No. 929

Search process evaluation for a hierarchical menu system by Markov chains

by

Hideaki Takagi, Muneo Kitajima, Tetsuo Yamamoto, and Yongbing Zhang

May 2001

Search process evaluation for a hierarchical menu system by Markov chains

Hideaki Takagi*[a], Muneo Kitajima**[b], Tetsuo Yamamoto[c], Yongbing Zhang[a]
[a]Institute of Policy and Planning Sciences, University of Tsukuba
[b]National Institute of Advanced Industrial Science and Technology
[c]College of Policy and Planning Sciences, University of Tsukuba

ABSTRACT

When computers are used to execute tasks, it is often necessary for the user to locate a target item in a menu or a list. For example, users of word processors and spreadsheet applications select appropriate commands in a hierarchical menu to display dialog boxes and edit file or table attributes. To locate the desired information on the World Wide Web, users select the most appropriate candidate out of those presented by a search engine, and proceed through a series of hyperlinks that appear to be related to the task. This paper applies a cognitive model of the user's item selection process to the task of target search in a hierarchical menu system that contains one or more of the following four operations: (1) item selection on the basis of similarity to the task, (2) consideration in various ways of the selection history when making the next selection, (3) backtracking when an appropriate item is not present among those selectable at a given point in time, and (4) abandoning the task unachieved. We model this selection process with Markov chains. We calculate the probability that task goals are achieved and the average number of selections to make until the task goals are achieved. Finally we use these results to propose a method of evaluating the structures of hierarchical menus and links on a website.

Keywords: hierarchical menu, search process, cognitive model, Markov chain

1. INTRODUCTION

In order to accomplish computerized tasks, users must give commands to applications through interfaces. Typical styles of human-computer interaction include menu selection, direct manipulation, and command input. However, a simple look at the interfaces widely used in daily life — web browsers, office applications (word processors, spreadsheet applications, and graphic applications), automatic teller machines (ATM), and remote controls for household appliances — indicates that the predominant interface style is the sequential selection from a hierarchically organized menu or a link structure.

When using a machine to accomplish a task in an ordinary situation, users often do not recall a series of operations from memory before executing them. They would rather comprehend the contents presented on a display and accomplish the task by selecting the appropriate item every time. It has been confirmed by experiments that in menu-based interaction users do not in fact remember the menu items even if the task is performed repeatedly [7]. Users are known to employ a "label-following strategy" to select items in such situations [12]. The primary criterion of selection here is the semantic similarity of the selectable items to the representation of the task. Therefore, when the design of an interface for item selection does not conform to the task in question, one may predict that the task will be difficult to accomplish.

In fact, this has been confirmed in research on the usability of menu-based applications and web pages. For example, to create a new chart in the Excel 3.0 spreadsheet application, the user must select the item "New" from the "File" pull-down menu. But since "File", the item that must be selected, is not consistent with the task goal of "creating a chart", it was extremely difficult to accomplish this task [1]. As another example, when subjects were asked to search for

*takagi@shako.sk.tsukuba.ac.jp; phone 81 298 53 5003; fax 81 298 55 3849; Institute of Policy and Planning Sciences, University of Tsukuba, 1-1-1 Tennoudai, Tsukuba-shi, Ibaraki, 305-8573 Japan; **kitajima.muneo@aist.go.jp; phone 81 298 61 6650; fax 81 298 61 6731; National Institute of Advanced Industrial Science and Technology, 1-1-1 Higashi, Tsukuba, Ibaraki, 305-8566 Japan

some information provided on several websites, they exhibited extremely low success rates ranging from 12% to 43%, because the links they needed to select were not consistent with the information they were searching for [11].

The objective of this paper is to propose an interface evaluation method that will prove useful in designing interfaces that make it possible for tasks to be accomplished with certitude in situations where tasks are accomplished by selecting items from hierarchical menu and hyperlink structures. Previous research on the evaluation of menus has focused on their hierarchical structures and offered a guideline that "menu hierarchies should be broad and shallow rather than narrow and deep" [9]. However, none has addressed the issue of optimal design for the item contents.

The rest of this paper is organized as follows. In 2 we describe a cognitive model of the user's item selection process. In 3 we present a Markov chain to model the item selection process. The structure of a given hierarchical menu is evaluated in terms of such task performance measures as the probability of success in the selection and the average execution time by the numerical calculation of the Markov chain. In 4 we add four models of users. They are divided into a class of high-speed evaluation models with ease of modeling but poor approximation and a class of granular evaluation models which enjoy fine approximation but conduct the evaluation by means of simulation. The task performance of a realistic example of the hierarchical menu is examined using these evaluation models. We discuss the results obtained from both models, and claim that they can be combined to offer a useful tool for the efficient design of hierarchical menus and link structures. In 5 we conclude the paper with a brief summary.

