Journal of Universal Computer Science, vol. 4, no. 11 (1998), 839-855
submitted: 1/9/98, accepted: 3/11/98, appeared: 28/11/98 © *Springer Pub. Co.*

Addressing Design and Usability Issues in Hypertext and on the World Wide Web by Re-Examining the "Lost in Hyperspace" Problem[1]

Yin Leng Theng
Middlesex University, London
y.theng@mdx.ac.uk

Harold Thimbleby
Middlesex University, London
h.thimbleby@mdx.ac.uk

Abstract: Users tend to lose their way in the maze of information within hypertext. Much work done to address the "lost in hyperspace" problem is *reactive*, that is, doing remedial work to correct the deficiencies within hypertexts because they are (or were) poorly designed and built. What if solutions are sought to *avoid* the problem? What if we do things well from the start?
This paper reviews the "lost in hyperspace" problem, and suggests a framework to understand the design and usability issues. The issues cannot be seen as purely psychological or purely computing, they are multi-disciplinary. Our *proactive, multi-disciplinary* approach is drawn from current technologies in sub-disciplines of hypertext, human-computer interaction, cognitive psychology and software engineering. To demonstrate these ideas, this paper presents HyperAT, a hypertext research authoring tool, developed to help designers build usable web documents on the World Wide Web without getting "lost."
Key Words: Lost in hyperspace, World Wide Web, hypertext, human-computer interaction, software engineering, cognitive psychology, authoring tools, design, usability.

1 Introduction

Hypertext technology still continues to excite many, ever since it became popular in the late 1980s. Many people then were sceptical that the technology might just be a passing fad. The promise behind hypertext is too fundamental to disappear quickly, and there are reasons to believe that hypertext technology promises something special: hypertext is everywhere, from mobile phones, television information systems (e.g., teletext), to the World Wide Web. By the end of 1998, the Web had an estimated 100 million users. Hypertext has affected us directly or indirectly in almost every facet of our lives, ranging from scientific work to business and education needs, to our general way of life.

Although hypertext users benefit from the information they read and from the richness of associations supported by the network of nodes and links, there are problems. The problems fall into two main classes [Conklin 1987]: problems with current implementations, which include delays in the display of referenced

[1] This is an extended version of a paper presented at the WebNet '98 conference in Orlando, Florida. The paper has received a "Top Full Paper Award."

materials and deficiencies in browsers; and secondly, problems that seem characteristic of hypertext, such as cognitive overload and disorientation.

Cognitive overload is the additional effort and concentration users find necessary to maintain several tasks or trails in a hypertext at one time. The resulting disorientation experienced by users has been called the "lost in hyperspace" (LIH) problem. LIH can refer to any of the following user problems [Conklin 1987, McKnight *et al.* 1991, Nielsen 1996]: the problem of not knowing where they are in the network; not knowing how to get to some other place they know (or think) exists in the network; not knowing how to return to a topic left previously; forgetting why they want to go to a certain topic in the first place; the problem of forgetting what topics they have browsed; and the problem of forgetting which key points are covered.

Ironically the LIH problem has given rise to much controversy itself. Some think that the LIH problem is one of the most difficult issues in hypertext research and that there is yet more to be done to ameliorate the problem. However, there are others who believe it is not a significant problem, and they feel that efforts should be channelled to address other more pressing issues (such as delays, or history mechanisms). And if you ask users what they think, they are likely to concentrate on 'obvious' problems — almost by definition a problem caused by cognitive overload leaves little cognitive resources to notice, let alone explain, the problem!

This paper examines the LIH problem and questions whether LIH is a significant problem that still warrants the attention of the research community.

Terminology

The abbreviation LIH refers to the "lost in hyperspace" problem throughout. For conciseness, we will call multimedia, hypermedia and the WWW "hypertext systems" since the issues surrounding the LIH problem with which this paper is concerned with apply to all of them.

The term "hypertext" is used to denote a hypertext document, made up of interlinked pages or nodes. The term "hypertext system" will refer to the software tools used to create a hypertext, and the term "browser" to the systems that users use to read the hypertext. Some hypertext systems are browsers.

2 Re-visiting the "lost in hyperspace" problem

In a sense LIH *is* the user's problem — it is they who are lost, after all. Therefore research has been directed at improvements in the presentation of hypertext information, the design of navigation, and in the design of 'spaces' that are more intuitive for users. Not surprisingly many research solutions involve the use of graphical browsers and novel query/search mechanisms. These approaches seem to make the following assumptions referenced to users' supposed failings [Theng *et al.* 1997]: users have a wrong or incomplete conceptual model; users lack experience in using hypertext for performing tasks such as browsing; users are distracted because of the "embedded digression problem"; and users don't understand the chosen display conventions.

Because both hypertexts and hypertext systems are difficult to build, many commentators are happy to seek solutions in better understanding users and helping them cope. There has, of course, been some success in this approach — maps, virtual reality visualisation. Whilst such improvements are welcome, even necessary, we believe they are not sufficient. Worse, it may be that by seeking 'pleasant' browsing experiences, more fundamental problems (of which LIH is one) may be being palliated or concealed.

So we argue that perhaps wrong or inappropriate solutions are being sought because incorrect or incomplete assumptions are made. Hypertext design is hard, and hypertexts are used less effectively than we might wish. The question we ask is: is LIH primarily a psychological or an engineering problem? In other words, is LIH a problem for users, or is it a symptom of poor design, which itself may be a psychological problem for hypertext authors.

