Modeling an Opportunistic Strategy for Information Navigation

Craig S. Miller (cmiller@cs.depaul.edu)
School of Computer Science, DePaul University, 243 S. Wabash Ave.
Chicago, IL 60604 USA

Roger W. Remington (rremington@mail.arc.nasa.gov)
NASA Ames Research Center, MS 262-4
Moffett Field, CA 94036 USA

Abstract

A computational model of a user navigating Web pages was used to identify factors that affect Web site usability. The model approximates a typical user searching for specified target information in architectures of varying menu depth. Search strategies, link ambiguity, and memory capacity were varied and model predictions compared to human user data. A good fit to observed data was obtained for a model that assumed users 1) used little memory capacity; 2) selected a link whenever its perceived likelihood of success exceeded a threshold; and, 3) opportunistically searched below threshold links on selected pages prior to returning to the parent page.

Introduction

The World Wide Web continues to revolutionize how people obtain information, buy products, and conduct business transactions. Yet many companies and organizations struggle to design Web sites that customers can easily navigate to find products or information. The identification of factors that affect the usability of the World Wide Web has become increasingly important. While many of these factors concern the graphical layout of each page in a Web site, the way in which the pages link to each other, often called the site's information architecture, plays a decisive role in the site's usability, especially for sites allowing access to large databases (Rosenfeld & Morville, 1998) Our effort focuses on understanding how a site's information architecture impacts a user's ability to effectively find content in a linked information structure such as a Web site.

We develop our understanding through the construction and testing of a working computational model. The model simulates a user navigating through a site making choices about whether to select a given link or evaluate an alternate link on the same page. Constructing and testing a working model not only complements empirical studies, but also offers advantages over empirical usability testing. Empirical studies are generally too expensive and time consuming to address the wide range of content, configurations, and user strategies that characterize the Web. In contrast, an implemented model can run thousands of simulated sessions in minutes. Also, empirical studies do not inherently provide explanations for their results and thus make it more difficult to determine how a given result generalizes to other circumstances, whereas a cognitive model can describe the underlying processes that produce behavior. For example, computational models have been used to highlight patterns of interactions with a browser (Peck & John, 1992) and report on the accessibility of the site's content (Lynch, Palmiter & Tilt, 1999).

In this paper, we build upon methods that we presented in an earlier paper (Miller & Remington, 2000a). For the sake of presentation, we describe the methods and our model in its entirety. We introduce a new navigation strategy and show how the model's aggregate behavior tightly fits results from an empirical comparison of different site architectures (Larson & Czerwinksi, 1998). Finally, we experiment with the model's assumptions by exploring alternate designs and parameters in order to help identify critical elements in the model's design.

Modeling Information Navigation

We simulate common patterns of user interaction with a Web site with the goal of providing useful usability comparisons between different site architectures. A model that precisely replicates a user's navigation is not possible, nor is it necessary. Useful information can be obtained from a simple model that captures functionally significant properties of the user and site architecture. Here we show how a simple model can predict and explain benefits of one design over another, such as when it is advantageous to use a two-tiered site instead of a three-tiered site.

In constructing our model, we use the following principles:

- The model should only perform operations that are within the physical and cognitive limitations of a human user. In Web navigation, for example, limits on visual attention dictate that a user can only focus upon (and evaluate) one link at a time. Likewise, limits on short-term memory dictate navigation strategies that minimize memory requirements, an assumption consistent with evidence that people often adopt memory minimization strategies (Ballard, Heyhoe, Pook, & Rao, 1997).

- The model should make simplifying assumptions whenever they are not likely to have much impact on aggregate behavior. For example the model takes a fixed amount of time to evaluate a link even though human users' times are certainly variable. Since the model simulates the average user, this simplification will provide a good fit given a reasonable estimate of fixed time from human performance data (Card, Moran & Newell, 1983).
- The model should employ the most effective strategy for a given environment unless compelling evidence from human usage suggests otherwise. Given the large set of navigation strategies that can operate within reasonable physical and cognitive limitations, we examine a strategy that is most effective within known cognitive constraints. This design constraint is the rationality principle (see Card, Moran & Newell, 1983), which assumes that human cognition is generally rational.

