

HYPERTEXT II



JAKOB NIELSEN

Hypertext II was the major conference in Europe this year in the hypertext field and attracted 200 participants. Actually it had "attracted" many more who were just not there but had been turned away because the organizers had committed the same mistake as the planners of the Hypertext'87 conference in North Carolina and placed the conference in a location which would only hold 200 people. Their original assumption was that a hypertext conference in the UK might attract 100 participants, but that was the underestimate of the year. Hypertext is *hot* these days.

Most of the participants came from the UK, but there was also a fair number of participants from many other European countries as well as from USA/Canada and Asia (Japan and Singapore, of course).

As can be seen from the name, Hypertext II was the second UK hypertext conference. The first was (surprise) Hypertext I which was held last year for an audience of only 35 people. So since the proceedings of Hypertext I had seen only limited distribution, the organizers had taken the somewhat unusual step of including them in the material for Hypertext II (The Hypertext I proceedings are also published as *Hypertext Theory Into Practice* by Ablex).

Furthermore, the registration packet included a hypertext version of the Hypertext I papers in Guide form. Unfortunately this hypertext version was quite poorly done, as can be seen from a few examples: The very first chapter contains the following words: "Figure 1, on the next page ..." - a piece of text which was obviously taken unchanged from the printed version of the chapter. Since Guide uses scrolling text fields, there is no such thing as a next page,

and furthermore any decent hypertext would provide a link for the user to click on here. Even worse is of course the fact that the figure has not been included in the hypertext version. In general, only very few of the figures from the book are also in the hypertext. The hypertext also uses an inconsistent notation for the anchors for Guide expansion and reference buttons.

Ten Apples a Day

It was amazing to see the explosion of activity generated by HyperCard at various universities and research centers. This conference did not have very much work in evidence done at actual *companies* using HyperCard, however, even though I personally promote it also for prototyping "real" user interfaces in a paper presented to a broad audience at the NordDATA'89 Joint Scandinavian Computer Conference. In any case, many of the uses of HyperCard demoed at this conference had nothing to do with hypertext as it is traditionally understood but were simply prototypes of general graphical user interfaces. In spite of the heavy presence of Apple systems at the conference there were no speakers or demoers from Apple who had preferred to concentrate on some commercial event taking place in Glasgow.

One of the more interesting examples of the use of HyperCard was shown by **Harry McMahon** and **Bill O'Neill** from the University of Ulster who had placed a few Macintoshes with sound and image digitizers in an elementary school to get the pupils to **create their own interactive fictions**. Of course, most of these stories were fairly simple, such as (created by a 7-years old) "the teddy bear went for

a walk in the forest and met another teddy bear" - shown over a sequential series of HyperCard cards like a cartoon strip. More advanced designs used a facility called bubbles where the children can first draw their cards and then choose from various shapes of comics-like speech and thought-balloons to add to the image. The interesting idea is that it is possible to add multiple bubbles to each card whereupon they will be displayed to the reader one at a time. In this way, it is possible for the child to generate a dialogue between the characters in the story. It is even possible to contrast what the characters *say* with what they *think*. For example, in a story about a mouse about to be killed, it asked for a last wish: To sing a song. This wish was spoken out loud (placed in a speech balloon) but the mouse's thought (placed in a thought balloon) was "I am not as stupid as I look". The next speech balloon revealed that the mouse had chosen to sing the well-known song about bottles on a wall (falling down one at a time), but starting with "A thousand million green bottles sitting on a wall..." So this smart mouse would survive for some time to come.

Most of these stories were basically linear in nature which is why I said that they did not really have all that much to do with the concept of hypertext. McMahon and O'Neill had on purpose avoided to introduce commercial hyperstories (such as e.g. *Inigo Gets Out* or *The Manhole*) to the children so that they could observe the natural evolution of their approach to the new medium of interactive fiction. It actually did happen that a few 10 years olds did discover the hypertext principle on their own. They were creating a story about a person who was visiting an alien world and was captured by the aliens. He was offered a job by the alien boss and now thought to himself: Should I try to escape or should I take a job? The reader could click on either of these two thought balloons to proceed with the story. McMahon remarked that an interesting aspect of this story design was that the pupils had had to change their perspective on writing. Originally they thought of creating a story as they went through it (writing for the writer, as it were), but in this new situation, they had to consider what the reader could do and would want to do, so they had to change their perspective to writing for readers.