2. A COGNITIVE MODEL OF THE USER'S ITEM SELECTION PROCESS: CoLiDeS

We begin with an outline of the CoLiDeS model, and then discuss in some detail the *similarities* that play a significant role as mediation in modeling search processes by Markov chains.

2.1. Outline of the CoLiDeS model

Figure 1 shows a sketch of the CoLiDeS (Comprehension-based Linked model of Deliberate Search) cognitive model of the user's item selection process. For details of this model, readers can refer to [3], [4], and [5]. Process simulation in the CoLiDeS model is based on the Construction-Integration Architecture [2], a cognitive model of text comprehension. The CoLiDeS model is briefly described in the following.

First, in the **goal generation process**, the user generates several sub-goals in order to accomplish the ultimate goal. For example, for the goal of "browsing books on cognitive science," the user would generate such sub-goals as "visiting a browsing subsite," "selecting a category relating to cognitive science," and "browsing the books available there."

In the **goal selection process**, the user examines the current interface display to select the most appropriate out of the sub-goals generated.

In the **attention process**, the user directs his attention to the specific portion of the display (e.g., a menu bar or a navigation tab) associated with the sub-goal he is attempting to achieve.

In the **action selection process**, the user makes a synthetic evaluation of the objects (e.g., menu items or hyperlinks) to which his attention is directed in terms of semantic similarity [6], textual similarity (literal matching), and the past frequency of selections with similar context; he then selects one of the objects that is the closest to the representation of the sub-goal.

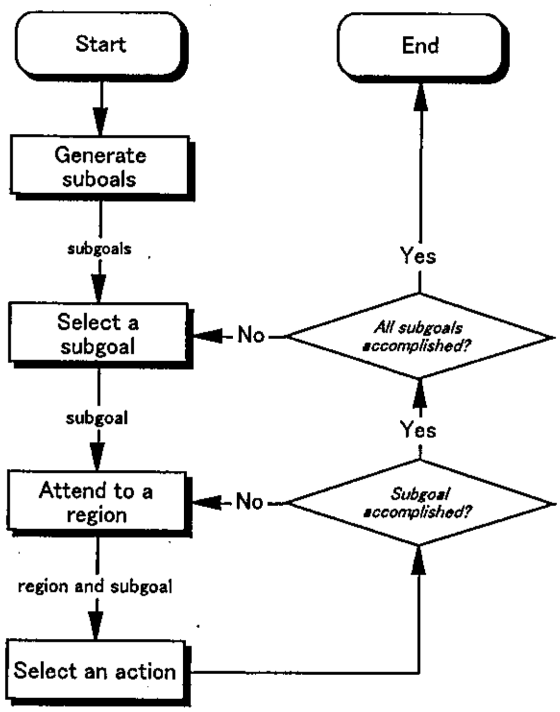

Figure 1: The CoLiDeS model.