Representing a Web Site

Our model interacts with a simplified, abstract representation of a Web browser and a Web site. Each site has one root node (i.e. the top page) consisting of a list of labeled links, each leading to a separate child page. For a shallow, one-level site, child pages are terminal pages, one of which contains the target information that the user is seeking. For deeper, multi-level sites, a child page consists of a list of links, each leading to child pages at the next level. The bottom level of all our sites consists exclusively of terminal pages, one of which is the target page. Our examples are balanced trees (i.e. pages at the same level have the same number of links) since we generally compare our results to studies that use balanced tree structures (e.g. Miller, 1981 and Larson & Czerwinski, 1998). However, our representation does not prevent us from running simulations on unbalanced trees, or even on structures involving multiple links to the same page and links back to parent pages.

When navigating through a site, a user must perceive link labels and gauge their relevance to the targeted information. Rather than model the complex and interesting process of link evaluation, we fix a number for each link, which represents the user's immediately perceived likelihood that the target will be found by pursuing this link. This simplification allows us to easily investigate a range of numerical relationships between the link label and the target information.

In an ideal situation, after evaluating a link, the user would know with certainty whether to select and pursue that link. Figure 1 represents a site with such links. Each link (underlined number) on each Web page is understood without ambiguity. The user need only follow the links labeled with a 1.0 to find the targeted page with no backtracking. We describe the architecture of this site as having a two-tiered, 4x2 structure.

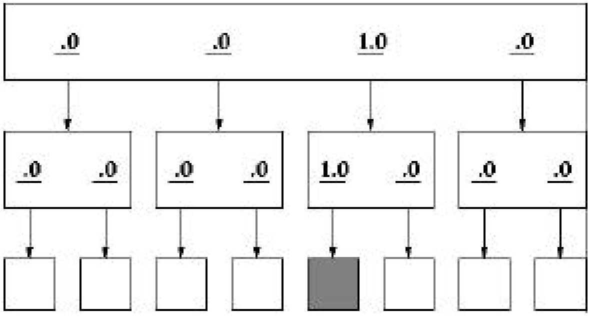

Figure 1 Site with sure path

More often, the user is less certain of which link to select. The links in the site shown in Figure 2 are more ambiguous. For the top page, the most likely link has a perceived likelihood of only .7, thus indicating that the user is less certain that this link will lead to the targeted item. In some cases, a user strategy that merely follows the most likely links would directly lead to the target. However, this figure shows a site where the user would find the target under what he or she perceives as a less plausible sequence of link selections (the target is under a likelihood value of 0.2 instead of the 0.5 value). In this way it is possible to represent sites that differ widely in strength of association between link label and target information.

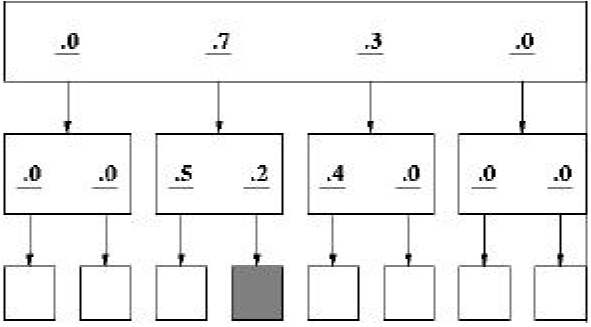

Figure 2 Site with ambiguous labels

Modeling the Browser and User Actions

Byrne et al. (1999) found that selecting a link and pressing the Back button accounted for over 80% of the actions used for going to a new page. Consequently, we focused our modeling on the component actions underlying these behaviors. These include:

- Selecting a link
- Pressing the Back Button

- Attending to and identifying a new page
- Checking a link and evaluating its likelihood

To further simplify our model, attending to and identifying a new page can be folded into the actions of Selecting a Link and Pressing the Back Button since this action only occurs when either of these actions occur. Our revised model has three primitive actions:

- Selecting a link (and attending to and identifying a new page)
- Pressing the Back Button (and attending to and identifying a new page)
- Checking a link and evaluating its likelihood

Because of physical and cognitive limitations, only one of these actions can be performed at any one time. Fixed times are assigned to each action to account for its duration during a simulation. The model also simulates changing the color of a link when it is selected so that the modeled user can "perceive" whether the page under this link was previously visited.

