Applying Discourse Theory to aid Hypertext Navigation

Robert Inder, Jon Oberlander
Human Communication Research Centre
University of Edinburgh
2 Buccleuch Place, Edinburgh EH8 9LW, Scotland
fax : +44 31 650 4587
e-mail: {R.Inder,J.Oberlander}@ed.ac.uk

Abstract

We discuss ways of improving navigation facilities in hypertext systems, considering theoretical and implementation issues, from a natural language processing research perspective. The key claim is that certain ideas from the theory of discourse structure can be exploited to improve the context-sensitivity of navigation facilities. In particular, (i) after a node is reached by a jump, links to other nodes can by dynamically suppressed where these are rendered irrelevant by the user's mode of arrival; and (ii) at any point after a jump (or sequence of jumps), the user can easily return to a limited set of structurally accessible nodes. The approach therefore combines certain advantages of navigation based on document structure with those based on individual interaction histories. The ideas have been tested by altering the implementation of *Info*, the hypertextual help system built into the *Emacs* text editor, which runs on Unix, Macintosh and PC systems. Evaluation studies are currently being planned, and we see further potential for exploiting notions from discourse structure to inform the design of navigation aids.

KEYWORDS: Navigation, discourse structure, context-sensitivity, *Emacs Info*

1 Introduction

Amongst the facilities offered by hypertext systems, jumps offer the user the greatest freedom; but they are also the most dangerous, in that they lead to the possibility of a user becoming disoriented, or 'lost in hyperspace' [2]. A reader may not know where they are in a document's structure, or where to go next, or how to get there [4]. For this reason, systems provide both facilities for finding things, such as searches and indices; and also facilities for orientation and navigation through the document structure.

Navigational facilities include textual overviews, such as indices and tables of contents, and graphical overviews, such as browsers and maps [7]. The use of overviews has been contrasted with direct node-to-node navigation, the former adopted by the 'architects', the latter by the 'navigators' [9]. Studies have suggested that choice of navigation method may be influenced by both document structure and reader task, and that designs should make room for both methods [25], although there is also some evidence that this may slow down users' acquisition of mental maps of the hypertext [3].

By and large, overviews tend to enshrine an author's view of the static structure of the hypertext; an alternative approach therefore emphasises the importance of the user's individual interaction with the document, and provides facilities for recording the temporal sequence of an interaction, and presenting it to as a basis for further actions [18]. Facilities provided include backtracking, history lists, timestamps, and footprints. For example, histories record all of a user's steps, and in some cases therefore list the same node several times, but in other cases, as in *HyperCard*'s `Recent` facility [1], collapse multiple visits. By contrast, sequential backtracking allows a user to retrace their steps, in exact reverse order; depending on the implementation, backtracking either adds or deletes entries from the history list.

Here, we follow Neilsen [18] in pursuing an approach to navigation which emphasises 'the actual user's previous personal use of the system' [p302]. We set out to extract additional structure from a user's session, and to exploit that structure to allow navigation to be more sensitive to the individual user's context of interaction. In particular, we turn to the theory of discourse structure to locate the appropriate tools [6]; this body of theoretical and empirical work is directly applicable to the problem of hypertext navigation, and has hitherto remained under-exploited.

In the next section, we discuss informally the role of discourse context in hypertext use, and identify two problems relating to context-handling; we then go on to introduce a small number of theoretical concepts from the theory of discourse structure. This allows us to address the problems. We describe two simple ways of adding further context-sensitivity to a navigation facility. We then indicate how the appropriate facilities have been introduced into an example hypertext system, the *Emacs Info* system [24]. Implementing the changes leads to a number of further choice-points, and we conclude by discussing these, and existing related work.

2 Context in Hypertext

Consider the structural organisation of a conventional book. Although linear in form, it also usually possesses a degree of hierarchical structure, breaking into chapters, sections, subsections, and so on, depending on the genre. In an expository text, the subsections and even paragraphs possess systematic structure; for example, a typical definition-style paragraph will identify an object, and then attribute properties and constituents to it, amplify on one or more of the properties, and cite a particular example of the object [16]. By contrast, in a narrative text, the structure will be somewhat different, with a background setting followed by a sequence of events and states, which can be inferred to have happened in the order in which they are described [20, 15].