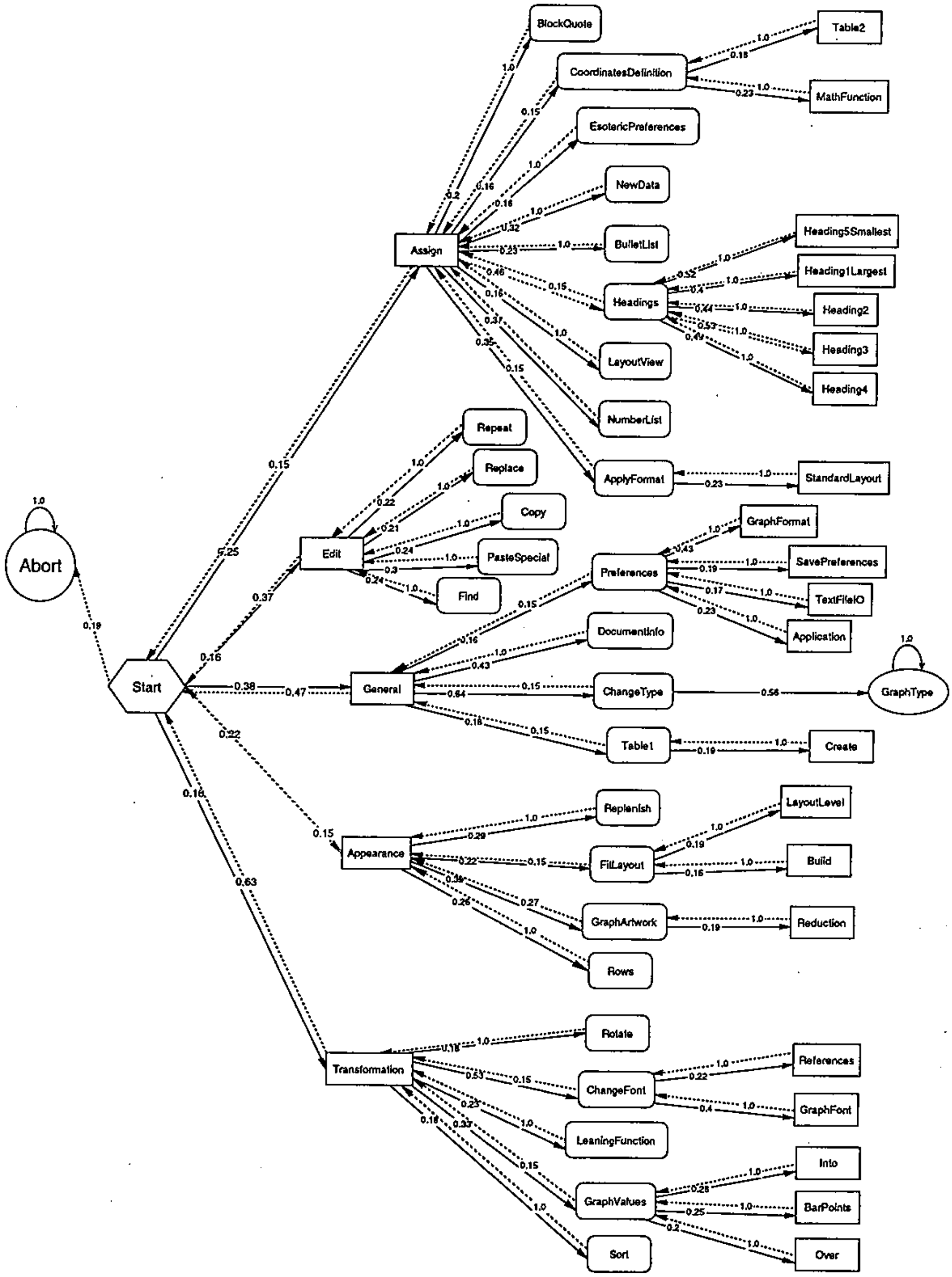

Figure 2: A hierarchical menu with similarity indices (a memoryless model).

2.2. Quantification of similarities

In CoLiDeS, when a user selects a text link from a set of menu items encountered for the first time, the similarity at the semantic level is combined with that at the textual level. If there are no text links with a label identical to the representation of the goal among the menu items currently being processed, the selection is made on the basis of semantic similarity.

Latent Semantic Analysis (LSA) is used to quantify the semantic similarity in CoLiDeS [6]. As a means of statistically evaluating the relationship between a word and the context (document) containing that word, LSA expresses this relationship in a vector form in a semantic space of some 300 dimensions. At the website *http://lsa.colorado.edu*, one may obtain interactively the similarities between pairs of words and compound words in a semantic space that is constructed from the vocabularies of American university students. The values representing the semantic similarity in this paper have been obtained from this website.

The semantics of compound words are expressed by combining vectors. The similarity between a pair of compound words is defined as the cosine of the two vectors in the semantic space. For example, the similarity of *human computer interaction* and *software engineering* is 0.64. This reflects the fact that they often appear together in context, i.e., they are similar words. On the other hand, the similarity of *parenting* and *human computer interaction*, a pair unlikely to appear in the same context, is 0. Semantic similarity may thus be quantified objectively by means of LSA.

3. ITEM SELECTION PROCESS IN THE MARKOV MODEL

In this section we assume that a hierarchical menu is comprised of three levels — a menu bar, pull-down menus and submenus. We present a method of calculating the success probability and the average number of item selections to the success when selecting a target item in the hierarchy. We first describe the menu hierarchy considered, and then show our method for modeling it with a Markov chain.

3.1. Expression of a hierarchical menu and assignment of similarity

Figure 2 shows a hierarchical menu used in this paper, adapted from [10] which investigated how the label quality in the menu affects learning and performing by exploration. Users select a series of menu items in this hierarchy as follows.

1) The user selects a menu bar item to display a pull-down menu, or completes the task.
2) The user selects a menu item in the pull-down menu to display a dialog box or a submenu, or returns to the layer above without selecting any menu item.
3) If a submenu is displayed, the user selects a menu item in the submenu to display a dialog box, or returns to the layer above without selecting any menu item.
4) If the user displays the correct dialog box, the task is complete. Otherwise, the user returns to the layer above.

The menu bar is made up of seven items: File, Edit, General, Assign, Transformation, Appearance, and Tools. When an item in the menu bar is selected, a pull-down menu appears. Each pull-down menu may contain about ten items. When an item in the pull-down menu is selected, either a submenu appears, or a command is executed, e.g., displaying a dialog box. When an item in the submenu is selected, a dialog box appears.

