"Lost in hyperspace": Psychological problem or bad design?

Yin Leng Theng, Matthew Jones & Harold Thimbleby

School of Computing Science, Middlesex University
Bounds Green Road,London N11 2NQ
Tel: +44 181 362 5000
Email: {y.theng, m.jones, h.thimbleby}@mdx.ac.uk

Abstract

A pervasive criticism of hypertext systems is that users tend to lose their way. Although much work has been done to address this "lost in hyperspace" problem, it still remains unresolved, even accepted as an inevtaible feature. But using the "lost in hyperspace" problem as an example of usability problems, we argue *usability*problems are not *user* problems. Once viewed like this, more appropriate solutions are applied to address it. We argue that some usability problems have been mis-classified as "psychological" and therefore are only palliated rather than cured. However, we show that there is much scope for new systems approaches that avoid such problems, and which offer considerably more creative ways of going about design of interactive systems.

Keywords

Hypertext, lost in hyperspace, conceptual model, executable user models, multidisciplinary design

1. Introduction

Ever since Vannevar Bush envisioned his hypertext Memex in 1945 (Bush, 1995), many diverse hypertext systems have been designed and built to meet a variety of functions: tutorial and educational needs, support for collaboration among a number of individuals, to production of on-line manuals. In hypertext, users not only benefit from the information they read but also from the richness of associations supported by the network of nodes and links. However hypertext systems are often criticised since in practice it is found that users readily lose their way in the network. After a few minutes a promising network of associations becomes a maze, with a lost user who has forgotten what he was trying to do. Ironically, the more useful a hypertext, the sooner a user gets so distracted he gets lost!

Fifty years later, Bush's Memex idea finds routine application in almost all modern CD-roms as well as in the World Wide Web. The usability of hypertext is a real issue. Even tiny usability gains for each of the millions of the Web's users would collectively aid humanity significantly!

In general, the "lost in hyperspace" phenomenon refers to any of the following conditions: users cannot identify where they are; users cannot return to previously visited information; users cannot go to information believed to exist; users cannot remember what they have covered; and users cannot remember the key points covered. The last point is worst: the others might have been bearable had users been able to get value from the experience even though it felt like being lost. Users really are lost; it is not just a superficial disorientation. The name of the problem, which we abbreviate *LIH*, is very appropriate! No wonder the most popular forms of system to design are computer games, where getting lost is made into fun - in the tasks fo which hypertext is promoted, though, it is *not* fun, a point well made by Carroll (1982).

2. The conventional design process

All user-centred methods can be recruited in design to reduce the impact of LIH:

- *Getting to know users and their needs.* It is a well-known fact 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. Approaches to task analysis and cognitive modelling could be used to help designers build an accurate representation of users' behaviour and actions when they perform or try to perform common tasks such as browsing, information search, seeking references and recall. These findings could be used to guide hypertext design.

- *Early and continual user testing.* Because it is imperative that hypertext designers build into hypertext systems an accurate representation of user models, early and continual user testing is essential.

- *Iterative design.* Interactive systems require iterative design. The most promising approach is to iterate design and evaluation until a satisfactory result is achieved. We need to ensure that good hypertext design guidelines and principles are incorporated into the building of hypertext in the first place (Theng, Jones and Thimbleby, 1995a).

- *Prototyping.* We need better support tools for hypertext authoring to reduce the complexity of hypertext authoring, thus freeing hypertext authors to concentrate on the design and structure of hypertext systems (Theng, Jones and Thimbleby, 1995b). With good support tools, prototyping of hypertext systems can be achieved more efficiently in quicker time.

The conventional approaches above make routine assumptions:

- Users have a wrong or incomplete conceptual model of how information is structured and linked within the hypertext system (Elm and Wood, 1985).

- Users experience 'lack of closure' since they are not able to tell the extent of a network or what proportion of relevant items remains to be seen (Shneiderman, 1992).

- Users face the 'embedded digression problem' where they lose track of digression since they are distracted from the main tasks by lots of interesting information (Foss, 1989).

- Users generally lack the experience in using hypertext for learning, and this makes it difficult for them to remember, consolidate and understand the semantic content of nodes, resulting in a lack of detailed memory of any particular item and an inability to summarise what has been covered (Foss, 1989).

