Improved Conceptual Design For Better Hypertext

Yin Leng Theng, Cecile Rigny, Harold Thimbleby & Matthew Jones

School of Computing Science, Middlesex University
Bounds Green Road,
London N11 2NQ

Tel: *+44 181 362 5000*
Fax: *+44 181 362 6411*
EMail: *{yin2, cecile1, harold, matthew16}@mdx.ac.uk*

This paper discusses an iterative, engineering approach with an improved Conceptual Design Stage for the design of better hypertext. We emphasize the importance of getting design right "up front" so that many design blunders can be avoided. Task analysis and cognitive user modelling techniques are used to make design recommendations for rapid prototyping and testing of a prototype hypertext. We also explore the potential of executable user models as a cost-effective means to rapidly iterate and test design, without the attendance of real users.

Keywords: Hypertext, cognitive user modelling, iterative design, conceptual design

1 Introduction

Hypertext systems are hard to design and develop successfully. This is because hypertext systems consisting of a rich network of interconnected nodes and links present to hypertext authors design choices that are difficult to manage [8]: vast number of potential structures to create hypertext systems; and astronomical number of ways to create links. Because designing and producing hypertext is a relatively new discipline, there are few established tools or procedures that designers can readily use to assist them build well-structured hypertext systems [3]. As a result, most hypertext systems are poorly designed and built in terms of how information is structured and displayed.

One of the problems is the lack of a disciplined and systematic approach to designing well-structured hypertext systems that will meet users' needs [3]. We need a cost-effective way of designing that adheres to good usability practice yet avoids as many design errors as soon as possible. The more errors that can be avoided "up front" by the right method, the less work both test users and designers will have to put in to make prototypes acceptable. If conventional iterative development process is found lacking, we need better ways to ensure that better hypertext systems are produced. Figure 1 shows the six stages described in the well-accepted iterative development process.

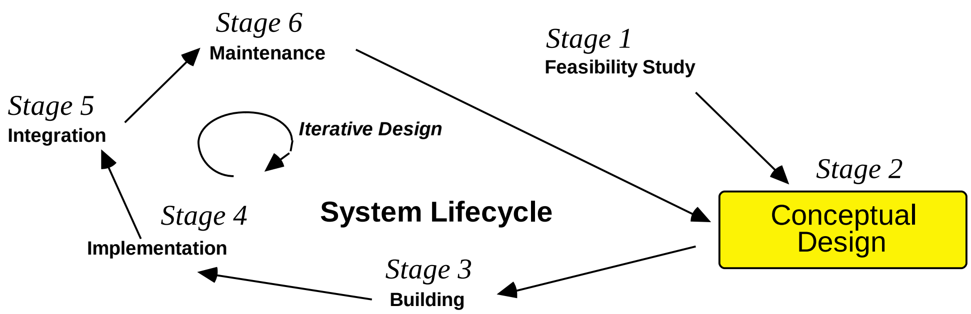

Figure 1. Conventional iterative development process

This paper argues *Conceptual Design* (Stage 2 of the conventional iterative process) can be usefully systematised (see Figure 2). We present our work including new tools, prototypes and studies, justifying this claim, and its encouraging preliminary results. We believe Conceptual Design is vital to the success or failure of the final system as it encompasses all activities relating to gathering and analysing users' needs, specifying users' requirements, building and testing prototype. This paper discusses how some of these activities can be systematically and effectively achieved. Our objectives include:

1. analysing tasks users want to perform (or try to perform) using task analysis techniques, leading to the building of cognitive task graphs, which are then used for design recommendation for building a prototype hypertext;

2. investigating whether a prototype built with the design recommendations from task graphs performs better than a prototype built without using the design recommendations; and

3. investigating the potential of using executable user models in place of real users for rapid prototyping and testing.

