Using Discourse to Aid Hypertext Navigation

Robert Inder and Jon Oberlander *

*Human Communication Research Centre, University of Edinburgh, 2, Buccleuch Place, Edinburgh EH8 9LW, Scotland

Hyper-media is about following links, but this obliges readers to track what they have and have not seen. Navigating within a document can be a significant task, which, if not done well enough, can leave one 'lost in hyperspace'. Displaying the structure of a hyper-document is one way of helping readers move through it. But we have adopted an alternative approach, as advocated by Neilsen [1]. Guided by ideas from Discourse Theory, we are trying to recognise the structure of the reader's own interaction with the system. We have created DS-Info, an enhanced version of the Info hypertext system within the Emacs editor. DS-Info uses the distinction between structural and cross-structural links to identify topics and digressions. We are currently empirically evaluating DS-Info. Preliminary results are encouraging.

1. THE IMPORTANCE OF CONTEXT IN HYPERTEXT

When authors write good conventional text, they have a structure in mind. This structure is reflected in the text at all levels: in the use of anaphors, in the sectioning of the document, in the knowledge assumed of the reader, and in the provision of navigational information—outlines and reviews, so that readers know where they are, and where they are going. However, in hypertext, the reader is expected to diverge from the author's context—to read text in an order of their own choosing. So authors cannot predict—and therefore build on—what a reader has 'just been told'.

One suggested response to this is to try to make the text cards, nodes or pages 'modular' or self-contained, so that each node can be read in isolation [2]. But writing such modular text is hard. When writing the detailed discussion of a topic, trying to address the needs of those who may encounter it without reading the introduction or basic description will produce text that is repetitive, or involve so many links as to be confusing. Readers are more likely to get 'lost in Hyperspace' [3,4].

It is better to use tools and techniques which help make the author's structure clear to the reader. Thus many tools support 'next' and 'previous' links. These structural links cluster hypertext nodes into a mother/daughter hierarchy that mirrors the section/sub-section structures found in normal text. As readers read, they build
up a context: there will be topics they have covered, things they have seen mentioned recently and so forth. But if a reader is using the hypertext as a hypertext, their context may well not match that of the author. A hypertext includes cross structural links, and when a reader reaches a node by means of such a link, they will be reading it in a context, but it will not be the context that the author envisaged. When this clash of contexts occurs, the structural links that embody the author's context will distract, not support, the reader.

We focus here on two specific problems flowing from this clash. First, a hypermedia document on Scottish history describes St. Cuthbert's Casket, and includes a link to information on the saint himself. This node is one of a set describing saints, and is linked to Next and Previous saints. But no saint is meaningfully Next for anybody reading about the casket. This exemplifies the problem of St. Cuthbert's Follower. Secondly, a digression may lead to several inter-related nodes—perhaps on St. Cuthbert's historical setting, life and influence. When readers finish a digression, they want to resume their previous thread. But hypertext systems have no concept of thread or digression, and users must seek their point of digression among all the nodes they have visited. This is the problem of Topic Resumption.

2. DISCOURSE THEORY

To address these problems, we have turned to the theory of discourse structure. There, the aim is to model the construction and evolution of the structures underlying extended discourses. Most approaches agree that discourse is hierarchically structured, and that a limited set of relations link its sub-parts. Differences arise over the number of relations (and levels of representation) proposed by the theories.

Grosz and Sidner [5] propose three levels of representation: intentional, attentional, and linguistic. Within any level, there are just two relations; discourse purposes within the intentional level are related by either (i) immediate dominance (a part-whole, subordination relation); or (ii) satisfaction precedence (a sequencing, coordination relation). By contrast, Hobbs [6] uses just one level, with eight coherence relations, such as Explanation and Parallel. Rhetorical Structure Theory [7] also uses one level, but recruits some 23 relations, including some similar to Hobbs', and some finer-grained. Here, we follow [5], and use two relations.

Whatever the theory, it must explain how new information is integrated into the existing structure of a discourse. 'Discourse popping' is particularly important—it happens whenever the topic under discussion changes, or reverts back to an older topic. Figure 1 illustrates the type of structure underlying a discourse pop.

Applying a two-relation theory of discourse structure to hypertextual discourse suggests a number of more-or-less obvious theoretical identifications. First, Next and Previous nodes can be seen as coordinated with the current node. Secondly, included nodes, and cross structurally linked nodes can be seen as subordinated to the current node. Finally, linking in general is equivalent to discourse attachment. Thus, we can stipulate that the rules for discourse attachment should apply to hypertext navigation, so that contexts change smoothly. For example, hypertext navigation facilities should make discourse popping easy.
Before attachment

After attachment

Figure 1. An example of discourse popping. Before: segments a, c, f and g are available for attachment. After: c, f and g are no longer available.

3. EMACS INFO

These ideas about discourse structure help us reflect 'the actual user's previous personal use of the system' [1]. To explore them, we have created DS-Info, based on the Info hypertext system embedded within the Emacs family of editors. Info handles only plain ASCII text, and expects keyboard commands (see Figure 2). However, it has a number of features that make it suitable for us; in particular, several link types are supported, allowing document structure to be made explicit.