- Users do not have adequate systems, that they should have graphical browsers and query/search mechanisms (e.g., Edwards and Hardman, 1989, etc.).

- Users do not use their senses fully. They should have multi-modal experiences, fully using sensory and motor skills, audio, stereolocation and so on. This view leads to solutions being sought in virtual reality.

Note that all assumptions are referenced to the users' supposed failings: *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"; *users* don't understand the chosen display conventions; etc. And because we know that 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, *etc.* - that no doubt seems to confirm it.

However, the repeated emphasis on *user* problems (ironically for such a promising technology!) seems suspicious. It seems to us, rather that *anything* would get lost in current hypertext systems. If so, then the problem is not based in user perception nor in user psychology. We might note that successful systems tend to be realistic (e.g., simulation environments) and that better realism is always attainable through incremental expenditure on the equipment. For example, by using a faster processor, bigger disc and a higher resolution display, then better pictures can be displayed. As systems get better *in this way* they get more expensive. It is possible, then, that cognitive dissonance influences the popularity of psychoperceptual (realism) solutions to LIH: the more expensive a solution, the less likely anyone is to question its appropriateness, especially when spending a little more on it would make it better. (This is also hill climbing, a simple problem solving strategy that fails in all but the simplest cases.)

Note. It seems to us that anything would get lost. By "anything" we mean any device capable of algorithmic thought. To show that anything in this sense gets lost, then, we need to show that not being lost invokes non-computable functions, which we do by contradiction.

Consider (as just one possible way of arguing) the analogy of a Turing Machine (*head, tape)* with User (*head, hypertext)*, from which it follows that if what the user does with the hypertext is computable then, in particular, it is computable in finite memory (in fact, if the hypertext is writable, only a two state memory is required). Now consider a computable function, *f* : screen display X screen display -> user actions, with the intended meaning that given one display and an intended display, *f* gives the user actions that take the hypertext from one screen to the other. To evaluate *f*, the user must be able to distinguish screens. We can clearly construct a hypertext (indeed, a realistic hypertext!) having more screens than the bounded state space the user's head can distinguish. Therefore *f* is not computable under the assumptions (recall that it must be computed by the user).

One might argue that it is necessary for the user to distinguish screens since any well-designed hypertext would provide appropriate labels on screens, and labels on buttons (or other potential actions, such as menus) leading to screens. However, this only makes *f* trivial fo adjacent screens, or for screens a bounded number of steps apart. In the limit, if each screen refers to every other then *f* is computable if it is computable for certain hypertexts (rather dull ones at that; being complete graphs they have no organising structure); in general, *f* remains uncomputable.

The Church-Turing Thesis suggests that human users have no way around this non computability. As *f* is not computable, users will at best have a tough time deciding waht actions to take, they can only be given approximate advice by help systems, and they will readily become lost.

The Church-Turing Thesis is rather strong. More psychological arguments can be found in Johnson-Laird (1993), which suggest that users are even more limited than our arguments to show the inevitablity of LIH required. *End of note.*

3. The central issues

Hypertext design is hard, and hypertext systems are used less effectively than we would wish. Is the problem

- *A design problem*? Design (for whatever reasons) is done poorly. Poor design causes psychological problems.
- *A user problem*? The problem may be entirely due to users' inability to exploit computer screens, complex information structures, and that nothing in the design is to ameliorate this. Thus, as a psychological problem it can be alleviated but not solved by better design.

Most people think it is the latter, with the result that improvements are sought in presentation of the information.

We argue that there is more truth in the former view. We believe users now enter a world of distractions (of dramatic sounds and images) and tend to blame themselves when they get lost.

3.1 The book analogy

It is interesting to contrast this state of affairs with book design:

- book design is also difficult (there are good and bad authors)

- but readers can and do make accurate judgements of a book's quality. They make these judgements easily. They do not assume that the difficulty of understanding a badly designed book is their own fault.

- a book (or any other serial medium, like a film) can be evaluated in reasonable time, whereas a hypertext can take practically unlimited time to evaluate. Every time the designer adds a choice for the user, the user's job of understanding the book can double. After only twenty choices, the user can no longer model the complexity of the hypertext.