The answer to this question will have serious implications on the sorts of solutions being sought to address the problem. If LIH is a psychological problem for users, then the problem may be entirely due to their inability to exploit computer screens, complex information structures, and that nothing in the design is going to fundamentally ameliorate this — though the experience might be made more pleasant.

Though disorientation arises within the user's mind (which most research findings support) we argue that research should not rest on users *alone*! The LIH is also an engineering problem. Our view implies that LIH is at least in part, and perhaps significantly, attributable to bad system design, and poor design causes psychological problems for users too. We believe the user's problems can be traced back to bad browsers, bad hypertexts and bad authoring systems — hypertext designers themselves get LIH, thus creating problems that are more obvious with end users. We agree with [Mayes *et al.* 1990] that addressing the LIH problem within hypertext goes beyond providing more and more navigational aids. It is a deeper problem: because hypertext authors themselves can get lost in the process of designing and authoring hypertexts, they inadvertently contribute to poorly designed hypertexts, which in turn leads users often being lost.

How can we find better organisational principles for hypertext? We argue for a move away from treatment to prevention, from treating the user's symptoms — themselves a reaction to bad design — to avoiding the bad design [Theng *et al.* 1997]. We need to re-examine the way hypertexts are designed and built. We need to examine fundamentally how information should be structured and displayed. We should not assume that if certain design features work well for some information contents and purposes, it will be appropriate for others.

3 A multi-disciplinary approach to address the "lost in hyperspace" problem

Building systems with effective design involves a variety of skills, drawing upon and effectively harnessing many backgrounds and disciplines [Baecker *et al.* 1995]. Hypertext is no exception. If previous attempts drawn from fields like human-computer interaction or hypertext research are inadequate to solve this problem, then perhaps there is a need to get help from other disciplines. Kim advocates that where one discipline is weak, we should look for another that is strong [Kim 1995]. We should also learn to build and use bridges between the many relevant disciplines and perspectives to create the opportunities for creative synergy.

Since LIH is a complex problem, a multi-disciplinary approach seems essential to draw upon knowledge and findings from diverse disciplines such as human-computer interaction, software engineering and cognitive psychology, and integrate this knowledge into hypertext. By "discipline," we mean the body of shared concepts and criteria collectively agreed upon by researchers in the same field. The values defined by a discipline are criteria for deciding what good and bad research is for that discipline. Immediately there are problems, because 'good' may be very different when seen from each discipline! Briefly, good in human computer interaction is typically grounded in empirical data from usability studies; good from software engineering is typically grounded in sound engineering practice, such as formal specification; and good in cognitive psychology is typically grounded in models of human behaviour and performance. Academic research may wish to promote a discipline-specific good, without seeing the wider picture — so, for example, few software engineers are seriously concerned with usability. This paper is concerned with ways of enabling how the concepts, values, methods and procedures from these disciplines can be integrated into hypertext to address the LIH problem (see Figure 1).

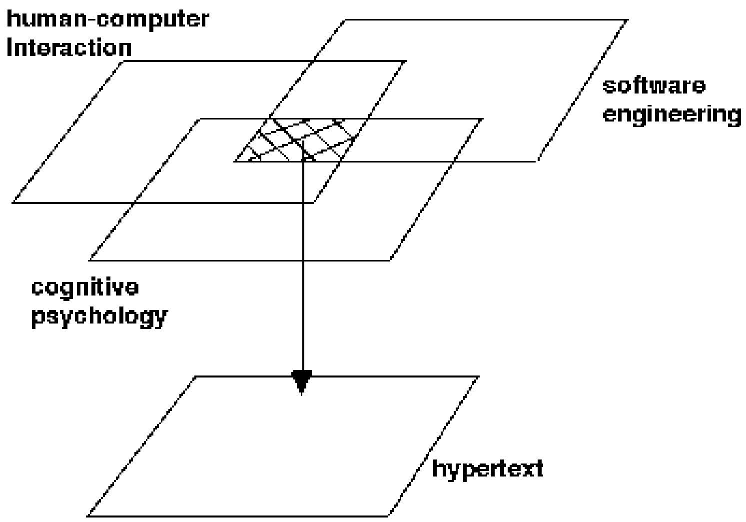

Figure 1: Integrating commonalities among the disciplines of human-computer interaction, software engineering and cognitive psychology into hypertext research.

We took a breadth-first investigation of the LIH problem: this is a productive problem-solving approach — as opposed to a reductive problem-solving approach. A reductive problem-solving approach would be a hindrance to finding a solution since it focuses on known aspects of the problem, and therefore may not be able to see new solutions.

Figure 2 outlines an overview of the methodology taken so that more usable hypertexts are produced. Since LIH is a complex problem involving both engineering as well as psychological issues, there is a need to take a step back and examine more fundamental issues. Four proactive, multi-disciplinary approaches to eliminate, or at least ameliorate the LIH problem are proposed, based on: design principles and guidelines; engineering, task-based approaches, to understand users' needs; hypertext structure; and support tools for authoring.

Each of these four approaches can contribute towards an integrated solution in the form of an authoring tool to address the LIH problem on the web. These user-centred methods used in the methodology focus on the activities in the "Design Stage" of a typical user-centred design cycle as shown in Figure 3.