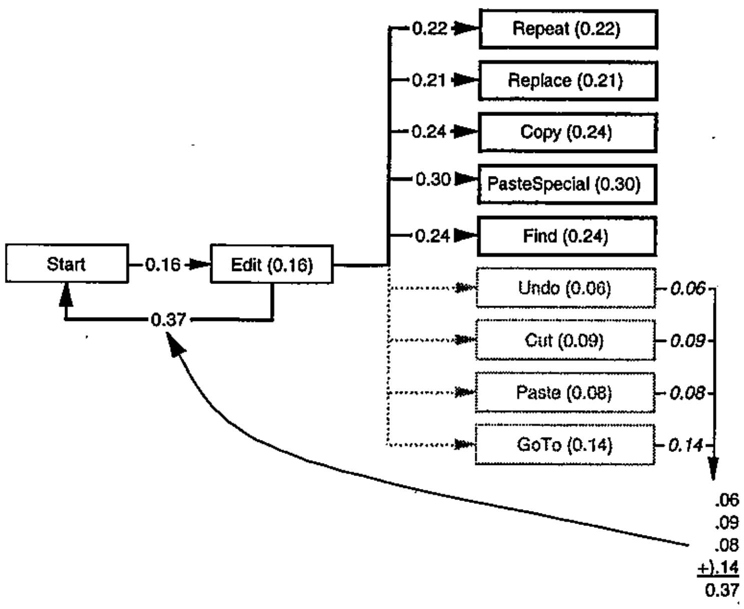

Figure 3: Assignment of weight for "backward" transition.

Figure 2 shows a portion of the hierarchical menu. The values labeling the solid-line arrows in the figure denote the semantic similarities between the task and each menu item when the user is attempting to perform the task "change the graph type to column." These values have been derived from the above-mentioned semantic space of the American students' vocabularies. The shortest path to the target for this task is "Start → General → Change Type → Graph Type." If Graph Type is selected, a dialog box consistent with the task is displayed, and then the user realizes a success in his task.

In reality, however, users may return to the upper layer if an incorrect dialog box appears or if there is not any appropriate menu item to select. The values labeling the dashed-line arrows in Figure 2 denote the weights assigned to the links for returning to upper levels. Since an incorrect terminal node always results in returning to the level above, it has a link weight of 1.0. There are no links from the correct terminal node and from the Abort node at which the task is terminated. The returning link weights of other nodes are determined according to the procedure illustrated in Figure 3. Since it is assumed that users apply a label-following strategy to make selections, selecting items with low similarity is not allowed. Thus, the possibility of selecting a node with the similarity to the task goal below a threshold (say, 0.15) is discarded, and that similarity value is incorporated into the returning link weight. In Figure 3, there are nine items in the pull-down menu of the Edit item in the menu, four of them having similarities below the threshold. With these similarities added up, the link weight returning from the Edit pull-down menu amounts to 0.37. If such conditions occur in a path from the Start menu, the similarities are added to the Abort link weight. When the sum of all sub-threshold similarities remains below the threshold or no selection item exists with similarity below the threshold, the returning link weight is reckoned as the threshold value.

Thus the links are weighted based on their similarities, and chosen by the user accordingly. All tasks are started with Start and ended at either the target or Abort in a finite amount of time with probability one.

3.2. Formulation by a Markov chain

The process of menu selection for a given hierarchical menu system can be modeled as a Markov chain with a discrete state space and a discrete time parameter in which each menu of selectable items corresponds to a state and the sequence of mouse clicks corresponds to the time parameter. Assuming that there are $r\,(<\infty)$ kinds of menus, let the state space be denoted by $S=\{1,2,\ldots,r\}$. Let X_n be the menu popped up at the nth mouse click. For the Markov chain $\{X_n; n=0,1,2,\cdots\}$, it is assumed that the time-homogeneous state transition probabilities

$$P_{ij} \equiv P\{X_{n+1}=j \mid X_n=i\} \quad i,j\in S$$

are given. Let T_{ij} be the first passage time from a transient state i to a recurrent state j of the Markov chain. Then the probability that state j is reached from state i in a finite number of time steps is denoted by $f_{ij}\equiv P\{T_{ij}<\infty\}$. Also the expected number of steps needed to go from state i to state j in a finite number of steps is denoted by $\mu_{ij}=E[T_{ij};T_{ij}<\infty]$.

According to the theory of finite Markov chains, they satisfy the following sets of linear simultaneous equations [8]:

$$f_{ij}=\sum_{k\in C(j)}P_{ik}+\sum_{k\in T}P_{ik}f_{kj} \quad i\in T \tag{1}$$

$$\mu_{ij}=f_{ij}+\sum_{k\neq j}P_{ik}\mu_{kj} \quad i\in T \tag{2}$$

for each recurrent state j. Here $C(j)$ denotes the recurrent communicating class to which state j belongs, and T denotes the set of all transient states of the Markov chain. Therefore, given $\{P_{ij}\}$, we can obtain $\{f_{ij}; i\in T\}$ and $\{\mu_{ij}; i\in T\}$ for each recurrent state j by solving these equations.

3.3. Construction of the transition probability matrix

We can evaluate the task performance analytically or numerically for the Markov chain with state transition probabilities. In this subsection, we first describe how to construct the transition probability matrix for the state transitions shown in Figure 2. We then show a procedure to calculate the task performance.