- we take for granted very many organisational structures in books, like page numbers and alphabetic orders in reference books. Though they can be used in hypertext, none are useful *central* organising principles in hypertext - because if they were used the generality of hypertext is defeated, and we've just got an electronic, conventional book.

The book's organisational principles took centuries to develop (Thimbleby, 1992). How can we more quickly find better organisational principles for hypertext?

We now 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.

3.2 Users are robots

For too long we have thought of users as being humans. Humans are remarkably adaptive, know a lot about the world, and adhere to deep conversational conventions. Yet in hypertext we wish to construct "virtual worlds" (virtual or metaphorical) with which the user has no prior experience. Indeed if users knew some hypertext world well, there would be less point in using it (we are making assumptions about what hypertext is for) - it is supposed to be an enhancement to users, not a substitute for something they already have.

When a function is non computable, it has to be evaluated by some extra-computational method. Since users are humans, it follows that hypertext systems (as presently conceived) *rely* on human skills - and humans' perseverance and their willingness to put up with bad systems. It would be better to design systems that in the first place did not rely on human skills to disguise their poor design. Instead, systems should be designed for robots to use. And only when it is *known* that a robot can use a system should additional presentation be added to make the system more attractive for human appreciation. At present we commonly make systems the other way around: we make them stunningly attractive, and then wonder how to make them usable. Often designers don't get that far, because the attractiveness is sufficient to disguise systems' fundamental usability. If a programmer designs a program, only half the job is done if they have only designed the data structures. They also have to design the procedures for operating on the structures. (Specifically, a programmer designs abstract data types.) Without the appropriate procedures for operating on data structures, a computer would literally get lost in the structures, even supposing it could start executing anything sensible. Notice that hypertext is always defined as a data structure! There is no emphasis on the user's procedures for navigating, finding information, or doing any general operations (Thimbleby, 1992). Therefore, users use whatever procedures "come to mind" - which we've suggested above are likely to be inappropriately carried over from the real world. The result is LIH.

3.3 Robots are not enough

We argued that treating users at least as well as we treat robots would be a good idea (see also Thimbleby, 1995c). Even so, standard user-centred practice in usability engineering is still required; because all the arguments for usability engineering are still valid (Nielsen, 1993). Unfortunately statistically useful involvement of users in hypertext design is infeasible (hypertext systems are combinatorially too large). Given our robot-oriented design stance, it is natural to suggest that what designers need is robots to model users. We call such "robots" executable user models.

Executable user models are software agents that simulate users. They can be faster, tireless, and they do more exhaustive checking of system designs. Moreover, they can embed multi-disciplinary knowledge that most designers and most users would not be expected to know or be able to verbalise in their accounts of interaction. Executable user models can work with hypertext prototypes, long before they have reached a stage where actual human user evaluation would be practicable. It is even possible for executable user models to be in iterative design cycles, managed by suitable artificial intelligence strategies.

4. Not merely difficult, but full of opportunities

We've argued that hypertext design is difficult, and viewing it as primarily a computational problem does not make it better. What it does do, though, is open up a very large range of potential solutions that have, so far, not been explored.

We mention three possibilities.

First, viewing the user as a robot suggests we search for appropriate algorithms to implement whatever tasks the user is engaged in. Certainly, algorithms books are much larger than the range of features that current interactive systems provide, so there are plenty new user interface styles to choose from! Such ideas have been explored at greater length elsewhere (Thimbleby, 1990 & 1992), where examples may also be found.

Second, computerised users in the form of executable user models could be used to simulate users' behaviour to help designers evaluate their systems without requiring (human) user attendance. Metrics on user performance, cognitive overload and satisfaction could provide approximations on the usability of the hypertext systems, which would pinpoint weaknesses of their systems and reduce the length of the evaluation process. We have started explorintg such approaches with our colleague Cecile Rigny who has implemented a LISP tool for running cognitive models, called CUM-DesTool. The LISP model simulates user cognition, but it is also a program that can run interactive systems, and therefore substitute for actual users in evaluation. Moreover, by randomising user models, it is possible to simulate large groups of users and to obtain useful statistical information.