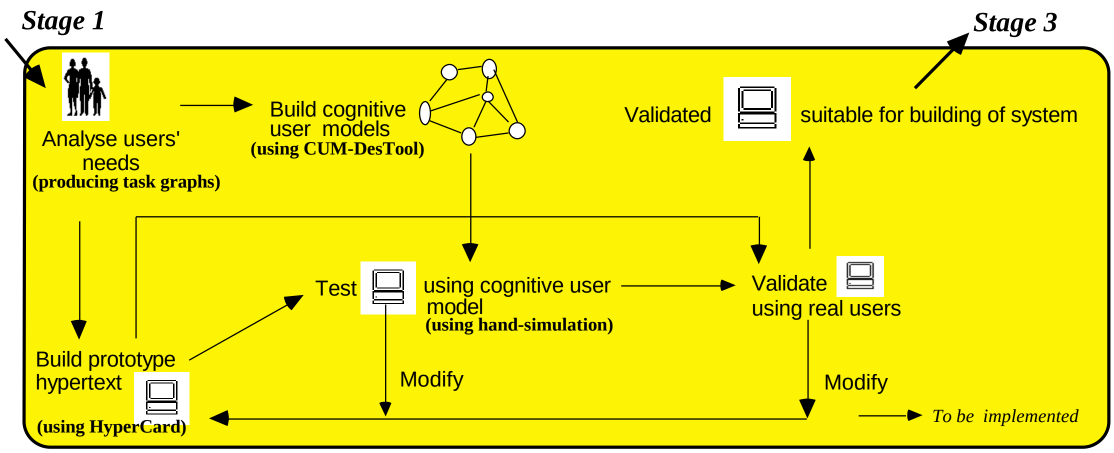

Figure 2. Improved Conceptual Design for better hypertext

2 Investigation I*

To achieve Objective 1, we need to know users and their needs by analysing the common tasks they want to perform or try to perform [e.g., 4, 5, etc.]. Based on the kinds of support provided by hypertext systems, we identify four representative tasks: browsing; information search; seeking references and recall. As an illustration, we have selected one out of these four: the task we call *browsing* to illustrate how cognitive task graphs were built. (Cognitive task graphs for other tasks can be developed similarly.) Generally, *browsing*, as we use the term throughout, refers to reading and navigating in hypertext without a definite goal to accomplish. In this paper, we also include focussed browsing. In focussed browsing, we are saying that users browsing hypertext systems generally have an idea of what they want to do to accomplish a goal.

To gather inputs for building the cognitive task graph for browsing, we performed the following activities: started with a rough cut of the task graph for browsing based on known facts about browsing and our experience with hypertext systems. Figure 3 shows the task graph for browsing and the possible subtasks that are associated to it: (1) starting; (2) getting general information; (3) understanding the system; (4) navigating; (5) finding out what topics or areas are covered in the system; (6) finding a particular topic; and (7) quitting.

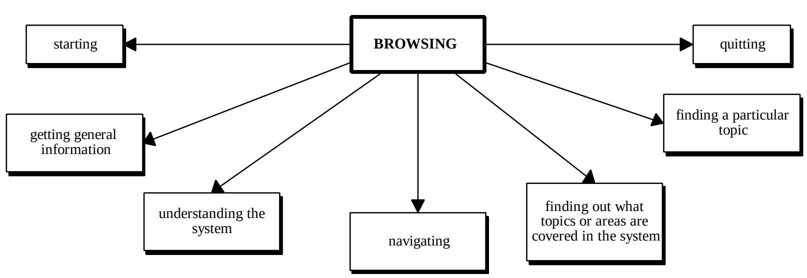

Figure 3. Task graph for browsing

* Investigation I is described in greater detail in [7]. It is presented in this paper to provide the framework for Investigations II & III.

To validate and refine the task graph for browsing, we recorded two "typical" users' behaviour and actions when using *ACM Hypertext-on-Hypertext* [1], a hypertext system, to complete some exercises associated to browsing. By "typical", we refer to a user who has general knowledge about navigation with hypertext systems. Through conducting this video protocol, we were able to interview users after the experiment and ask them to explain step-by-step why certain decisions and actions (e.g. moving to next screen, going to "Home" screen, moving to the "Table of Contents", etc.) were taken while browsing *ACM Hypertext-on-Hypertext*. Their inputs helped us understand users' reasoning and learning processes when they chose to perform certain actions. The task graph for browsing was then modified and refined accordingly.