Hypertext in the Real World (Peter Brown)

The best presentation at this conference was probably the invited speech by Peter Brown from the University of Kent. Brown was the inventor of Guide which is now being sold for the Macintosh and IBM PC by OWL while Brown continues to develop the Unix version.

After his presentation, Brown was asked what had prompted him to develop Guide in the first place. His answer was that Guide had not originally been developed specifically as a hypertext system. It was developed as a solution to a problem: That of viewing online documentation which was traditionally displayed in a horribly perverted form of the paper versions.

Brown's talk was also concerned with the issue of providing solutions. He said that it is easy for researchers to move

in a dream world divorced from real user needs and that this is often encouraged by funding agencies who often do not want to admit failure by stopping grandiose but useless projects. Therefore he wanted to relate an example from the real world to show us how hypertext was really being used. OWL had developed the world's first personal computer hypertext system on the basis of Unix Guide and now has a large amount of world-wide corporate business in e.g. the automobile industry. But the Unix version of Guide has also continued in development at the University of Kent. So Brown felt that Guide could now be seen as a mature product in terms of usage experience.

Brown said that he is often asked whether Guide is better or worse than HyperCard: His answer is "no", since it is neither better nor worse, but different. One major difference is that Guide is based on scrolling while HyperCard is frame-based. Some information lends itself naturally to being divided up into several cards, but some doesn't, and in those cases the scroll model in Guide helps by removing the complexity of having information split over several cards.

In choosing a problem to present in his talk, Brown faced several difficulties: A toy project of a few thousand lines of text would prove nothing. There is also a danger of choosing a project which is the first in some field since the "first" of anything always generates extra excitement, attracts the best people, and gets extra funding, and therefore again would not be realistic. Even though he realized this, he had chosen an application which was the "first" since it at least was large-scale and had strict funding criteria where the people on the project had to justify the commercial viability of the project. Brown presented a project from the computer company ICL called LOCATOR done in collaboration with the University of Kent. People at ICL are actually going to sit in front of the Guide screen 8 hours a day to use LOCATOR.

The project is concerned with answering hardware fault calls (so-called "laundering"): Customers call for service over the telephone and the goal is to diagnose the fault so that an engineer can be sent *if needed* and will bring the necessary spare parts to the customer site on the first visit.

The system runs on Sun workstations, and the launderer sits at the workstation with a document displayed in Unix Guide. In such a real world project, things are never as smooth as one would want. For example, users change their mind half way through the conversation. The caller may not know the answer to technical questions and may have to suspend the conversation and call back later. Because of the importance of the laundering service, it has to be up all the time even when the permanent staff is away, so it may have to be manned in certain circumstances by temporary staff with only half an hour's training, leading to high requirements for usability. The design was fairly simple and consisted of hierarchical clicking on inquiry buttons to answer various questions. There were also hypertext links to various help screens, e.g. showing how a laser printer control panel looks.

A question from the audience was how that solution to the diagnosis problem compared with using an expert system. Brown's answer was that there had been some inter-company fights within ICL between expert systems and hypertext, but that in this case, hypertext won. Since they had not implemented the expert system also, we could not know for sure which option would really have been best.

After having presented this case study, Brown turned to discussing **eight issues in practical hypertext use**:

The first issue was that people do not want hypertext systems, they want solutions. Hypertext of course may be part of that solution. We need to use the customer's existing tools and often need to **integrate the hypertext tool with other tools** in a single seamless package to be part of the company information system

The second issue was **authorship**. The ideal would be to use completely differently techniques from paper and e.g. Glasgow Online was lucky to get funding to do this by starting from scratch. But in most cases one has to do conversion from existing materials in a two step procedure. The first step was the use of automatic tool to convert from plain text to hypertext (some people like Tim Oren from Apple think that this is a waste of time since you don't add anything to the info by automatic means). The second step was tailoring the hypertext by hand: This is expensive but necessary.

The third issue was **testing** which needed to be done in four different ways:

- a) Debugging linkage errors: It was fairly easy to have automatic check for dangling links and slightly harder to check that each link pointed to the node where it was supposed to go.
- b) Spelling checks and check for house style of hypertext. In their case, these checks could be done with automatic tools since Unix Guide uses `troff` tags for markup, allowing existing tools to be used.
- c) Testing the quality of the material itself.
- d) Testing ease of use using human factors methods. For example, they frequently found that buttons were too small. Brown said that there were no easy answers to this testing, but that one vital thing was to log each user session for future analysis.