Modeling Navigation Strategies

The model navigates a Web site by serially executing these three primitive actions, meaning that links are sequentially evaluated. Serial evaluation is motivated by evidence that the human user has a single unique focus of attention that must be directed at the link for this decision.

A user could employ any of a large set of possible strategies for link selection that vary in sophistication. Two examples include:

- The threshold strategy: The user immediately selects and pursues any link whose probability of success exceeds threshold.
- The comparison strategy: The user first evaluates a set of links and then selects the most likely of the set.

The threshold strategy is most effective if the first likely link actually leads to the targeted object. The comparison strategy is more effective only if a likely link is followed by an even more likely link that actually leads to the targeted item. Both represent simple yet effective strategies. We chose to begin by examining the threshold strategy on the principle that it requires the fewest computational (cognitive) resources.

The model is neutral as to the actual order in which the links are evaluated. The design and layout of a page principally determine which links a user would evaluate first. Any understanding of how page layout and design affect the user's focus could eventually be incorporated into the model. For our purpose of investigating site structure, the model simply establishes a fixed order in which links are evaluated for each run. To avoid systematic order biases, our simulations randomly place the targeted item at a different terminal page for each run.

With the appearance of a new page, the model's threshold strategy first attends to the page, which, if it is a terminal page, includes checking if it contains the target information. If it does not, the model sequentially scans the links on a page selecting any link whose likelihood is equal to or above a fixed threshold (0.5 in the simulations reported below). When a page appears by selecting a link, the process of checking and scanning the page is repeated.

Once the model detects no unselected links above the threshold value, it returns to the parent page by pressing the Back button and continues scanning links on the parent page starting at the last selected link. It does not scan links it has already evaluated. Determining the last link selected places no demands on memory since the last selected link is easily detected by its color, and many browsers return the user to the location of the last selected link.

Selecting the most probable link will often lead to the targeted item. However, sometimes the targeted item lies behind ostensibly improbable links and, after some initial failures, human users must start selecting links even if the link labels indicate that they will probably not lead to the targeted item. An earlier version of our model (Miller & Remington, 2000a) started selecting improbable links only after completing a full traversal of the site. We will call this the traverse-first strategy. However, a more effective strategy may opportunistically select improbable links at a lower tier immediately after trying the more probable links and before returning to a higher tier in the site. We call this the opportunistic strategy.

Figure 2 illustrates how the opportunistic strategy may be more effective. The model scans across the top page and selects the second link (0.7). On the second level it selects the first link it encounters (0.5). After discovering that this is not the targeted item, it returns to the page on the second level. However, before returning to the top level, it temporarily reduces its threshold to 0.1, selects the second link (0.2) and finds the target on the new page. Had the targeted item been elsewhere in the site, the strategy backs up twice, returning to the top-level and resetting the threshold to the previous value (0.5).

The opportunistic strategy is more efficient than the traverse-first strategy. First, it explores less probable links when the cost of doing so is minimal, that is, when the less probable links are immediately available. Secondly, it implicitly takes into account the positive evaluation of the parent link, which had indicated a high likelihood that the targeted item is under a link on the current page.

The opportunistic strategy is not efficient if employed in cases where all the links on a page have very low likelihood values (defined as less than 0.1). In such cases our model assumes that the user has sufficient memory to know that rescanning the page would be futile, and returns to the parent page. This memory of knowing that the page has nothing worthwhile only lasts as long as the model remains on the current page. Thus, if the model leaves the page and then returns to this same page, the model must assume that the page may be worth rescanning and the opportunistic strategy is employed. This qualification is also consistent with our design principles in that it contributes to an effective strategy while minimizing memory resources.

While generally consistent with our design principle of preferring strategies that place minimal demands on memory, the opportunistic strategy does require state values to be held in memory. If opportunistic search fails to find the targeted item, the model must reset the link selection threshold to the previous value upon returning to the upper level. Resetting a value requires storing the old value before reducing it. Storing and recalling one or two values reasonably fall within the limits of human cognition, but storing and recalling an arbitrary number of values does not. For this reason, our model allows us to fix a limit on the number of previous threshold values it can recall. We initially set this number to one, but later in this paper we will explore the impact of being able to store and recall additional values.