Using the task graph for browsing, we built a prototype hypertext on *Basic Computer Anatomy* in HyperCard. Considerations for the design of the prototype included: how best to support and improve the various aspects of the task browsing; how to support the way users could carry out actions while browsing hypertext; how to identify how much information should be provided on the screen at any one time, as well as give recommendations and suggestions on how to present information on the screen. Figure 4 shows a visual representation of our interpretation of the design of a simple prototype hypertext produced from the cognitive task graphs. For example, if a user wants to perform browsing, a pop-up menu is shown which recommends the kinds of activities normally associated with browsing, as highlighted in Figure 3.

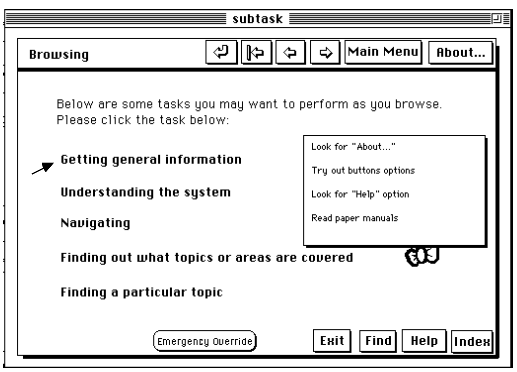

Figure 4. Interface design of a simple prototype hypertext

3 Investigation II and preliminary findings

To find out whether a prototype hypertext built using the design recommendations in Section 2 is better than a prototype hypertext built without the design recommendations (see Objective 2), we built another prototype hypertext on *Basic Computer Anatomy* in HyperCard using the conventional iterative "intuitive" development process recommended by Beekman [2]. We then selected a "typical" user to evaluate the two prototypes. She was asked to complete some exercises associated to browsing. She was then interviewed and asked to complete a questionnaire on the performance of the prototypes.

There was no significant difference in terms of her performance when using the prototypes as she was able to complete her tasks successfully in both cases. However, she indicated that she was more satisfied with the prototype that incorporated design recommendations such as quick tips for browsing, which included recommendations for possible activities normally associated to browsing (described in Section 2, Figure

4). She also rated it to be more effective, and as such she was more confident using it. These findings, though not entirely surprising, confirmed that users' needs can be straightforward if they are systematically analysed through using cognitive task graphs as shown in Section 2. Based on her feedback, we modified the prototype with the design guidelines. We then asked the same user to evaluate the modified prototype and she was more satisfied with its performance, compared to the other two prototypes.

4 Investigation III and preliminary findings

It appears that the approach described in Sections 2 & 3 is cost-effective: an expert's initial task graph, then protocols of only two human users, leads easily to an improved task graph. From the improved task graph, a prototype hypertext was built, which was evaluated to be effective in helping a human user complete her tasks successfully. Whether the additional cost of obtaining better task graphs and hence better prototype would in fact be worthwhile is a separate issue; we suspect there would be diminishing returns, especially when delay is considered.

To reduce the use of extensive and time-consuming real user validating and refining the task graphs, CUM-DesTool (Cognitive User Model Design Tool) has been developed to enable designers to automate the task graphs. CUM-DesTool's structure refers to ACT and the memory model proposed by Reason (1988) and is intended to enable a rapid and easy implementation of users' models. CUM-DesTool is a general simplified architecture to build operational cognitive user models which enable designers to understand, predict and simulate users' behaviour and performance [7].

Figure 5 shows the general interface of the executable cognitive user model for browsing generated by CUM-DesTool. The alphabetical list of concepts that have been implemented is displayed. When a goal is "set to true," for instance the browsing goal, the corresponding module of the simulation of the user's reasoning and activity is triggered. The trace of the reasoning mechanisms is stored and can be used to help designers understand users' behaviour (see Table 1). By inputting into CUM-DesTool parameters such as users' profiles and tasks, different cognitive user models can be created. These models can then be used in *Stage 2* (see Figure 2) to build, test and modify prototype.