Issue four was **handling large documents**. If one leaves out the images, the example ICL system is just a few megabytes, so it is not massive, but it is still large. The worst problem is taming complexity, and discipline is vital to doing this. To achieving such discipline, Brown advocated having a strong hierarchical backbone to the hypertext. One should also use common formats for the presentation of the information and develop a "**hypertext house style**" for how to use buttons, how to present them, etc. This discipline not only helps the author and the people maintaining the document, but also the readers.

Issue five was the **getting lost** problem and Brown noted that most of the papers at the conference were on this prob-

lem and on navigation. In their system, the getting lost problem is not so bad for the launderers but it is a moderate problem for the authors. Unix Guide's lack of `gotos` may have helped on this problem. In many cases they have paths which join: leading to the same conclusion from various starting points (e.g. the fault "no power" can be diagnosed from the symptoms "cannot boot" and "screen blank"). From the user's perspective, everything is represented as a natural hierarchy where the "no power" node is copied to every location where it is needed instead of having the user jump to a single "no power" node.

Issue six was the **cost of projects**. The LOCATOR experience was that diagnosis was 20% better than when done from a paper form of the information. The proportion of customer calls handled correctly had changed from 68% correct when using paper to 88% when using the hypertext, even rising to 92% after some use of the system. This leads to big savings in money when you don't have to have engineers drive in vain or back to get extra spare parts. It also does seem to be worth the extra cost of getting the workstation for displaying the info. The primary advantages of hypertext are better information and quicker access to the information. A further advantage is that when hypertext comes into a company it provides an opportunity to standardize the information.

Issue seven was **abstraction** in the information and getting away from `gotos` (Brown's pet peeve). ICL had produced a special authorship tool for the macro level in the original Unix philosophy of having special modular tools.

Issue eight was **handling change**. It is easy in hypertext to change the individual nodes, but hard to change the structure of the information base. The author needs to record why he/she did things and capture the design rationale: When another person looks at the document 3 years later it is nice to know why the structure is there and how it is. There is a potential serious problem with long term maintenance. We currently have no real experience with this aspect of hypertext but we can fear the worst because software maintenance typically accounts for at least 50% of the cost of a project and there is no reason to believe that it should not be just as bad for document maintenance.

We are roughly in the same stage with respect to hypertext development as programming was in the 1960s. In hindsight, the programming tools were not very good then, but people still did significant software projects, just as we can now do significant hypertext projects. We cannot assume that the problems will be completely solved (just as there is no "silver bullet" for software engineering), but we can aim for incremental improvements.

As a more general comment, Brown said that designers and computer scientists try to find a few general features which can cover all applications, but that the real world unfortunately is not like that. One needs special features for special applications which are so diverse, and sometimes, even different authors have different needs within a single application.

A Guru goes Commercial (Ted Nelson)

Ted Nelson was the second invited speaker and he gave his presentation actually dressed up in a nice suit. One of the main elements in Nelson's talk was his relief that he had finally been recognized by a major company in the form of Autodesk which had bought Xanadu and was going to release the Xanadu file system shortly. They are planning to release the product in the first quarter of 1990 in the form of software keeping track of addresses of information which is constantly moving around. You have virtual addresses which do not change and which can be kept in a document as pointers. The **Xanadu product in 1990** will be for single computers on a LAN connecting to a server. The bad news is that this is incompatible with all other current software. But we have to start over because of the current problem of people having so much data in different file formats which they cannot read anymore. Nelson called this a hideously tangled web. The Xanadu proposal is to have a stable repository of data which can be accessed by various other means. This is meant to be a win-win solution for everyone: A level playing field which can be used in many different ways. Project Xanadu will then release the full publishing system 2 years after the Xanadu server.

In true "hyper"-style, Nelson started his talk by quoting the conference chair, Ray McAleese's introduction to the proceedings. McAleese had written that "when Ted Nelson wrote that the structure of ideas is not sequential, he could not have envisaged how the idea of hypermedia would take off." *Oh, yes, he could*, said Nelson! It had been "obvious" to him that print would be replaced by online text by 1962.