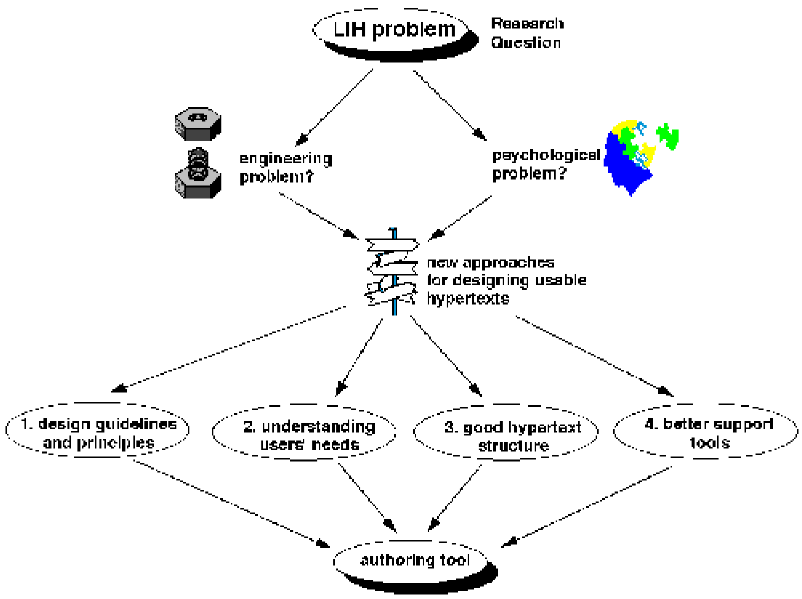

Figure 2: Overview of the methodology taken to address the LIH problem.

We integrated the following principles into our approach [Baecker *et al.* 1995], [Gould *et al.* 1991], [Landauer 1995], [Lewis and Rieman 1993], [Nielsen 1996], [Lindgaard 1994]:

Getting to know users and their needs It is well-known that designers often design for themselves unless they are trained to realise that people are diverse, and that users are unlikely to be like them. Solutions to the LIH problem should then address the issues of helping users navigate through conceptual space. It is important to have an accurate representation of users'

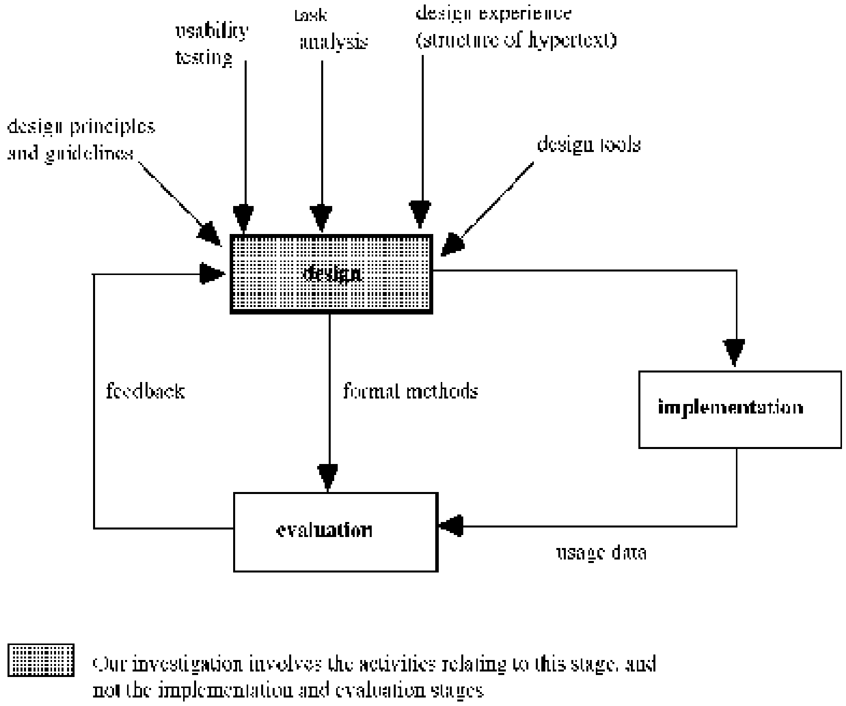

Figure 3: User-centred design cycle (adapted from [Perlman 1988]).

behaviour and actions when they perform or try to perform common tasks such as browsing, information search, seeking references and recall. By trying to make sense of what users should do or what they actually do, hypertext authors will stand a better chance of producing user-centred hypertexts that meet users' needs more effectively.

Iterative design Interactive systems require iterative design. The most promising approach is to iterate design and evaluation until a satisfactory result is achieved. To make iterative design possible when facing budget or schedule constraints, a rapid prototyping approach is used to support mocking up and trying out interfaces and dialogues. Since disorientation can occur in a spatial network of nodes and links, there is a need to re-look at design issues. It is important to ensure that good hypertext design principles and guidelines are incorporated into the building of hypertext in the first place. In considering design principles and guidelines, there is also a need to investigate the "best" structure for hypertext according to their functions.

Early and continual user testing Because it is also imperative that hypertext authors build into the hypertexts an accurate representation of user models when performing these tasks, early and continual user testing is essential. This will ensure that the system is designed to meet users' needs.

Integrated design All aspects of usability should evolve in parallel, rather than be defined sequentially, and should be under one management. We integrate these approaches onto a single hypertext system, the web, to help users navigate around it without getting lost.

Prototyping It is clear that *ad hoc* methods of designing, constructing and validating hypertexts are not enough. If users get lost in hypertext, designers must do too. This suggests that tools for designing hypertext should provide much improved computational support for designers. We will describe a prototype hypertext authoring tool built to help designers build well-structured hypertext prototype.