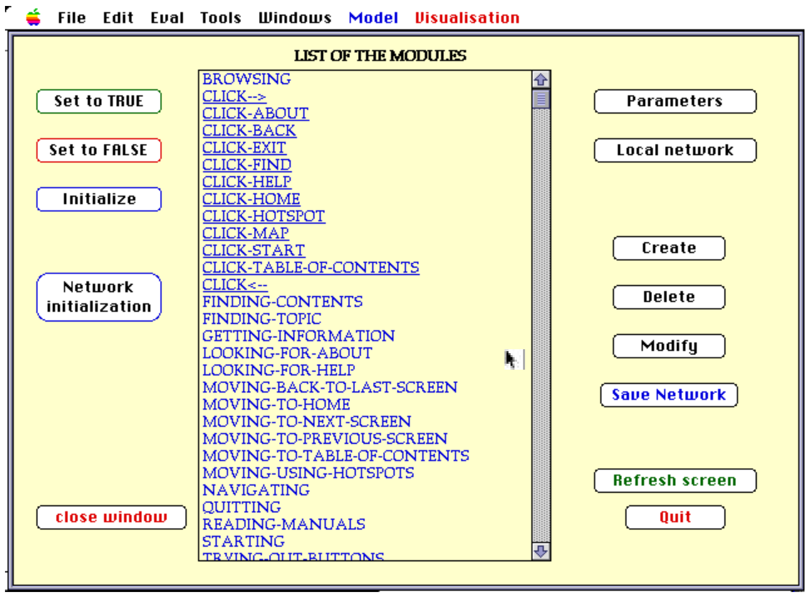

Figure 5. Example model description screenshot

To investigate whether CUM-DesTool is able to produce accurate enough executable user models (see Objective 3), we carried out the following experiment. We used the three prototypes built as described in Investigations I & II: one without the design guidelines; one with the design guidelines; and one with the design guidelines (improved based on user's feedback obtained in Investigation II). From Investigation II, we asked the user to write down the steps taken to perform the tasks using the three prototypes. We also recorded the user's feedback on what features were lacking in the prototypes. Using the executable user model for browsing, we carried out a hand-simulation of the user's behaviour to evaluate and analyse the usability of the three prototypes, by performing similar tasks the human user had to perform in Investigation II.

Table 1 shows the steps taken by the human user and executable user model in obtaining the answer to a particular task, for example, *finding a topic*.

Goal to perform: **Finding a particular topic**	Steps taken by **human user**	Steps suggested by **executable user model** (obtained from the *trace* and simplified for presenting below)	**Remarks**
(I) Prototype **without** the design guidelines	Look for "Find" button. Click "Find" button. Type in query. Select the relevant topic.	Browsing =>Finding a topic Finding a topic => Click-Find Click-Find => Type-Query Type-Query =>Select-topic	In the hand-simulation, the steps suggested by the executable user model matched the actions taken by the human user. This is because the "Find" button was found on the first screen of the prototype.
(II) Prototype **with** the design guidelines	Look for "Find" button. Click "Find" button. Type in query. Select the relevant topic.	Browsing =>Finding a topic Finding a topic => Click-Find Click-Find => Type-Query Type-Query =>Select-topic	In the hand-simulation, the steps suggested by the executable user model also matched the actions taken by the human user. This is because the "Find" button was found on the first screen of the prototype.

Table 1. Steps taken by the human user and executable user model in *finding a topic*

Goal to perform: **Finding a particular topic**	Steps taken by **human user**	Steps suggested by **executable user model** (obtained from the *trace* and simplified for presenting below)	**Remarks**
(III) Prototype **with** the design guidelines (**modified** based on user's feedback)	Look for "Find" button. Can't find button. Click on the next screen button. Look for "Find" button. Can't find button. Click on the next screen button. Look for "Find" button. Click "Find" button. Type in query. Select the relevant topic.	Browsing =>Finding a topic Finding a topic => Click-Find Click-Find not satisfied. Finding a topic => Looking-For-Table-Of-Contents Looking-For-Table-Of-Contents not satisfied. (Owing to space contraints, we will not list down all the possible buttons suggested by the user model such as "Back", "Previous", etc. When the suggestion was "Next", the simulation continued with going to the next screen of the prototype. The "Find" button was also not available in the second screen. The whole process continues for the second screen.) In the third screen the goal was satisfied because the "Find" button was found.	Based on user's feedback, the prototype was modified with the "Find" button only appearing in the third screen. The first two screens only provided general information about the system, which the user thought should only contain the "next" button. It is interesting to note that when the hand-simulation was performed, it took a number of cognitive steps to reach the third screen with the "Find" button. The goal was then satisfied like in (I) and (II).