Nelson complained that people always give him credit for having invented the word in 1965 but never ask him what he has done since then. One of the things he has done is to develop the Xanadu file system to serve as a backbone for large interconnected hypertexts. Nelson was very upset about the CD-ROM business, which is unaware of the general directions we must take. The good news is that you can have 500 Mb on a disk, but the bad news is that you can *only* have 500 Mb on the disk, because it isolates the knowledge on each CD-ROM: We are in a Balkanized information situation where higher and higher walls are being built around the borders of each document. It will not be true that Xanadu will only be useful when everything is in there. It will be useful in all situations where you want to include information from one document in another and keep track of versioning, such as e.g. a lawyer's office.

Literature is the interconnected set of documents in a specific field and it is the cross-connections which give the whole added value compared to the parts. Nelson admitted that free interconnections will result in a confusing tangle of stuff - but that is what we have already. When reading a specific document, you will see only the interconnections which a specific editor has decided are relevant to you, but you can push aside the editor's filter and see everything. Even though most of Nelson's talk was a repetition of material from his earlier books, this recognition of the need for editors seemed to be a new aspect of his thinking. We could actually develop the editing concept even further, since

there is of course the the democratic possibility of having several editors. But then we may start seeing the need for meta-editors in the form of people who recommend which editors you should pay attention to. This is somewhat like *The Whole Earth Catalog* which is a handbook of what handbooks are good to use.

Nelson was in favor of having hypertext through a concept he called **transclusions**. This involves not just having links but being able to take a part of another hypertext and include it in your own. Examples of this are collages and anthologies. Currently the copyright problem makes these kinds of publications difficult to produce in a kind of *n*-square problem where you have to write to everybody to get permission. For use in transclusions, the node model of hypertext is wrong. Nelson felt that we have to be able to make links to spans of bytes using a finer granularity than nodes. Furthermore, a span of characters may be discontinuous. Nodes are too primitive to support this, and in most hypertext systems, links usually point to either large chunks of information or to slits between chars (i.e. a single insertion point). Nelson said that he would be describing a lot of these ideas in an anthology to be released later in 1989 called *Replacing the Printed Word*.

At the end of this keynote talk, Nelson wanted to show some slides. Unfortunately some confusion had resulted from the transfer of the slides from his American carrousel to the British slide carrousel, resulting in the slides being completely out of order. And ironically, Nelson did not feel like giving a non-linear presentation of his slides (many people made fun of this). Instead, he got a chance to present his slides the next day when they had been put back in their intended linear order.

Nelson used plenty of one-liners in his talk, such as: "You have to dig deep to do anything serious with HyperCard and pretty soon you hit the cement at the bottom."

The Elastic Charles

The most technologically fancy system presented at the conference was the *Elastic Charles* by Hans Peter Brøndmo and Glorianna Davenport from the MIT Media Lab. This is an interactive film about the Charles River in Boston implemented on a Macintosh II with HyperCard, a ColorSpace II video overlay (genlock) card, and a video disk player. 15 students shot film about what they thought the Charles River was about (e.g. history, rowing regattas, bridge reconstruction, dams, pollution) and the *Elastic Charles* integrates all this film to a single hypertext magazine format.

Brøndmo said that they are not so much trying to develop sexy technology as to push the form of hypertext, especially with regard to extending the hypertext concept of linkage to temporal media such as video. They used icons to have a less imposing way to let users move along a story path. To generate good icons to represent video clips as hypertext destinations, they use miniaturized clips of video as icons which are shown on top of the main film image. They

had coined the term **micons** for these live video miniature icons.

When working with video, the linking paradigm gets a temporal component: The author has to define not just *where* on the screen an anchor micon should be placed (as in standard HyperCard), but also *when* it should appear and disappear while the user is viewing a piece of video. A further issues in jumping between film clips is what should happen when you return to the departure point: Should you continue to watch the film clip from where you interrupted it by jumping somewhere else, or should you start watching it over again (or just rewatch the last few seconds from before the jump)? It seems to me that there are major possibilities for disorientation buried here but unfortunately Brøndmo and Davenport did not present results of user testing of their interface.

They do have a filtering mechanism to allow users to see some links/micons and not others. They are also working on a voice interface to the filter so that the user can say e.g. "history" to see the history list icons.