Usability testing To find out if the prototype hypertext authoring tool is "good enough" to help designers produce a "good enough" hypertext, there is a need to know how to express in quantifiable, measurable terms when a "good" system is "good enough." Failure to do so will render designers incapable of substantiating claims of its usability and finding improvements in a system.

Though these approaches should contribute to improve the quality of authoring and hence the hypertexts produced, in practice the pressures of time, money and resources often leave designers having to come to a compromise between implementing quick-fix solutions as opposed to searching for effective solutions. The methodology mapped out in Figure 2 should therefore realistically be modified to accommodate elements of realism!

Though we attempt to re-consider the LIH problem from different angles in order that solutions to address it are more complete, we do not claim that the approach taken is the *only* best way to design usable, well-structured hypertexts. One reason this might not be the case is that hypertexts differ in their information structure and in the user tasks they support. For example, a user checking a store's website to find out more about a certain products would employ a different style of interaction from another user looking for the email address of a member of a society.

4 A hypertext example: The World Wide Web

Ad hoc methods of designing, constructing and validating hypertexts are not enough. The World Wide Web is chosen as the example to demonstrate our ideas since it is the most widely used and largest hypertext system ever. (In the ACM Hypertext'97 conference, John Smith's keynote address challenged the hypertext community not to ignore developments on the web, but to work alongside the web community.)

Nielsen predicts that due to a change in the dominating styles for web sites over recent years, a real contribution essential for web design would consist of further research into different knowledge areas [Nielsen 1995]: knowledge of icon design; knowledge elicitation to discover appropriate information space structures; usability testing; and task analysis techniques.

But searching for solutions in isolated disciplines, and recommending them to designers in the hope that they would somehow remember to put them into practice may not be as simple as it sounds. Many factors could prevent well-intentioned designers putting these suggestions into practice. One would be that designers might be overwhelmed, or might not have the time and capacity to attend to all the authoring details.

In order for Nielsen's suggestions to be effective and implementable, it should go beyond just providing designers with a list of 'do's and 'don't's. Designers need help. If some of the ideas could be automated so that designers need not worry about their implementation, chances are that better websites could be produced since designers would be freed to concentrate on other critical issues that cannot be automated, but require human judgement and expertise.

5 HyperAT: Implementation And Initial Evaluation

HyperAT is a research tool to help designers manage the complexity of the design and validation processes without themselves getting "lost." Multi-disciplinary approaches are integrated and culminated into a practical authoring tool. HyperAT stands for "Hypertext Authoring Tool." HyperAT is a prototype designer tool for authoring hypertext and web documents (see Figure 4). The philosophy and the underlying concepts taken in the design of the authoring and usability components in HyperAT are described in greater detail in [Theng *et al.* 1997]. In the Figure, Inputs refers to the multi-disciplinary approaches that underlie the design of the authoring and usability components. Approach One examines design principles, and in HyperAT, it is examined in the form of good web style guides. Approach Two emphasises the importance of understanding users' needs and the tasks they perform. In HyperAT, we explored users' browsing needs on the web. Approach Three stresses good structuring, and therefore in HyperAT, good web page structure to help both designers and users is investigated.

Besides providing the basic authoring facilities to produce web pages, HyperAT also delivers usability evaluations to designers regarding problems that might be detected during its analysis.

HyperAT is implemented in Macintosh Common Lisp (version 3.9) for PowerPCs. It does not provide a full range of editing facilities with an especially attractive interface, and it does not compete with commercial tools in this respect. However we see HyperAT as an analytical research tool for contextualising and delivering the results of hypertext usability to hypertext designers.

5.1 Authoring components

The main objective of HyperAT is to help designers build usable, well-structured hypertext. In designing the authoring components, we incorporated two underlying design concepts, that is, the need to impose a structure, and the need to incorporate good web style principles and guidelines.

The authoring component in HyperAT provides the basic authoring environment for the creation and modification of hypertext. HyperAT's facilities are accessed using a graphical, user-based interface. Hypertexts are created via a form-like screen (see Figure 5) and converted into predetermined HTML format, which can be displayed on the web using a web browser. During the conversion of the hypertext into HTML, HyperAT generates a table of contents, a hierarchical representation of the structure of the hypertext, accessible from every page of the hypertext, using the "contents" button. A fisheye view of related pages with respect to users' current page is also provided to help users better understand

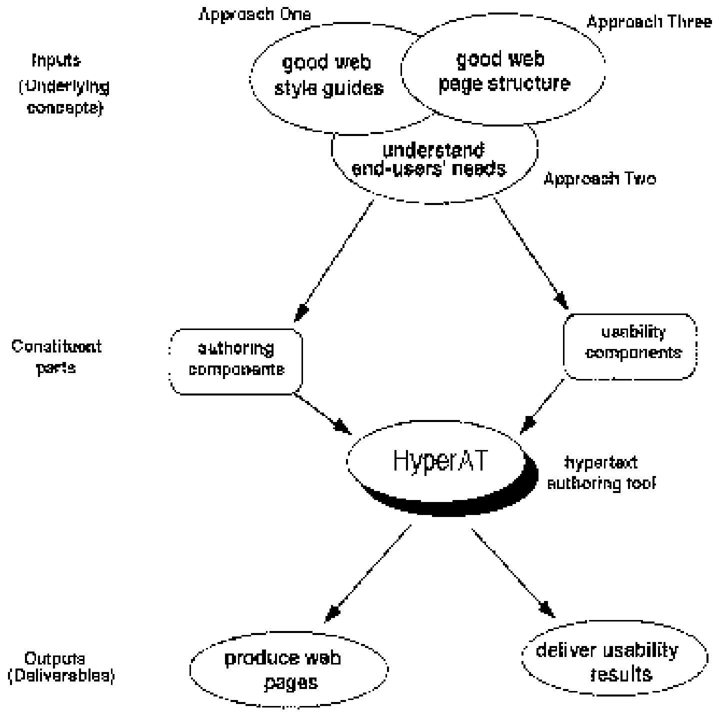

Figure 4: General overview of HyperAT, its inputs and outputs.