In both cases, however, an author selects a structure which is designed to help the reader integrate the information as effectively as possible. Because the author can usually assume that the reader has absorbed information in linear order, the author can exploit the structure of the previous discourse to communicate information in an efficient manner. For example, a pronoun can be used to refer to an entity already introduced to a text; or an explanation can trade on a term defined in a previous section. In turn, readers take advantage of various cues in a discourse context to help integrate incoming information. For example, when they resolve the reference of a pronoun, the discourse context restricts the range of candidates that must be considered; or when a reader encounters a 'cue phrase', such as *anyway*, *however*, or *by the way*, they can establish whether an incoming piece of text shifts back to previous topic, introduces a contrast, or introduces a parenthetical element.

In conventional texts, then, authors and readers rely on the mutual belief that the reader's previous history of interaction with the text—their discourse context—

matches the order of the text. This means that readers can be assumed to know the things they have just been told. In practice, of course, readers do dip into texts without reading from start to finish. But the prevailing convention is that those doing so must accept responsibility for any confusion that results from deviating from the author's chosen presentation. In hypertexts, however, the assumption that readers know what the author just told them does not hold, at least at the global level. Authors relate nodes via links that we may consider *structural*, in that they follow traditional textual structures, such as sectioning. They also use links that we may consider *cross-structural*, which escape traditional textual structure. Systems, of course, may or may not recognise the difference by typing their links. But given that readers are encouraged to pursue cross-structural links to explore a document in whatever order suits them, an author writing material for a particular node cannot assume that the reader shares the author's discourse context, or that any two readers will share a given discourse context.

For this reason, modularity is generally regarded as desirable. In the limit, any node can be read in complete isolation, without relying on the context in which the user encounters it. But modularity demands the mastery of a specific style of writing, requiring substantial cross referencing, or duplication of information, or both. In practice, this will tend to make reading the complete set of nodes repetitive or confusing.[1] At best, modular text will be harder to write than conventional text: ease of authoring calls for tools that can effectively present groups of interdependent pages. (cf. [8]'s 'node clusters'). Such clustering is part of 'context in the small' [18] [p304]. Obviously, when screen size is fixed and text will not scroll, as in *HyperCard*, a series of nodes may have to be linked together, with actions permitted to move a user from one node to the next (or previous) node. However, as we will see in section 4, even when text can scroll, and nodes can thus contain arbitrary amounts of material, authors tend to divide information into clusters of smaller, linked packages. For example, when describing a particular entity, it is plausible to link the node for an identification to separate nodes for attributes, amplifications, and particular examples of the entity. In some cases, the order in which these nodes are accessed is irrelevant; in other cases, it will matter. When authors care about the order, they will once more link the nodes into a sequence, with actions permitting moves to next and previous nodes.

Now, consider how hypertext facilities influence a reader's discourse context. Some systems provide user-controlled in-line expansion of anchors, enriching the discourse context with additional detail for those who request it. Popups function as short digressions, providing more detail, but necessarily returning the user to the same context. Neither disrupts the sequence or structure of the text at the node. Discourse context is therefore held constant throughout these actions. Secondly, given a linked set of nodes, a user may be expected to traverse them in a canonical order. Discourse context is constructed incrementally, and an author can expect a user to have integrated information from early nodes by the time they encounter subsequent ones. Finally, however, the availability of jumps causes radical problems for context construction. If a user jumps from node a into the middle of a sequence of nodes b_1, b_2, b_3, the author's designation of next and previous nodes no longer makes sense: b_1 is only "previous" in the *author*'s context, not the user's; and b_3 is not the next node in a logical sequence from a to b_2 to b_3; in fact, it is only "next" in the author's context. To make this concrete, consider the following case. From a node describing Scottish Retail Organisations, we follow a link to a node describing the St. Cuthbert's Co-operative Society. From here we follow a link to a node describing the saint after which the co-operative is named. This node may

[1] It may be possible to use in-line expansion to tailor node content, possibly even doing so automatically using an approach like [5], but driven by discourse history rather than a user model (cf. [23]).

have been written as part of a set which considers saints in alphabetical order, and in this context, the previous node is St. Baldred, and the next is St. David. However, St. David is not in any useful sense 'next' in the context of the user's study of Scottish retailers. In honour of this case, we term the problem of irrelevant context the problem of St. Cuthbert's Follower.