Table 1(cont'd). Steps taken by the human user and executable user model in *finding a topic*

From Table 1, we note that the executable user model suggested steps that matched the way the human user would do to complete the task *finding the topic*. The goal was satisfied when the "Find" button was found. Prototype III, which was modified based on user's feedback, had the "Find" button removed from the first two screens. Therefore, it took us a number of cognitive steps to perform the task completely, when we carried out the hand-simulation. Whether the user was right in suggesting that the "Find" button was to be removed is debatable. What is interesting is that the executable user model randomly provided a list of suggestions that the human user would have in mind when performing the task *finding a topic*. This list of suggestions could provide valuable insights for designers to ensure the usability of the prototype. Taking away the "Find" button may make one user happy but may also make other users unhappy, if so many cognitive steps had to be taken to perform the task successfully.

We based our design of the prototype hypertext on the recommendations of one "typical" user. However, it may be presumptuous to assume that all users behave like her. Although CUM-DesTool is in its early stages of development, initial results are encouraging. We see immediate benefits to the designers: validated executable user models generated by CUM-DesTool can be used to generate different user models, rapidly iterate and avoid many design blunders, thus reducing the use of extensive and time-consuming human users validating and refining them.

5 Conclusion and future work

This paper presents an improved *Conceptual Design Stage* built on top of the conventional iterative development process. Through the Conceptual Design Stage, we have presented ways of understanding users' behaviour and navigation issues when browsing in hypertext by giving them more structure using cognitive task graphs. We have shown that experts' initial task graphs can be easily refined with real users. Based on the task graphs, we have implemented a prototype hypertext system with the design guidelines recommended. Through user's studies, we have shown that it was rated better than the prototype without design guidelines. We have also shown that the resultant task graphs can be interpreted by CUM-DesTool, a cognitive modelling tool, to produce executable user models. We then compared the performance of the executable user model for browsing with real users. There is, thus great potential in using executable user models for rapid prototyping and testing, without requiring real users' attendance.

Subsequent work will involve building different cognitive user models for different user types (e.g., novice, intermediate, experienced) and different tasks (e.g., information search, seeking references and recall).

Acknowledgements

The authors would like to thank the users for taking part in the experiment, Ann Blandford and Mark Addison for their helpful comments in the earlier version of the paper.

References

[1] ACM (1989), "ACM Hypertext-on-Hypertext," ACM Press Database and Electronic Products Series.

[2] Beekman, G. (1995), *HyperCard 2.2 in a hurry: The fast track to multimedia*, Wadsworth Publishing, U.S.A.

[3] Glushko, R.J. (1995), "Seven ways to make a hypertext project fail," In Baecker, R.M., Grudin, J., Buxton, W.A. and Greenberg, S. (Eds), *Readings in Human-Computer Interaction: Toward the Year 2000 (Second Edition)*, Morgan Kaufmann, pp. 849-853, U.S.A.

[4] Hansen, W. (1971), “User engineering principles for interactive systems,” *AFIPS Conference Proceedings*, **39**, AFIPS Press, pp. 523-532.

[5] Lewis, C. and Rieman, J. (1995), "Getting to know users and their tasks," In Baecker, R.M., Grudin, J., Buxton, W.A. and Greenberg, S. (Eds), *Readings in Human-Computer Interaction: Toward the Year 2000 (Second Edition)* , Morgan Kaufmann, pp. 122-127, U.S.A.

[6] Reason J (1988). Framework models of human performance and error: a consumer guide. In L.P. Goodstein, H.B. Andersen & S.E. Olsen (Eds.), *Tasks, errors & mental models*. pp. 35–49, Taylor & Francis: London.

[7] Theng, Y.L., Rigny, C., Thimbleby, H. and Jones, M.(1996), “Cognitive task graphs and executable user models for better hypertext,” *APCHI96*, Singapore.

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