For the Markov chain with a discrete state space, let the (i, j)th element of the transition probability matrix be the probability of going from state i to state j in one step. The set of similarity indices, calculated by the method given in Section 3.1, is converted to the state transition probabilities by normalizing them proportionally so that each row sums up to unity. We number the 56 states in Figure 2, beginning with state 1 for Start to the terminal nodes. For convenience, let state 55 be our target (Graph Type), and let state 56 be Abort. We then have the following transition probability matrix:

$$(P_{ij})=\left(\begin{array}{cccc|cc} P_{11} & P_{12} & \cdots & P_{1,54} & P_{1,55} & P_{1,56} \\ P_{21} & P_{22} & \cdots & P_{2,54} & P_{2,55} & P_{2,56} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ P_{54,1} & P_{54,2} & \cdots & P_{54,54} & P_{54,55} & P_{54,56} \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right).$$

Note that states 1–54 are transient and that only states 55 and 56 are recurrent (absorbing) in this Markov chain.

Let us define the 54×54 matrix $Q=(Q_{ij})$ by

$$Q_{ij}=\delta_{ij}-P_{ij} \quad 1\le i, j\le 54$$

where $\delta_{ij}=1$, and $\delta_{ij}=0$ if $i\neq j$.

It follows from (1) and (2) that

$$\begin{pmatrix} P_{1j} \\ P_{2j} \\ \vdots \\ P_{54,j} \end{pmatrix}=Q\begin{pmatrix} f_{1j} \\ f_{2j} \\ \vdots \\ f_{54,j} \end{pmatrix}, \quad \begin{pmatrix} f_{1j} \\ f_{2j} \\ \vdots \\ f_{54,j} \end{pmatrix}=Q\begin{pmatrix} \mu_{1j} \\ \mu_{2j} \\ \vdots \\ \mu_{54,j} \end{pmatrix}.$$

Solving these sets of equations for $j=55$ and 56, we can obtain $f_{1,55}$, $\mu_{1,55}$, $f_{1,56}$, and $\mu_{1,56}$.

Now $f_{1,55}$ and $\mu_{1,55}$ are interpreted as

$$f_{1,55}=\frac{\text{number of accesses that reach the target}}{\text{number of all accesses}}, \quad \mu_{1,55}=\frac{\text{sum of the numbers of clicks reaching the target}}{\text{number of all accesses}}.$$

Thus, the average number of clicks for those accesses that reach the target is given by

$$\frac{\mu_{1,55}}{f_{1,55}}=\frac{\text{sum of the numbers of clicks reaching the target}}{\text{number of accesses that reach the target}}.$$

Similarly, the average number of clicks for those accesses that are aborted is given by $\mu_{1,56}/f_{1,56}$.

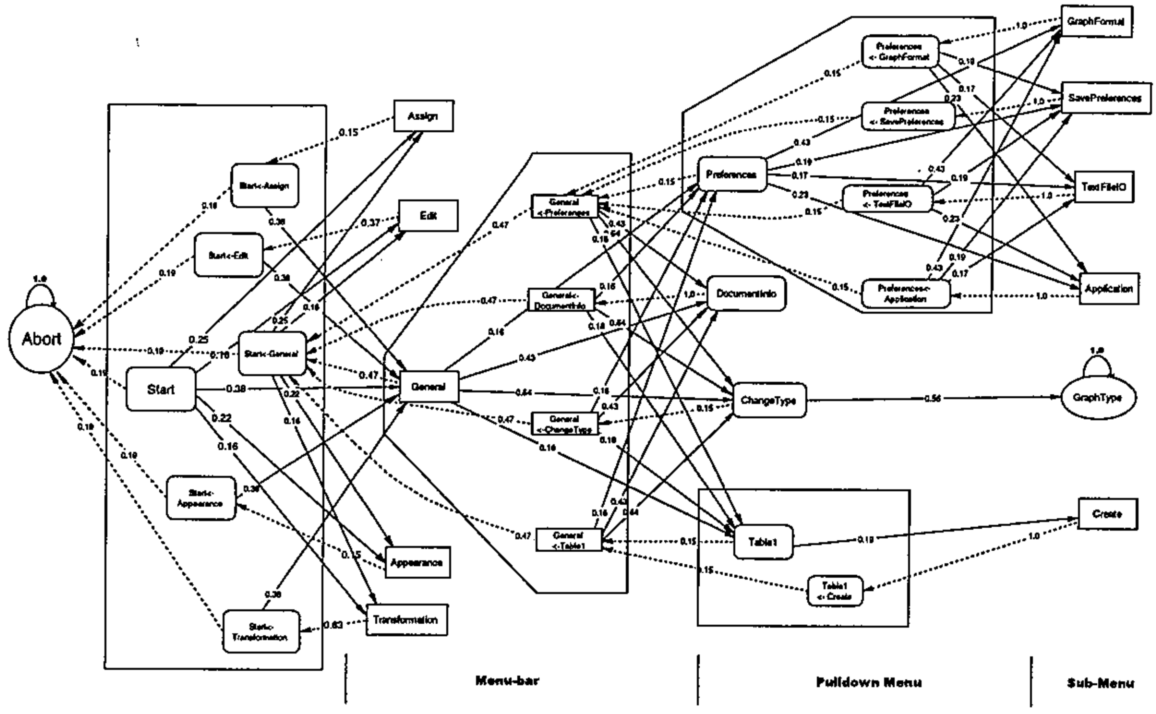

Figure 4: A Markov model of patient users (M2).