Jumps, then, cause clashes between the author's and user's discourse contexts. Modularity is a means of minimising authorial context, and thereby reducing the problems caused by such clashes; but in the real world, few texts are completely modular; to the extent that these represent 'bad writing', it may be that we are stuck with it [10]. But even if all texts were modular, jumps would still cause problems for progress through subsequent discourse. For each jump represents a sort of digression from a current topic, and after a sequence of digressions, it may be that a user wishes to return to the original topic. At this point, the availability of navigational aids will influence how they return to the topic; if there is an overview node, they may move to that node, and re-trace their steps to the desired topics; if there is a backtrack facility, they will be able to undo all their steps until they reach their goal. If there is a history list or a graphical browser, they can jump back directly. But using these last facilities imposes a significant cognitive load on the user, and success depends on the user's ability to recognise and visually locate the node to which they wish to return. These methods undoubtedly work, some better than others; but none of them corresponds very closely to how digressions are terminated in ordinary discourse. In discourse, we can terminate digressions and return to previous topics extremely rapidly. Current hypertext, by comparison, suffers from the problem of Topic Resumption.

3 Discourse Theory

Theories of discourse structure have set out to model the construction and evolution of the structures underlying extended discourses. In the field of natural language processing, the focus has been on the generation and understanding of coherent discourse. Hobbs [11, 12], Reichman [22], Grosz and Sidner [6], Mann and Thompson [17], and Polanyi and Scha [21] have made major contributions, proposing theories which attribute hierarchical structure to discourse, and relations of various types between the discourse segments.

The theories vary in the granularity of their analysis of discourse structure. At one extreme, Grosz and Sidner [6], use three levels of description for discourse: intentional, linguistic and attentional structure. At the intentional level, they see two types of connection between discourse purposes: immediate dominance, and satisfaction-precedence. The former is a part-whole relation; the latter is a sequencing relation. The attentional level records which entities are accessible for subsequent reference, using a stack of context spaces. At the linguistic level, the relationships mirror the intentional relations, so that there are just two relations between discourse segments, functioning as subordination and cöordination. At the other extreme, Rhetorical Structure Theory proposes an open-ended set of relations between discourse segments; in Mann and Thompson's work [17], there are twenty-three such relations, including Evidence, Cause, Contrast, and Elaboration. There is no direct mapping from one theory to another, but certain rules of thumb hold: for example, Elaborations are a type of subordination.

Such theories help to characterise which structures and knowledge sources are relevant in either interpreting or generating connected discourse. Part of the task of interpretation consists in retrieving the relational structure underlying a piece of discourse; this structure helps to determine where to integrate the information carried by an individual sentence. The knowledge required to compute the appro-

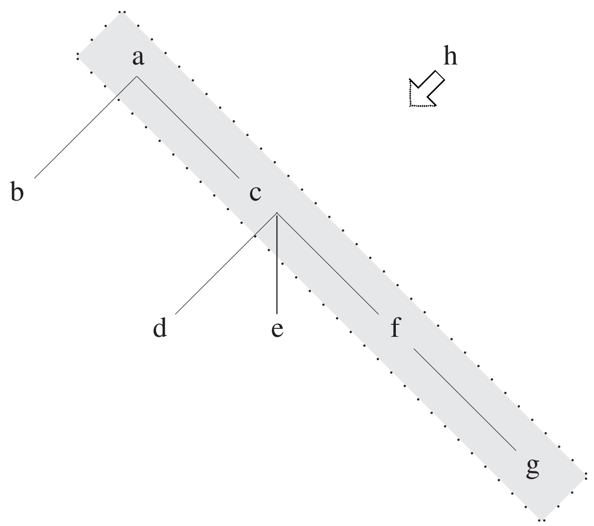

Figure 1: An example of discourse popping. *Before:* segment h must be integrated into the discourse context given by segments a–g. Segments a, c, f and g are the available attachment points, on the 'right frontier' of the tree. g is the default attachment point.

priate relationship is diverse, and involves world knowledge, knowledge about how language is used, and knowledge about the meaning of discourse cue phrases. For our current purposes, the important point is that to integrate new information, we must relate it to the interpretation of the previous discourse.