Yin Leng Theng has built a hypertext prototyping system, called HyperAT, also written in LISP. We are now in the process of combining the two tools to explore the ideas proposed in this paper.

Third, it is clear that *ad hoc* methods of designing, constructing and validating hypertext systems are not enough. If users get lost in hypertext, designers do too. (The theoretical arguments are easier than for users: having an effective procedure for designing hypertext systems where users do not get lost is equivalent to the halting problem.) This suggests that tools for designing hypertext should provide much improved computational support for designers (Thimbleby, 1995a).

5. Conclusions

Although much research effort has been invested to address the LIH, it still remains unresolved. We argued that wrong or inappropriate solutions are being sought because incorrect or incomplete assumptions are made. Disorientation can arise in conceptual space (within the user's mind), which most research findings support - but we argue the "blame" should not rest on users alone! LIH is surely attributable to bad system design, and is not a psychological *problem* - it is just a psychological *symptom*. This paper has set down the reasons for a new approach designing hypertext.

We furnished an outline theoretical argument about the computability of the user performing certain tasks within hypertext. This line of enquiry could be extended. For example, it would be worthwhile exploring the complexity of the user performing certain actions and how to design hypertexts to optimise the user's performance (Thimbleby, 1995b). It would also be interesting to explore other computational models: even the simple Turing Machine model suggests that having user-writable hypertexts would make the user/hypertext system more expressive.

Since LIH is a complex problem, a multi-disciplinary approach is necessary to draw upon and integrate knowledge and findings in seemingly diverse disciplines such as Cognitive Psychology, Artificial Intelligence and Software Engineering, and to integrate this knowledge into human-computer interaction. This paper has shown some ways of doing this.

References

Bush, V (1945), "As We May Think," *Atlantic Monthly*, **7**, pp. 101-108.

Carroll, J M (1982), "The adventure of getting to know a computer," *IEEE Computer*, **15**(11), pp. 49-58.

Edwards, D M and Hardman, L (1989), "'Lost in hyperspace': cognitive mapping and navigation in a hypertext environment," *Hypertext: Theory into Practice*, Intellect Books, pp. 90-106.

Elm, W and Woods, D (1985), "Getting lost: A case study in interface design," *Proceedings of the Human Factors Society 29th Annual Meeting*, pp. 927-931.

Foss, C L (1989), "Tools for reading and browsing hypertext," *Information Processing and Management*, **25**(4), pp. 407-418.

Johnson-Laird, P N (1993), *Human and machine thinking*, Lawrence Erlbaum Associates.

Nielsen, J (1993), *Usability Engineering*, Academic Press.

Shneiderman, B and Kearsley, G (1989), *Hypertext Hands-On! An Introduction to a New Way of Organising and Accessing Information*, Addison-Wesley.

Theng, Y L , Jones, M and Thimbleby, H (1995a), "Reducing information overload: A comparative study of hypertext systems," *IEE Colloquium on "Information Overload*," Digest No: **95/223**, pp. 6/1-6/5.

Theng, Y L , Jones, M and Thimbleby, H (1995b) , "Designer tools for hypertext authoring," *IEE Colloquium on "The Authoring and Application of Hypermedia-Based User Interfaces,"* Digest No: **95/202**, pp. 4/1-4/4.

Thimbleby, H (1990), *User Interface Design*, Addison-Wesley.

Thimbleby, H (1992), "Heuristics for Cognitive Tools," in NATO ASI Series **F**, *Proceedings NATO Advanced Research Workshop on Mindtools and Cognitive Modelling, Cognitive Tools for Learning*, Kommers P A M, Jonassen D H & Mayes J T, editors, pp. 161-168, Springer Verlag.

Thimbleby, H (1995a), "Hypertext authoring without getting lost," *HCI'95 Adjunct Proceedings*, pp. 118-124.

Thimbleby, H (1995b), "Users as computers": An approach to VR design and conceptual evaluation, *Proceedings Interface to Real and Virtual Worlds*, **IV**, pp. 305-313.

Thimbleby, H (1995c), "Treat People Like Computers?" *Extraordinary People and Human-Computer Interaction*, Edwards A, editor, Cambridge University Press, pp. 283-295.