4. EVALUATION OF HIERARCHICAL MENUS

In addition to the Markov model given above, we introduce four new models of the user's item selection process and evaluate them by assigning concrete values. Based on the numerical results for the five user models, we discuss the effects of the user's history of mouse clicking on the performance of the search process. Finally we propose a general method for evaluating and designing the hierarchical menu structure by combining our user models.

4.1. Five user models

4.1.1. Memoryless users (M1)

Figure 2 may be thought of as a state transition diagram for the model of users without memory (M1). Although the probability of reaching a target can be calculated exactly, the evaluation of the number of clicks may not be realistic in this model.

4.1.2. Patient users (M2)

Markov chains rely only on the current state to calculate the probability of future states under all historical conditions, i.e., independent of the past history. In the actual item selection process, however, it is hard to imagine revisiting a state where the target was not found on a previous visit. In other words, the item selection process is not a little influenced by the past history. Figure 4 illustrates a model of patient users (M2) who would never select the item from which they have just returned. For example, consider a user who selects General from the menu bar and Preferences from the pull-down menu (Start → General → Preferences). The user may then return without selecting the Preferences submenu. At this point the destination of return is the node labeled General←Preferences in the figure. At this node there is no link to Preferences but links to other items in the General pull-down menu. In this pull-down menu, if the user again returns to the layer above without selecting any item, he returns to the node labeled Start←General, where the link to General is cut off.

4.1.3. Impatient users (M3)

Note that in M2 users are allowed to traverse a menu hierarchy indefinitely, making a tremendous number of transitions before either arriving at the target or Abort. This cannot be reckoned as a realistic model of searching menus. Figure

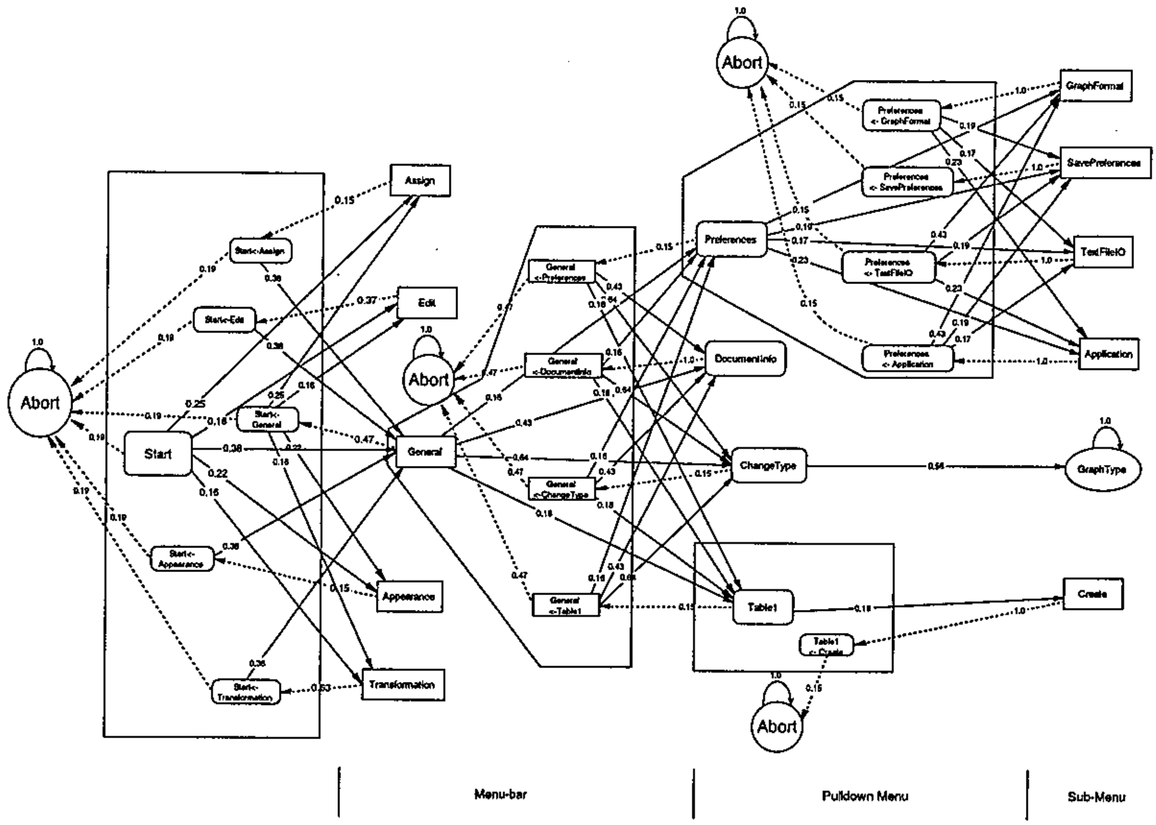

Figure 5: A Markov model of impatient users (M3).