First, consider the problem of Topic Resumption. Notes often initiate digressions. How do we end a digression, and return to the original point? In Info, there is only one way: backtracking. But if a sequence of nodes has been followed after a Note, this is cumbersome and distracting. Instead, DS-Info offers a command, Return to, which closes off a digression and returns to the previous context—it functions as a discourse pop, just like saying Anyway! in a conversation. Following a Note, or jumping to an arbitrary node, initiates a new discourse context, subordinate to the current one. Whenever it displays a node as part of such a nested discourse, DS-Info creates a link to the point of departure from the parent discourse and adds it as a Return to decoration on the target node. Compare Figures 2 and 3.

Secondly, consider the problem of St Cuthbert's Follower. If we follow a digression, the target text is usually presented in a different context from the one for which it was written. This turns whatever discourse information the author has incorporated into an irrelevant distraction. DS-Info handles this by removing any 'discourse context' decorations on nodes which are reached out of context: the reader can still see links to subordinate nodes, but those to coordinate or super-ordinate nodes are suppressed. Thus, a node will bear different decorations, depending on whether it's reached in or out of context. See Figure 4.

Finally, using the first version of DS-Info quickly showed that although the Next
Tabs vs. Spaces

Emacs normally uses both tabs and spaces to indent lines. If you prefer, all indentation can be made from spaces only. To request this, set "indent-tabs-mode" to "nil". This is a per-buffer variable; altering the variable affects only the current buffer, but there is a default value which you can change as well. *Note Locals:*

There are also commands to convert tabs to spaces or vice versa, always preserving the columns of all nonblank text. "M-x tabify" scans the region for sequences of spaces, and converts sequences of at least three spaces to tabs if that can be done without changing indentation. "M-x untabify" changes all tabs in the region to appropriate numbers of spaces.

Figure 2. A node presented by Info. 'Decorations' at the top indicate the destinations of single-key commands to move to previous and superior (up) nodes. Backtracking goes back to the last visited node, and erases all trace of the visit to the current node. *Note Locals:* is an example of an embedded link.

Figure 3. A node presented by DS-Info. This is the same node as shown in Figure 2, but DS-Info has added a Return to decoration.
Figure 4. Different headers for the same node. The upper window shows how \textit{DS-Info} has modified the header specified in the document, shown in the lower window, by removing the structural links and adding a \texttt{Return to} decoration.

and previous links were not missed, the ability to move up the document was. \textit{DS-Info} frequently presented text that built on other nodes that had not themselves been presented, and provided readers with no way to access the prerequisite text. We added a new decoration, leading to information about a node that is reached out-of-context. The destination of this link is the node that was originally up from the current node, but the link signals a different purpose within the discourse, and the reader following it arrives at the superior node out of context.

4. EMPIRICAL STUDY

We predicted that \textit{DS-Info} would be easier to use than \textit{Info}. Davis [8] has carried out a pilot experiment to test this. A new version of \textit{DS-Info} was implemented; more reliable than the original version, it also logs much useful information about how it is used. An experiment was conducted to compare the performance of subjects using this system to their performance using the basic \textit{Info} system.

There were 12 subjects—regular \texttt{Emacs} users unfamiliar with \textit{Info}; the design was within-subjects, balanced across order of questions and system use. After a tutorial on the capabilities of \textit{Info}, subjects were presented with a 94-node hypertext describing, for the benefit of newly arrived students, the entertainment and transport facilities within the (imaginary) university town of Lance. Subjects answered questions about how best a student can be entertained on a limited budget. The questions required movement back and forth between a number of nodes to combine
or compare information on them. The system logged data on the type and timing of the user's keystrokes, and the logs have been analysed to look for effects on the total time taken, error rates, number of keystrokes and so forth.

Analysis revealed no statistically significant differences between Info and DS-Info on these parameters. However, there were several suggestive trends. For instance, 6 out of 24 responses prepared with Info were incorrect, compared with just 1 out of 24 with DS-Info. Secondly, the About command was indeed used when subjects jumped into a new list. Finally, Return to was used in preference to Last when both were available. However, at least one subject stopped using Return to after its behaviour clashed with that of Last. This indicates that supporting multiple models for backtracking leads to new difficulties for users (cf. [9]). All of these issues deserve further investigation.

5. CONCLUSIONS

We believe there is both theoretical and practical value in looking closely at how theories about interpersonal phenomena can be applied to interactions with information technology. We have tried to illustrate this by implementing DS-Info—albeit on the basis of the simplest form of discourse theory—and then using the statistics we have gathered to move ahead. Next, we will shortly run a larger controlled experiment, using Davis's as a pilot. Furthermore, via the Internet, we have gathered a reasonable number of regular Info users who are willing to try out the new system. We intend to gather log data and evaluations from their experience in the field.

REFERENCES