In the Grosz and Sidner framework, relating an incoming item to the existing structure involves adding it by cöordination or by subordination. The former attaches the new segment as a 'sister', at the same level in the hierarchical organisation; the latter attaches it as a 'daughter' to an existing accessible 'mother' node. The default attachment point is the bottom right-hand node of the tree; however, various discourse cues can cause attachment to occur further up on the 'right frontier'. Subordination in the linguistic structure corresponds to pushes onto the attentional stack; discourse cue phrases such as *anyway* cause items to popped from the stack. At the linguistic level, such phrases thus indicate that a subordinated segment has come to an end, and the new segment should be attached at a higher level in the tree. All lower nodes then become inaccessible for attachment. A case of discourse popping is illustrated in Figures 1 and 2.

Depending on the number of link types supported in a hypertext system, different discourse theories may be more appropriate or less for modelling their use. It is not hard to see how the Grosz and Sidner framework applies to the discourse history generated by a user traversing through a network of nodes in a hypertext system with a small number of link types. "next" and "previous" node links represent cöordinations, while nodes that are included in some sense represent subordinates. If we interpret links for hyperjumps as digressions, then they may be considered a form of elaboration, or (more generally), a further type of subordination. This theoretical identification suggests various modifications to existing hypertext systems.

In the next section, we consider just two: the first addresses St. Cuthbert's Follower; the second addresses Topic Resumption. Before doing so, however, it is worth noting that we have not taken a position on the question of whether hypertext is monologue, or dialogue, or prompted monologue, or any other form of discourse. Most theories of discourse structure apply across these forms; Grosz and Sidner's

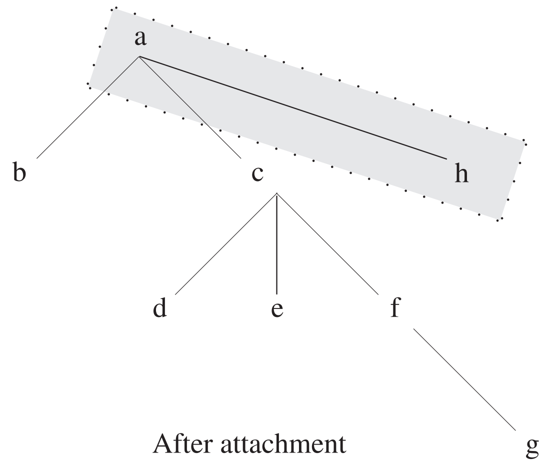

Figure 2: An example of discourse popping. *After*: Having failed to attach lower down the tree, segment h has been attached as a subordinate of a. Segments a and h are now the only available attachment points.

---

theory, for example, was developed for task-oriented spoken dialogues, but applies just as well to written text.

4 Enhancing Info in Emacs

To explore the ideas discussed in the previous section, we have created *DS-Info*, a modified version of the *Info* system which is embedded within the *Emacs* family of editors ([24]). The *Info* system has a number of features that make it a particularly suitable basis for our purposes. Full source code is available, and because *Emacs* provides such a rich environment, *Info* itself is very small—around 1000 lines of *Emacs* lisp. Moreover, it supports a number of link types, which allows document structure to be made explicit. We will illustrate the effects of using *DS-Info* by showing how it presents nodes from the on-line manual for *Emacs*.

Info handles only plain ASCII text, and by default expects keyboard commands. Nevertheless, the document navigation facilities it provides are rich enough for it to unquestionably qualify as a hypertext system. Users see a document as a collection of inter-linked nodes which they can view in any order. Figure 3 shows how *Info* presents the node entitled 'Just Spaces'. This is exactly as it is specified in the document. Although the 'Just Spaces' node happens to be small enough for it all to be displayed, nodes can be any size, and readers use *Emacs*'s normal cursor motion and paging commands to move about the text of the node. This means that there are none of the 'context in the small' problems caused by text being broken into screen-sized pieces [18].