5, a slight revision of Figure 4, shows a model of impatient users (M3) in the sense that it implements the depth-first search and places additional restrictions on returning for completing the search quickly. This model differs from M2 in the sense that a user cannot return by two or more levels upward in the hierarchy, aborting the task in such a case. For example, if a user selects to return in the General←Preferences state, he moves to Abort rather than to Start←General.

4.1.4. Cautious users (M4)

User models M1, M2 and M3 have the advantage of being able to generate the transition probability matrices for the Markov chain, but are not realistic models of actual user's item selection processes. A model that does not select a link previously selected would be more appropriate in reality. Thus our model of cautious users (M4) assumes that, in the state transition in Figure 2, the user never revisits a node visited before. For example, if a user moves from Start to Appearance and returns to Start without selecting anything in the Appearance pull-down menu (Start → Appearance → Start), the link from Start to Appearance is severed entirely. Thus it is not possible to reach any layer below Appearance afterwards. It implies that the user is unable to reach a certain section of the hierarchy even though that section has never been visited. Therefore this model may be considered as a model of users who would return to a level above only with great caution. It is the case where there is clear indication of previous visits such as we meet on web pages.

4.1.5. Users with good memory (M5)

Finally, our model of users with good memory (M5) is a modification of M4 in such a way that describes users who neglect to revisit only terminal nodes. As can be seen in Figure 2, not all the terminal nodes correspond to the depth of a certain layer in the menu hierarchy. A user with good memory would remember perfectly which were not correct terminal nodes. This model may apply to the users of office applications such as Excel.

Table 1: Success and failure probabilities and average number of clicks.

		Success Cases				Failure Cases		
Model	Description	success prob.	95% confidence interval	average number of clicks	95% confidence interval	failure prob.	average number of clicks	95% confidence interval
M1	memoryless users	.509	—	131.2	—	.491	124.6	—
M2	patient users	.451	—	85.9	—	.549	81.5	—
M3	impatient users	.124	—	4.1	—	.876	10.8	—
M4	cautious users	.304	(0.294,0.313)	16.3	(15.8,16.9)	.696	24.3	(23.9,24.8)
M5	users with good memory	.520	(0.510,0.530)	32.3	(31.5,33.1)	.480	27.3	(26.5,28.1)

4.2. Results and discussion

4.2.1. Numerical results for the user models

Table 1 gives the probabilities $f_{1,55}$ and $f_{1,56}$ of reaching the target or aborting the task, respectively, along with the corresponding average first passage times $\mu_{1,55}$ and $\mu_{1,56}$, i.e., the average numbers of clicks, for five user models M1 through M5. For user models M1, M2 and M3, they have been calculated as described in 3.3 from the transition probability matrices of the Markov chains. For user models M4 and M5, we have conducted simulation with 10,000 item selections, and counted the success and failure frequencies as well as the number of clicks. In the latter case, the results are shown with 95% confidence intervals.

4.2.2. Discussion

Based on the results in Table 1, we observe that taking into account the history of selecting menus has significant impact on the performance of the search process. First, comparing the results for M1 and M2, we find that M2 gives much fewer average clicks for both success and failure cases. This is because the number of selectable items after selecting a return in M2 is fewer by one than that in M1. For the same reason, the success probability is slightly lower in M2. For instance, following the transitions Start $\rightarrow$ General $\rightarrow$ Start, the probability of abortion is 0.19/(0.19+0.25+0.16+0.38+0.22+0.16) = 0.088 in M1 (Figure 2), while the probability of abortion after the state Start$\leftarrow$General is 0.19/(0.19+0.25+0.16+0.22+0.16) = 0.107 in M2 (Figure 4). On the other hand, the probabilities of transition towards the target in M1 and M2 are comparable. Thus the probability of reaching the target is lower in M2.

Looking at the successful case in M3, we see that the average number of clicks is 4.1, which is close to the length of the shortest path (Start $\rightarrow$ General $\rightarrow$ Change Type $\rightarrow$ Graph Type). On the other hand, we find that the average number of clicks in the failure case is 10.8; in this case a user moves to Abort after experiencing several trials and errors. The results for M3 come from the fact that a user cannot return across layers in a hierarchy for the depth-first search. The results for M4 are similar to those for M3, but exhibit higher success probabilities and higher average number of clicks, because a user in M4 may return across layers. Since the strongest priority on the depth-first search is put in M3, the results obtained in M3 may provide the lower bound performance for M4.