the structure of the hypertext in relation to where they are, thus ameliorating LIH [Theng 1997]. Designers can graphically display both global and local views of the structure of hypertext created. Hard copies of the hypertext structure and the associated HTML coding can be printed and kept for documentation as well as for maintenance purposes. An HTML-editor provides designers with a menu to write HTML, without designers having to memorise HTML codes.

Other authoring aids include a generated trace of created pages during a HyperAT session to provide useful memory jots for designers, who may be interrupted during the HyperAT session or are simply confused over the nodes created. Another is a generated global map showing the structure of the hypertext with its constituent nodes. Clicking onto a node will bring up another map, with that node as the root node, providing designers with a fisheye view cancelling off other details not related to it.

We built in some established website design principles and guidelines to illustrate that some can be automated in an authoring tool without designers having to worry about their implementation. Thus, every web page has a standard "look and feel" with navigational buttons at the top and bottom. Each web page contains essential elements like the title, author, date and time of creation, button bars which represent fixed links that allow users to move to content page,

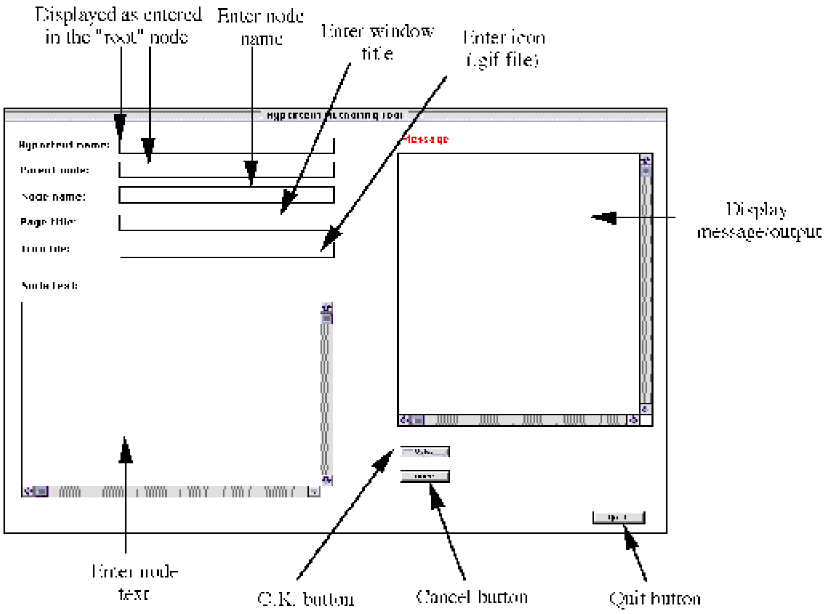

Figure 5: Entry screen for creating nodes in HyperAT.

previous page, or next page. Links to other related web pages are also generated for each web page so that users can have a better understanding of how they can obtain related materials.

HyperAT also provides users with prospective information by generating the title of the pages users would move to if they were to click onto the navigational buttons. Because these prospective views that accompany navigational buttons are not hard-coded but are automatically generated by HyperAT, no extra effort is required from designers to ensure the inclusion and maintenance of this feature.

5.2 Usability Components

For the usability components, HyperAT incorporates features that help designers to better understand users and their browsing behaviour. Hence, apart from the basic editing facilities of create, edit and save, an experimental, authoring testbed allows hypertext designers to carry out different modes of usability testing on the hypertext created by HyperAT: (see Section 5.2.1) structural analysis and (see Section 5.2.2) real user evaluation.

5.2.1 Structural Analysis

Not only do we want to help designers structure information, we also want them to avoid structural inconsistencies and mistakes. HyperAT allows designers to analyse structure by performing integrity checks on the nodes and links, and secondly, measuring the complexity of the structure of hypertext. The metrics implemented to measure the complexity of the structure of the hypertext are: number of nodes; number of links per node; all possible paths from a given node; depth of a structure; number of successors; and others can be added. In HyperAt, we have implemented some possibilities. Such structural analysis of the hypertext alerts designers to take corrective measures as early as possible in the design process before it is too late, and often before much effort has been spent on contributing context.

5.2.2 Real User Evaluation

Real user evaluation is important because hypertext are designed for users and not just what designers think or feel are important. Real users can be employed to evaluate hypertext on the web with their transactions logged by the server log files [Berners-Lee 1995]. However, Berners-Lee cautions that analysing the server log files takes time if designers have to do it manually. To help designers analyse log files, HyperAT has a facility that parses and analyses server log files and interprets them, providing designers with useful insights into understanding users' browsing pattern. This data can be useful design aids to pin-point usability problems, and to guide design decisions. HyperAT also measures: frequencies of visits; clients' browsing information; pages visited; and clients visited — and so on.