In her presentation, Davenport mentioned that she considered *Aspen* as one of the most interesting art works of the early 80s. But unfortunately it never left the lab. Actually, it seems to me that MIT should donate the *Aspen* system to the Computer Museum if they have not done so yet.

Navigation and Browsing

About half of the sessions at the conference had navigation and browsing as their theme. Pat Wright from the Applied Psychology Unit in Cambridge listed five navigation tasks:

- 1) Going to a known place.
- 2) Going to a ill-defined place.
- 3) Going back to where you were earlier.
- 4) Going somewhere new.
- 5) Knowing how much reading would be involved to read a given section *before* you start reading it.

Wright also presented the results of studies looking at the first two of these issues. She had investigated the two main approaches to help readers goto a designated place in a text:

a) To have a separate navigation display. This leaves more room for text on main display and also gives more space for navigation support when the navigation display is actually shown because it can potentially take over the entire screen. On the other hand, having to shift back and forth between two displays may be cumbersome for readers. Wright tested this principle in an index design where users could go to a separate HyperCard card listing all the other cards in the stack.

b) Navigating from the text page itself. This saves an intervening step but since there may not be much screen space left over, users need to do more navigation. Wright tested this principle in a page design having some hypertext buttons directly on each HyperCard card.

She had tested the two different navigation designs with two small hypertexts and used subjects who were members

of general public with a mean age of 40 years. The results showed that subjects performed best with the index design for one of the hypertexts but better for the page displays for the other hypertext, so the conclusion had to be that there is no universally best navigation style. In some cases it may be reasonable to separate the hypertext navigation from the text display to provide users with a better overview but in other cases it is most efficient to allow users direct links.

In a questionnaire they asked their subjects to comment on how good they would think online information would be for various kinds of information to be read by various kinds of users. The subjects having used the index design rated 48% of the items as suitable while the users having used the page design only rated 20% as suitable for electronic presentation. This is an indication that the users of the system with a separate navigation display had had the most positive experience with hypertext.

In another study, **Cliff McKnight** from the HUSAT research center at Loughborough had tested various different ways to provide overview diagrams. They had used the text on houseplants that had also been used in Pat Wright's experiments so there might over time be a chance to collect several different research results based on the same text.

They had tested three different factors in the design of overview diagrams: Listing the elements either hierarchically or alphabetically, the presence or absence of a current position indicator, and finally the use of typographical cues to signal the hierarchical structure of the text (by the use of CAPS for major headings, etc.). Unfortunately they had not tested graphical layouts versus purely textual layouts, but of course their experiment was complex enough already.

In describing the design of their experiment, McKnight stated that "obviously" the subjects had been prevented from using simple string search in answering the questions. Actually this indicated to me that they may be testing the wrong kind of tasks. Personally I find it more interesting how overview diagrams perform when they are used in combination with other navigation methods since it is exactly when users jump around in an uncontrolled manner that the overview diagrams may be the most useful.

The results showed that subjects using the hierarchical contents list navigated more efficiently than subjects using the alphabetic contents list and that having a current position indicator also improved efficiency. There was no significant effect of typographical cues.

The overview diagram was accessed more times by subjects who had the hierarchical contents list than subjects using the alphabetical contents list which might indicate that the hierarchical list was more useful. This study used a very small hypertext of 24 HyperCard cards and it is likely that the hierarchical contents would do even better in a larger hypertext where it would not be possible to show a complete alphabetical listing on a single card.

Finally, McKnight mentioned that they had used an objective measurements of task performance to assess whether

users were lost in the hypertext and not the users' own subjective feelings of being lost. In real life, it may be that the users' feelings are the most important since they will determine whether this kind of system will be used.

Hypertext meets Interactive Fiction

Gordon Howell from the Scottish HCI Centre in Edinburgh gave an introduction to the field of interactive fiction. There is a fundamental difference between interactive fiction and information systems because the goal is not to find the answer as soon as possible but rather the experience itself, so you may want to draw out the resolution of questions for quite long. Howell started by giving some real life examples of creating your own stories as you go along:

Conversations (you are not sure what you are going to say before you say it), **panel sessions** at conferences, **games** such as adventure games, and several books, including both encyclopedias, Borges' *Ficciones*, and Pavic's *Dictionary of the Khazars* which was totally sold out in the Edinburgh bookshops after a favorable review in the Scotsman.