Since revisiting to a terminal node is prohibited in M5, it naturally yields higher success probabilities as well as higher average number of clicks than M4. Although the state transition diagram of M5 is very close to that of M1 (Figure 2), it shows somewhat higher success probabilities and much lower average number of clicks. Therefore, the success probability in M1 may be close to that in M5.

4.3. Proposal of a method for evaluating hierarchical menus

As demonstrated above, it is possible to find the success probabilities and the average numbers of clicks for the five user models, given a hierarchical menu structure with a user target. User models M1, M2 and M3 are subject to poor accuracy for models of the actual user's item selection process, but they enable us to perform high-speed evaluation as they yield the evaluation results in a short time by means of numerical calculation. On the other hand, user models M4 and M5 reflect the behavior of actual users. While it takes much long time to obtain the evaluation results by means of simulation, they can provide accurate detailed evaluation.

We now propose a method for the evaluation of hierarchical menus that takes advantage of these characteristics. Our method provides an efficient way for evaluating and improving the organization of hierarchical menus, and works as follows:

1) Define a number of hypothetical targets (they may be represented by terminal nodes).
2) Design several candidates of the hierarchical menu structure.
3) Use the high-speed evaluation models (M1~M3) to find the success probability and the average number of clicks for all the targets, and evaluate the candidates of hierarchical menu structure. For a small set of hierarchical structure candidates that are rated high by the high-speed evaluation, perform the detailed evaluation by either M4 or M5, depending on whether the menu structure is of web page-type (M4) or of office application-type (M5).

5. CONCLUSION

We have discussed methods of evaluating a hierarchical menu, a structure that is a major user interface indispensable to gain access to the rich information resource made available by today's information technology, with respect to its task execution performance. Our models of the process by which users select target items in the hierarchical menu are based on a cognitive model of user behavior. Five models have been presented corresponding to the user's different item selection behaviors. Three of them are capable of generating the transition probability matrices for Markov chain models, and can thus find the success probabilities and the average numbers of clicks at high speed. The other two models capture more realistically the user's behavior on web pages and in office applications; as such they can provide highly accurate evaluation at the expense of long computational time for simulation. We have demonstrated the possibility of finding with these models the actual success probabilities and the average numbers of clicks to the success for a sample hierarchical menu structure. On the basis of this result, we have proposed a method of evaluating and improving hierarchical menu structures. Work is currently underway on the implementation of this method. Further experiments will be required to verify its effectiveness using numerous samples from real applications.

REFERENCES

1. M. Franzke, "Turning research into practice: Characteristics of display-based interaction," In *Proceedings of the CHI'95 Conference on Human Factors in Computing Systems*, ACM Press, pp. 421–428, 1995.
2. W. Kintsch, *Comprehension; A Paradigm for Cognition*, Cambridge University Press, 1998.
3. M. Kitajima and P. G. Polson, "A comprehension-based model of correct performance and errors in skilled display-based, human-computer interaction," *International Journal of Human-Computer Studies*, **43**, pp. 65–99, 1995.
4. M. Kitajima and P. G. Polson, "A comprehension-based model of exploration," *Human-Computer Interaction*, **12**, pp. 345–389, 1997.
5. M. Kitajima, M. H. Blackmon, and P. G. Polson, "A comprehension-based model of web navigation and its application to web usability analysis," In *People and Computers XIV – Usability or Else!* (S. McDonald, Y. Waern, and G. Cockton, Eds.), Springer, pp. 357–373, 2000.
6. T. K. Landauer and S. T. Dumais, "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge," *Psychological Review*, **104**, pp. 211–240, 1997.
7. T. J. Mayes, S. W. Draper, A. M. McGregor, and K. Oatley, "Information flow in a user interface: The effect of experience and context on the recall of MacWrite screens," In *Proceedings of the HCI'88 Conference on People and Computers IV*, Cambridge University Press, pp. 275–289, 1988.
8. E. Parzen, *Stochastic Processes*, Holden-Day, Inc., 1962.
9. B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction,* Second edition, Addison-Wesley, 1992.
10. R. Soto, "Learning and performing by exploration: Label quality measured by latent semantic analysis," In *Proceedings of the CHI'99 Conference on Human Factors in Computing Systems*, ACM Press, pp. 418–425, 1999.
11. J. M. Spool, T. Scanlon, W. Schroeder, C. Snyder, and T. DeAngelo, *Web Site Usability: A Designer's Guide*, Morgan Kaufmann, 1999.
12. C. Wharton, J. Rieman, C. Lewis, and P. Polson, "The cognitive walkthrough method: A practitioner's guide," In *Usability Inspection Methods* (J. Nielsen & R. L. Mack, Eds.), John Wiley & Sons, Inc., pp.105–140, 1994.