, but it needs more powerful, parallel machines. Our main objection is, however, that even if they are eventually delivered in an economical way, they will just replace the messy desktop with a *messy office*, which may be much harder to manage. Shneiderman's pseudo-3D office [7] is an indication of how difficult would it be to manage a screen in which documents, applications ant tools are hiding in every corner of the 3-D space. We believe that, what we need instead is a *virtual administrative assistant*, where we can just ask for a particular file, using an approximate description, and have it available without having to look for it ourselves. (Microsoft's PERSONA research project, a parrot called a "conversational assistant" is a good step in this direction.)

Recall that the user interface is the "thing" between the human and the computer that helps them communicate. This is a two-way communication, from human to computer and from computer to human, and they *do not* have to be symmetrical. We will, thus, examine them separately.

From human to computer: Sound and touch-screens. Humans are efficient when communicating among themselves. They primarily use their voice, but also crude drawings on a piece of paper, and gestures. Sound is, therefore, essential for their communication. We believe that the next generation UIs should give the users the ability to find their way around the information using simple phrases like "Find the file that I used this morning," "Open the emailer and prepare a letter for John," and "Remind me to call home at 5 PM." This is not very difficult computationally: Given any restricted domain, we can quickly come up with a short dictionary for it. The sound interface does not have to understand every word in the language, it only has to recognize roughly the same number of words that today's menus contain, plus a few more (articles, numbers, etc.). Indeed, it has been observed [8] that the current state of the art in speech recognition and speech synthesis is quite adequate for this application. As an example of a successful product, we mention AsTeR, the impressive mathematics (TeX) reader created by T.V. Raman[9].

Sound is not enough, of course. A touch screen and a stylus is helpful for drawings and text. Lots of progress has been made in this direction lately, and new cost-effective products have reached the market of PDAs.

From computer to human: Sound and 2-D images. Sound can also be used in the other communicating direction. It is very often the case that we prefer to hear a message read to us in order to understand it quickly. This message could have been recorded by the originator, or the system software could read it to us. But listening without seeing has limited impression. What the screen should provide is 2-D images, carefully designed and laid-out. We have a five-hundred years experience in designing and typesetting book pages, and this experience will help a lot in designing the screen shots of such an interface. We should built on this strong experience before going to more exotic 3-D designs.

Bibliography

[1] Edward Tufte. Visual Design of the User Interface. Graphics Press, 1989.

[2] J. Matthews, P. Gloor, F. Makedon, D.B. Johnson, P. T.Metaxas (Contr. (Eds.) Parallel Computing: Issues and obstacles in the implementation of parallel machines, TELOS/Springer-Verlag, 1994,

[3] D.B. Johnson, F. Makedon, P.T. Metaxas. Proceedings of the 1992 DAGS Symposium, Hanover, NH, 1992.

[4] M. Cheyney, P. Gloor, D.B. Johnson, F. Makedon, J. Matthews, P.T. Metaxas. Towards Multimedia Conference Proceedings, Communications of the ACM, January 1996.

[5] P.T. Metaxas (Ed.) Introductory Lectures in Data-Parallel Computing, AKPeters, Inc., 1996.

[6] R.S. Wolff and L. Yaeger. Visualization of Natural Phenomena. TELOS/Springer-Verlag, 1993.

[7] B. Shneiderman. Designing the User Interface. Strategies for Effective Human-Computer Interaction. Second edition. Addison-Wesley, 1992.

[8] D. Weimer and S.K. Ganapathy. Interaction Techniques using hand tracking and speech recognition. In Multimedia Interface Design, ACM Press, 1992.

[9] T.V. Raman. ASTER — Towards modality-independent electronic documents. In Proceedings of DAGS'95, Boston, MA.