5.3 Initial Evaluation

We carried out initial evaluation of HyperAT to get qualitative results and impressions on its usefulness and usability as a web authoring tool by: (see Section 5.3.1) comparing web documents made with and without it; (see Section 5.3.2) getting feedback from hypertext designers; and (see Section 5.3.3) comparing it with other tools.

5.3.1 Comparing Web Documents

The main objective is to compare a website re-constructed by HyperAT with the original website produced by the original creator of the website using a standard HTML editor. Figure 6 shows a sample web page of the "Childnet International" website created by AtlasWWW, the original creator. Figure 7 shows the same web page created by HyperAT. It is to be noted that by using HyperAT, not only was the original "Childnet International" index page re-created but useful information for each web page to help user navigation were automatically generated. They include the automatic generation of text-labelled navigational buttons with prospective views, local views of related pages in relation to users' current page, table of contents providing a global view of the structure of the "Childnet International" website. For easier printing, a feature to print the website as a single document is also incorporated in this web page. Other automatically-generated information such as the date and time of creation, and creator of the web document, are also displayed in the "table of contents" web page (see Figure 8).

We have shown that HyperAT is capable of creating any web document produced by any standard HTML editor. In contrast with the original web page, the web page generated by HyperAT contains navigational aids that are useful for effective user navigation. Because these aids are automatically generated by HyperAT, incorporating them into the web page would require little or no effort from designers. Designers can then concentrate on the content of the web documents, thus increasing the chance of producing better, usable web documents.

Figure 6: A sample web page from the original Childnet International website.

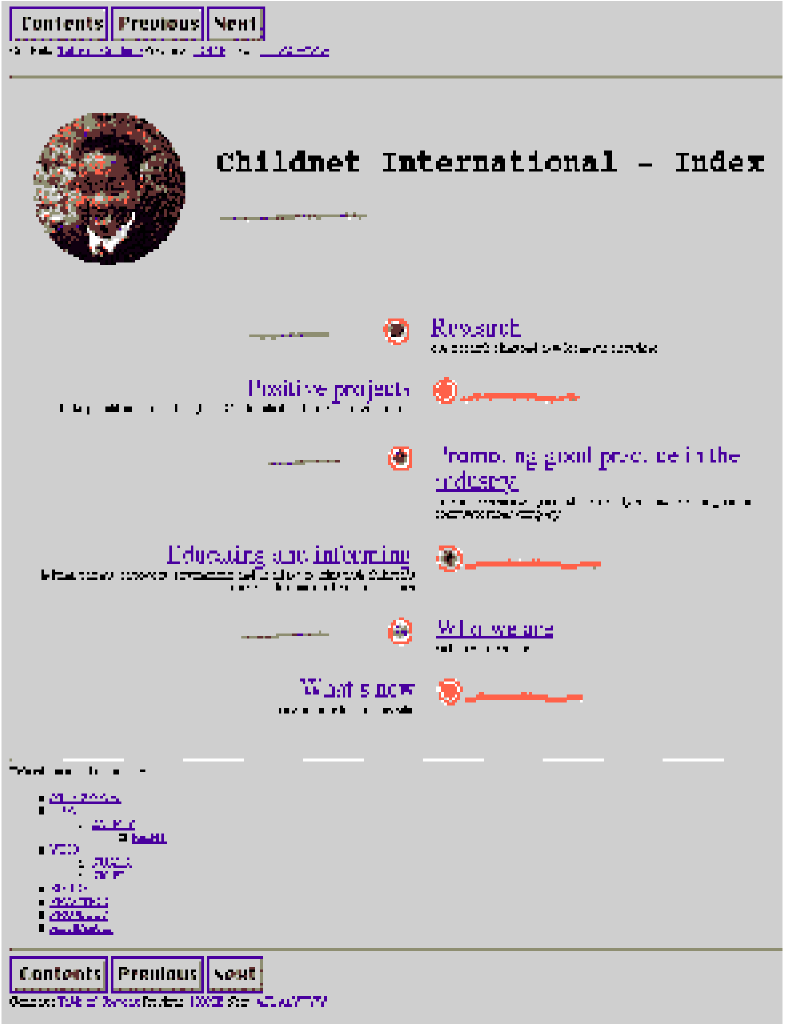

Figure 7: A web page generated by HyperAT, showing the automatically generated navigational information; compare with Figure6.

5.3.2 Getting Feedback From Hypertext Designers

We selected three subjects which consisted of two researchers and one lecturer to help us to evaluate HyperAT. The researchers worked at the Interaction Design Centre (Middlesex University), and their research project was to design and develop the Royal Society of Arts web pages. They were chosen because they were experienced in web design. The lecturer was from the School of Computing Science (Middlesex University) teaching undergraduates web design. He was cho-