There are difficulties in interactive fiction in practice, however. Current technology limits the size of an interactive fiction so that after a short time you will have read it all or at least have an indication of what the author has to offer. Current interactive fictions are also too predictable so that the reader does not really feel in control: The illusion of free will is lost and it becomes a game instead. Finally, a lack of serious consideration in the literary world leads to interactive fiction being on the fringe, leading to mostly trivial works being produced.

For the future of interactive fiction, Howell assumed that we would get more interactive fiction shells like *Storyspace* and more sophisticated models and algorithms for interactive fiction. This would not be enough, however, because we would also need a serious treatment of interactive fiction in education where the students try out experiential reading and a literary consideration of interactive fiction by real writers. Commercial interactive fiction would lead to the development of better hardware and user interfaces as well as to marketing concerns. All of this should be a natural for the entertainment industry. Howell felt that we have a responsibility to open up new forms of self-expression for people and therefore called for more research and practice in the field of interactive fiction.

System-Assisted Customized Browsing

It is fashionable in the user interface field to talk about allowing users to customize their interfaces. In real life, however, most systems either don't allow any customization or are limited to surface changes such as color assignments. Andrew Monk from the University of York had designed a facility for customization in hypertext by the construction of a **personalized browser** which would hold hypertext links to those places in the hypertext network to which the user would want to have easy direct access.

Of course, just providing a facility for a personalized browser is not enough. It must also be easy for users to construct such a browser, since they will otherwise not use the facility. In Monk's design, ease of construction was achieved by having the system unobtrusively monitor the user's movements through the hypertext. If the user returned to a specific node frequently enough, the system would assume that that node would be a candidate for inclusion in the personalized browser and would ask the user if the node should be added to the browser. If the user answered yes (a single click), a link to the node would be added to the personalized browser.

One problem with this approach which I raised during the discussion following Monk's presentation was that it is somewhat similar to the principle of "activist help" where the computer interrupts the user's work. An activist interruption may be helpful in situations where the user is having trouble, but it may be disruptive to users in the navigation situation. Unfortunately, Monk did not yet have enough practical experience with his design to be able to tell whether this was indeed a problem or not.

Monk had implemented his design in HyperCard as had so many other speakers at this conference, but Dan Russell asked what would happen in hypertext systems with multiple windows rather than a single frame. In e.g. NoteCards, the user's state could be viewed as consisting of the complete set of currently open windows, so one would want to have a reference to such a "tabletop" from the personal browser. The reference itself would be no problem since they have already implemented a tabletop facility at Xerox PARC, but the monitoring of the user's navigation behavior would be harder since users rarely return to exactly the same configuration of windows. Therefore the simple-minded solution of just counting how many times a given state occurred could not be used to activate the prompting mechanism for adding the state to the browser. After the session I came up with the possible solution of using a clustering algorithm to construct sets of related windows based on their similarities as measured by how frequently they were opened at the same time. The system could then count how many times each cluster was displayed, and the customization could proceed as with Monk's current design.

It is interesting how each new idea in hypertext gives rise to many additional user interface issues that we don't really know how to answer. One more such issue that came up in the discussion of Monk's personalized browser was what would happen in large hypertexts where the browser would tend to accumulate many references and therefore grow unwieldy. The answer seems to be that the user would probably have a current working set of hypertext nodes which were important at any given time, and that it would be those nodes which should be listed in the browser. One approach would be to use methods from virtual memory management and throw away the least recently referenced node if too many links are added to the browser. To stay with the paradigm of a customized interface we should probably require the user to confirm the deletion before it takes place, but it would still be nice to have the system come up with a suggestion for what link to remove.

Travel Difficulties

The travel metaphor is popular among hypertext enthusiasts so it gave rise to many jokes that workers at British Rail chose the day before the conference to strike and close down all train service in the UK. The conference participants from the UK knew about the strike in advance and so had planned to drive to York by car. I however, arrived in Manchester Airport from Denmark without expecting any trouble. When I asked the taxi driver to take me to the rail-

way station, he replied "why do you want to go there, there are no trains running today." Of course by then it was too late to get a rental car (completely sold out), so I ended up having to take a taxi all the way from Manchester to York. Luckily taxis in Manchester are of the London model with large, comfortable interiors, so I was able to spend several hours in the taxi reading submitted papers for the Hypertext'89 program committee and typing reviews on my trusty Z88 laptop computer.