Part of our reason for adopting the opportunistic strategy in place of the traverse-first strategy was our examination of usage logs for a site search task. We conducted a pilot study using a Web site whose structure mirrored a popular department store's organization. Preliminary results suggest that users frequently select ostensibly less probable links before backtracking to other possibilities (see Miller & Remington, 2000b, for more details and an example). We plan future studies that could further identify usages of this strategy.

Simulation Parameters

Our previous work established plausible time constants for link evaluation and link selection (Miller & Remington, 2000a). We compared the model and results from hierarchical menu selection studies and obtained good fits with link evaluation costs set to 250 ms and link selection costs set to 500 ms. The use of time constants is well established (e.g., Card, Moran, & Newell, 1983) and these values are consistent those previous estimates.

To assign likelihood factors to the links, the ideal link values (1, 0) are perturbed with noise according to the formula below:

$$g * n + v$$

where g is a number chosen randomly from a standard normal gaussian distribution (mean=0, stdev=1); n is the noise factor multiplier (equivalent to increasing the variance of the normal distribution); and v is the original likelihood value (0 or 1). Since this formula occasionally produces a number outside the range from zero to one, our algorithm may repeatedly invoke the formula for a link until it generates a number in this range. The noise factor n thus models the level of label ambiguity in the site. Higher levels of ambiguity lead to more frequent backtracking, which may be more prominent in Web search than menu search.

Simulations

To further evaluate the model's design decisions, we compare its performance to the Web navigation results of Larson and Czerwinski (1998). They studied users navigating two-tiered (16x32 and 32x16) and three-tiered (8x8x8) site architectures that were otherwise comparable. Participants took significantly longer to find items in the three-tiered site (58 seconds on average) than the two-tiered sites (36 seconds for the 16x32 site and 46 seconds for the 32x16 site).

Simulations of the Opportunistic Strategy

For our simulations using the opportunistic strategy, sites were constructed as described above, except that the noise was not applied to the bottom level, which leads to the terminal pages. This reflects the fact the participants in Larson & Czerwinski could clearly tell whether the link's label matched the text of the targeted item.

For each site architecture (8x8x8, 16x32, and 32x16) 10,000 simulations were run using the following time costs: 250ms for evaluating a link, 500ms for selecting a link, and 500ms for return to the previous page (pressing the back button). Following Larson and Czerwinski (1998), any run lasting more than 300 seconds was coded as lasting 300 seconds.

Figure 3 shows the calculated mean times of the simulation runs. Not surprisingly, the time needed to find a target increased with link ambiguity. What is more interesting is how link ambiguity interacts with site structure. The 8x8x8 architecture produced slightly faster times at low levels of noise but substantially slower times at noise levels above 0.2. At these higher noise levels the results are consistent with the human users. At noise levels of 0.4 and higher, simulated times were faster with the 16x32 architecture than the 32x16 architecture. This difference was also noted in the study with human users, albeit not reported as statistically significant.

At a noise level of 0.4, the simulation results closely match the human results in absolute terms: 62s (compare to 58s for humans) for 8x8x8, 43s (compare to 46s) for 32x16, and 35s (compare to 36s). It appears that the 0.4 serves a good parameter estimate describing the amount of label ambiguity in the sites used by Larson and Czerwinski.