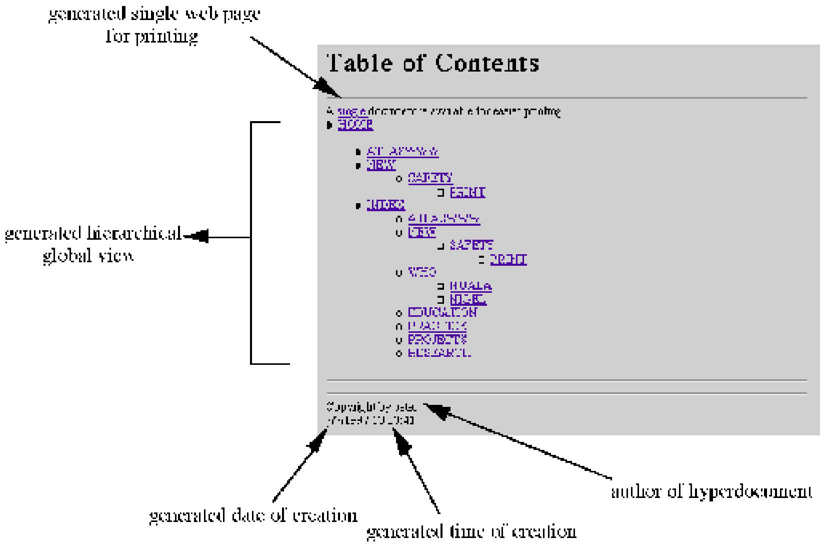

Figure 8: Generated "table of contents."

sen to take on the role of a less experienced designer and thus able to highlight potential problems novice designers may encounter.

Each of the subjects was given a 15 minute guided tour of HyperAT covering both the authoring and usability components of HyperAT. They were then asked to give feedback on whether the authoring features implemented in HyperAT were useful: creation of nodes and links; generation of map and structure; editing facilities; tracking and checking; testing and evaluation; and capturing users' browsing pattern. The subjects were also asked to comment on the usability of HyperAT.

The feedback gathered from the subjects indicates that HyperAT is a useful tool for designing usable web documents. The subjects indicated that all the features implemented in HyperAT are useful in helping designers develop usable web documents. They found the built-in functions for the creation of nodes and links useful. The subjects found the hierarchical structuring simple and intuitive. Because HyperAT separates page content from structure, the subjects felt it would help them concentrate on the content. The automatic generation of overall map and structure was a good feature to provide designers with a graphical overview of the web document. The subjects were also impressed that many processes in HyperAT were automated. These include the automatic structuring of web pages, generated "table of contents" and generated trace of created nodes. The subjects liked the different levels of usability testing HyperAT can perform. For example, facilities to check the structural inconsistencies and complexity were seen as helpful features. The ability to switch from the author mode in the HyperAT environment to the user mode in the Netscape environment was perceived as a useful feature. What makes an impression on the subjects is the idea that HyperAT is capable of parsing server log files, and analysing them to provide information about users' browsing pattern. They also liked the idea of HyperAT parsing and converting an existing website into a structure recognised by HyperAT.

With regard to the usability of HyperAT, it was difficult for the subjects to make any conclusive remarks without actually using the tool. However, the impression they obtained from the guided tour was that HyperAT appeared to be fairly easy to use, once they got used to working with it.

The subjects also made some suggestions to further improve HyperAT. These suggestions basically involve making HyperAT more sophisticated in order to handle even more complex designer needs. One comment against HyperAT is that though the hierarchical structuring process to create nodes and links is intuitive, it may be too restrictive. The suggestion is to provide designers with meta-templates to select the different structuring processes. It is also suggested that different templates could also be provided for designers to select the different visual styles of web pages. Another feature that needs further enhancement is to allow for easier modification of the structure based on a "drag-and-drop" feature, and that updating could be done in real-time. To make the analysis of the transaction log files more useful to designers, it is suggested that the map should show the most common paths based on analysis of the transaction log files.

5.3.3 Comparing HyperAT With Other Tools

The task of comparing HyperAT with other tools, be it commercial or research, is not an easy one owing to a plethora of tools available. We carried out a comparison of HyperAT with four selected tools (two commercial and two research) to highlight the benefits and limitations of HyperAT.

Compared with HyperAT, NetObjects Fusion is a more powerful page editor and site management tool, allowing designers to create, view and modify site structure graphically. Like HyperAT, it also gives semantics to the relationships between web pages. Although NetObjects Fusion provides analysis of the web documents, there is no attempt to integrate these analysis to make the necessary changes to the design of web documents. Microsoft FrontPage is another popular tool for managing websites and editing individual pages. Similar to HyperAT, it automatically generates scheduled inserts, table of contents, etc. Unlike NetObjects Fusion, it does not allow for direct alteration of site structure. It provides more utilities over the whole website such as spell checking and external link verification, compared to HyperAT.

WebWriter shares a similar view with HyperAT in that both are integrated systems for constructing web applications that support the creation of web pages. Crespo and Bier explain that inspired by the HyperCard system, WebWriter allows designers to build an application as a sequence of web pages, where each web page can contain text, images, HTML forms, and the content that is computed on the fly by WebWriter scripts [Crespo and Bier 1996]. HyperAT in allowing users to construct an application through a simple form-like interface automatically creates parent-child relationships between pages, instead of linear relationships captured by WebWriter. However, WebWriter's direct manipulation feature makes editing easier and less cumbersome compared to HyperAT.

Gentler is a web authoring tool developed to overcome authoring difficulties encountered by designers [Thimbleby 1997]. To make web authoring more systematic, Gentler uses a database of pages and a page layout language, as well as reliable design features including hypertext linkage and navigation. Like HyperAT, Gentler aims to reduce the cognitive load of authors, particularly by separating content and design, and by supporting quality control. Compared to HyperAT, Gentler provides a more powerful, flexible support for mathematical analysis of the hypertext structure. However, HyperAT has the ability perform more than just formal analysis of the hypertext structure. It has a program parser to read and analyse users' log files in order to understand users' browsing behaviour and goals. It is believed that by providing another way of conducting usability testing within HyperAT, designers may be able to gain useful insights into hypertext structural designs relating to cognitive issues from different angles based not only on formal analysis but on real users' behaviour.