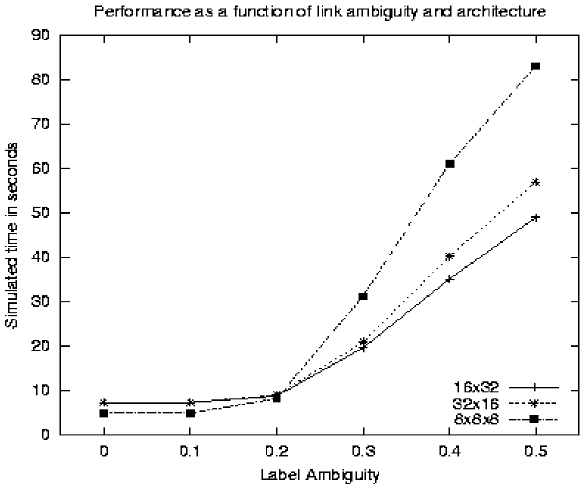

Figure 3 Simulating threshold and opportunistic strategies

Impact of Time Costs

While changing the time costs (250ms for link evaluations and 500ms for link selection and returning to the previous page) will affect absolute simulation times, it is less clear if different time costs will change which architecture produces the fastest times. For example, one may wonder if the 8x8x8 architectures would still produce the slowest times if the link selection cost were double, which may occur for a slower internet connection.

To explore the impact of time costs, we look at the number of link evaluations, link selections and page returns. If independent counts of these actions correlate with the aggregate simulation time, we conclude that varying the time costs have minimal impact on the relative performance of the different architectures. For example, if the 8x8x8 requires more evaluations, more selections and more returns than the other architectures, we know that 8x8x8 will produce slower search times regardless of the time costs.

Looking at the number of evaluations, selections and returns, we see that the 8x8x8 architecture required more of each action (173, 17, and 19 respectively) at the 0.4 noise level than the 16x32 (125, 3, and 5) and the 32x16 (134, 6, and 8). Further experimentation reveals that this relationship holds across all but the lowest noise levels (0.2 and less). We conclude that changing the time costs have no effect on the relative comparisons provided that the noise level is at least 0.3.

Impact of Memory Capacity

Recall that the opportunistic strategy requires the model to store and retrieve threshold values so that the previous threshold can be reinstated upon returning to a parent page. So far, our simulations have assumed that only one threshold value can be restored. Thus, if the model returned to the top level of a three-tier architecture, it would no longer be able to recall the previous threshold and would simply leave the threshold at its current state.

Because this limited memory capacity only hinders performance in a three-tiered site (e.g. 8x8x8), we ran simulations where the memory capacity could hold the additional threshold value so that the previous value could be reinstated when navigating through a three-tiered site. Figure 4 shows the results using the same scale as Figure 3. While we see that the extra memory capacity improves the performance of the 8x8x8 architecture, its navigation is still slower than the two-tiered architectures.

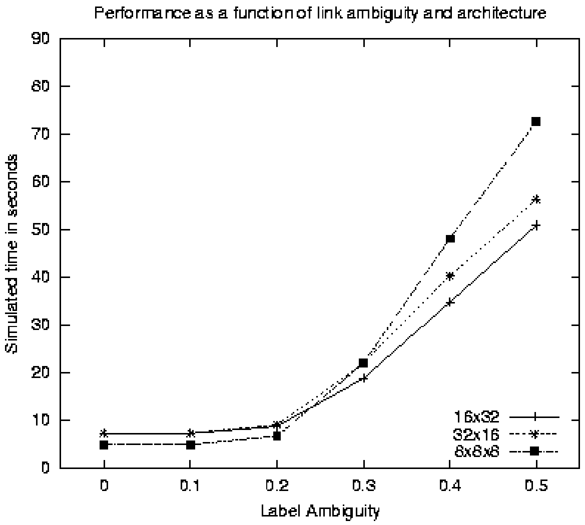

Figure 4 Results using a larger memory capacity

Discussion

We have shown that a simple model of a Web user can provide an excellent account of user behavior and reveal important factors underlying Web usage. The model suggests that link ambiguity interacts with the depth of information architecture to determine site navigation time. As link ambiguity decreases, better performance is found from architectures with deep structures that minimize the number of links searched. As link ambiguity increases, the model shows performance degradations for architectures with deeper structures. The same pattern is characteristic of human users. However, the preference for shallow hierarchies is observed only with sufficient ambiguity in the link labels and with no ambiguity at the bottom level. Thus, the results of Larson and Czerwinski (1998) may not generalize to large numbers of real Web pages with ambiguity at all levels.

As for Web search strategies, combining threshold-based selection with opportunistic search strategies produced simulated times that are very close to observed times for 0.4 noise level. This also corresponds to the behavior of several users searching a department store site in the pilot study mentioned above. We recognize the need for converging methods to independently determine link ambiguity and are exploring theoretical and empirical methods of estimating actual values.