6 Conclusions

This paper described HyperAT, a research tool to help designers develop usable web documents. On the basis of our experience developing it, we argued for a move from treatment to prevention; from treating users' symptoms, to avoiding bad design.

Initial evaluation shows that HyperAT does a useful job as a research tool in exploring authoring and usability issues. Compared with other tools, a distinguishing factor of HyperAT is that, besides providing the basic authoring facilities to produce web documents, it delivers usability results to designers regarding any usability problems that might be detected during its analysis. Because both the authoring and usability components are found within HyperAT, it would be viable as a future enhancement to HyperAT to integrate the usability results into the authoring process to allow designers to make necessary changes to the web documents more easily.

Thus we conclude that tools like HyperAT — that structure and analyse web sites, etc., as discussed — are helpful to web authoring, and hence useful for web users.

Work with HyperAT involves a more extensive evaluation with different types of designers (e.g., novice, intermediate, experienced), as well as strengthening the usability environment it can offer to help designers build better, usable web pages. This will be our future work.

References

[Baecker *et al.* 1995] Baecker, R. M., Grudin, J., Buxton, W. A. S. and Greenberg, S. (eds.) (1995), "Design and evaluation," *Human-Computer Interaction: Toward the Year 2000*, Morgan Kaufmann Publishers, 73–91, USA.

[Berners-Lee 1995] Berners-Lee, T. (1995), "Style Guide for Online Hypertext," `http://www.w3.org/pub/www/provider/style/all.html`

[Conklin 1987] Conklin, J. (1987), "Hypertext: An Introduction and Survey," *IEEE Computer*IEEE Computer, **2**(9), 17– 41.

[Crespo and Bier 1996] Crespo, A. and Bier, E. (1996), "WebWriter: A browser-based editor for constructing web applications," *Proceedings of the 5th W3 Conference*, 1291–1306.

[Dix *et al.* 1993] Dix, A. J., Finlay, J., Abowd, G. and Beale, R. (1993), *Human-computer Interaction*, Prentice-Hall.

[Gould *et al.* 1991] Gould, J., Boies, S. and Lewis, C. (1991), "Making usable, useful, productivity-enhancing computer applications," *Communications of the ACM*, **34**(1), 74–85.

[Kim 1995] Kim, S. (1995), "Interdisciplinary Cooperation," In Baecker, R. M., Grudin, J., Buxton, W. A. S. and Greenberg, S. (eds.) *Human-Computer Interaction: Toward the Year 2000*, Morgan Kaufmann Publishers, 304–311, USA.

[Landauer 1995] Landauer, T. K. (1995), *The trouble with computers: Usefulness, usability and productivity*, MIT Press.

[Lewis and Rieman 1993] Lewis, C. and Rieman, J. (1993), "Getting to know users and their tasks," in *Human-Computer Interaction: Toward the Year 2000*, Baecker, R. M., Grudin, J., Buxton, W. A. S. and Greenberg, S. (eds.), Morgan Kaufmann Publishers, 122–127, 1995, USA.

[Lindgaard 1994] Lindgaard, G. (1994), *Usability testing and system evaluation: A guide for designing useful computer systems*, Chapman & Hall.

[Mayes *et al.* 1990] Mayes, T., Kibby, M. and Anderson, T. (1990), "Learning about learning from hypertext," in *Designing hypermedia for learning*, Jonassen, H.D. and Mandi, H. (eds), Springer-Verlag, 13.1–13.24.

[McKnight *et al.* 1991] McKnight, C., Dillon, A. and Richardson, J. (1991), *Hypertext in Context*, Cambridge University Press, UK, 64–104.

[Nielsen 1996] Nielsen, J. (1996), "Relationships on the web," `http://www.useit.com/alertbox/9601.html`

[Nielsen 1995] Nielsen, J. (1995), *Multimedia and Hypertext: The Internet and Beyond*, AP Professional, USA.

[Perlman 1988] Perlman, G. (1988), "Software tools for user interface development," in Helander, M. (ed.), 819–833, as cited in Preece, J., Benyon, D., Davies, G., Keller, L., and Rogers, Y. (1993), *A guide to usability: Human factors in computing*, Addison-Wesley.

[Theng 1997] Theng, Y. L. (1997), *Addressing the 'lost in hyperspace' problem in hypertext*, PhD Thesis, Middlesex University, London.

[Theng *et al.* 1997] Theng, Y. L., Rigny, C., Thimbleby, H., and Jones, M. (1997), "HyperAT: HCI and web authoring," *BCS Conference on HCI*, HCI'97, 359–378, Cambridge University Press.

[Theng *et al.* 1996] Theng, Y. L., Thimbleby, H. and Jones, M. (1996), "'Lost in hyperspace': Psychological problem or bad design?," *Asia-Pacific Computer-Human Interaction'96*, 387–396.

[Thimbleby 1997] Thimbleby, H. (1997), "Gentler: A tool for systematic web authoring," in Buckingham Shum, S. and McKnight, C. (eds.), *International Journal of Human-Computer Studies* on 'Web Usability', **47**(1), 139–168.

Acknowledgements

This paper is based on Yin Leng Theng's PhD work, funded by the School of Computing Science, Middlesex University (London).