To make time predictions, our model assumes plausible time costs for link evaluation, link selection and returning to the previous page. By noting the actual counts for these operations, our simulations help us understand what happens when the link selection time is significantly longer, as would be the case for a slow internet connection. We found, however, that the time costs have no effect on the relative comparisons provided that the noise factor is at least 0.3. This suggests that a slower internet connection does not impact the relative advantage of shallow architectures when significant link ambiguity is present, at least for the case where no noise is present at the bottom level.

Our simulations also aid our understanding of how human memory impacts effective navigation. Increasing the model's memory capacity improved performance for the deep (8x8x8) structure but left the other two architectures largely unaffected. This suggests that memory is more useful in keeping track of site architecture than in searching within a page. Since searching a page is facilitated by visual cues (e.g., changes in the color of previously selected links) users can avoid reliance on memory. Visual cues are typically not present to remind users of the names and locations of previous links. The interaction of structure with memory capacity indicates further that simple heuristics for representing capacity are insufficient to capture memory phenomena of importance. Instead, it is necessary to examine how the structure of information sites provides aids to memory. Our analysis contrasts with previous advice suggesting that the number of links per page should be limited to 10 (Rosenfeld & Morville, 1998) (see Larson & Czerwinski, 1998, for a discussion based on experimental results).

We have shown that a simple model of a user interacting with a simplified Web site can reveal important factors that affect usability and can support the investigation of the interactions between those factors across a wide range of conditions. What we have presented is not a comprehensive model of Web navigation. No attempt is made to account for how people scan a page, or evaluate link labels or images. By abstracting these processes, and representing only their functionality, the model focuses on understanding how information architecture affects the navigation process. As an approximation of user navigation, the model can account for a range of human behaviors by varying likelihood factors in its site representations. We have shown that the model provides a good approximation of the behavior of the common (modal) user. By varying parameters it should be possible to extend the model to account for alternate strategies.

References

Ballard, D. H., Heyhoe. M. M., Pook, P. K., & Rao, R. P. N. (1997). Diectic codes for the embodiment of cognition. Behavioral and Brain Sciences, 20, 723-767.

Byrne, M. D., John, B. E., Wehrle, N. S., & Crow, D. C. (1999). The tangled web we wove: A taskonomy of WWW use. Proceedings of CHI'99 Human Factors in Computing Systems (pp. 183-190). New York: ACM press.

Card, S. K., Moran, T. P., & Newell, A. (1983). The Psychology of Human-Computer Interaction. Hillsdale, NJ: Lawrence Erlbaum.

Larson, K. & Czerwinski, M. (1998). Web page design: Implications of memory, structure, and scent for information retrieval. Proceedings of CHI'98 Human Factors in Computing Systems (pp. 25-32). New York: ACM press.

Lynch, G., Palmiter, S., & Tilt, C. (1999) The max model: A standard Web site user model. Proceedings of the 5th Annual Human Factors and the Web Conference. Retrieved February 5, 2001 from the World Wide Web: http://zing.ncsl.nist.gov/hfweb/proceedings/lynch.

Miller, C. S. & Remington, R. W. (2000a). A computational model of Web navigation: Exploring interactions between hierarchical depth and link ambiguity. Proceedings of the 6th Conference on Human Factors and the Web. Retrieved February 5, 2001 from the World Wide Web: http://www.tri.sbc.com/hfweb/miller/article.html.

Miller, C. S. & Remington, R. W. (2000b). Exploring Information Navigation Strategies with a Computational Model (Report No. 00-004). Chicago: DePaul University, School of Computer Science, Telecommunications, and Information Systems.

Miller, D.P. (1981). The depth/breadth tradeoff in hierarchical computer menus. Proceedings of the Human Factors Society (pp. 296-300).

Peck. V. A. & John, B. E. (1992). Browser Soar: A computational model of a highly interactive task. Proceedings of CHI'92 Human Factors in Computing Systems (pp. 165-172). New York: ACM press.

Rosenfeld, R. & Morville, P. (1998). Information Architecture for the World Wide Web. Sebastopol, CA: O'Reilly & Associates.