Nodes are 'decorated' with a set of links to the nodes which are '`Next`', '`Previous`' and '`Up`' from the current node, and these links can be followed with just a single key stroke (the 'Just Spaces' node happens not to have a `Next` node). The text of a node can also contain labelled pointers to other nodes, in the form of a menu of nodes, and any number of '`Notes`'—e.g. the link to 'Locals' in Figure 3. These pointers can be followed by moving the cursor to the link or typing enough of the

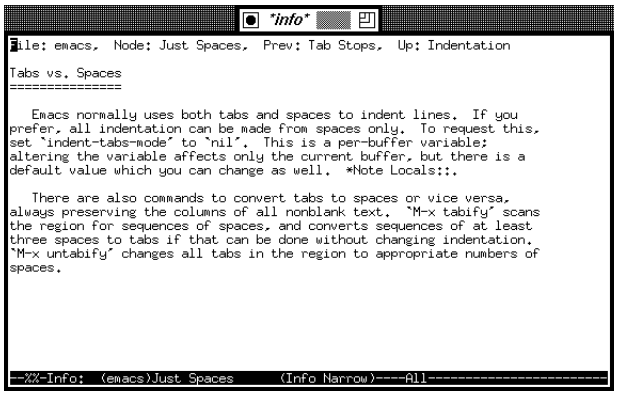

Figure 3: A node presented by *Info*. 'Decorations' at the top indicate the destinations of single-key commands to move to `previous` and superior (`up`) nodes. '*Note Locals::' is an example of a link embedded within the node itself.

link's label to distinguish it from any others in the node. Users can also jump to any node they can name, or to nodes containing any string they specify. Finally, *Info* provides a commands for backtracking, which goes back to the last visited node, and erases all trace of the visit to the current node.

A typical node in an *Info* document is part of a small group which are sequenced through their `Next` and `Previous` links. Since the size of a node is unrestricted, it would be possible to combine such ordered sets of nodes into a single, larger, node. Nevertheless sets of cöordinated nodes are created to reflect the logical structure of the material, as we mentioned in section 2. Such sets of nodes are grouped under—i.e. subordinated to—a single node which is `Up` from all of them, where they appear on its node menu. Most *Info* documents also contain one or more indexes, which are simply normal nodes that contain nothing but an ordered menu of pointers to other nodes.

The typical *Info* document has the structural/navigational features provided in a normal printed document to help readers navigate in terms of the *author's* structure. Selecting from a menu 'pushes' a new discourse context, while going `Up` from a node pops back to the previous one. However, because it is hypertext, readers are expected to depart from the structure the author used to create the document, by following `Note`s, or jumping to nodes specified by either name or content. While these *may* represent a radical change of interest, they will often initiate digressions from the current thread, to review details or concepts presented elsewhere in the document. As such, they are another mechanism for initiating a subordinate discourse segment, and the above discussion of discourse theory immediately highlights the issue of ending it, and returning to our previous thread.

The only way to achieve this within *Info* is to use the `last node` command to 'backtrack' to the node containing the `Note`. This is adequate when the user ends the digression immediately after following the `Note`. But following up a `Note` may lead one to consult several sub-nodes, and possibly even to follow further `Note`s. In such circumstances, ending the digression by backtracking through all these screens is cumbersome and distracting. *Info* therefore suffers from the problem of Topic Resumption. However, given the discussion in the last section, it is clear that we can, and should, offer a command which will close off the digression and return to the previous context.

DS-Info does just this. It interprets following a `Note`, or jumping to an arbitrary node, as initiating a new discourse context, subordinate to the current one. Whenever it displays a node as part of such a nested discourse, it creates a link to the

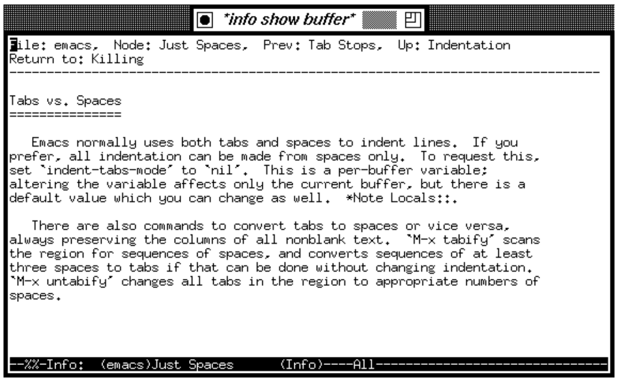

Figure 4: A node presented by *DS-Info*. This is the same node as shown in Figure 3, but *DS-Info* has added a `Return to` decoration.

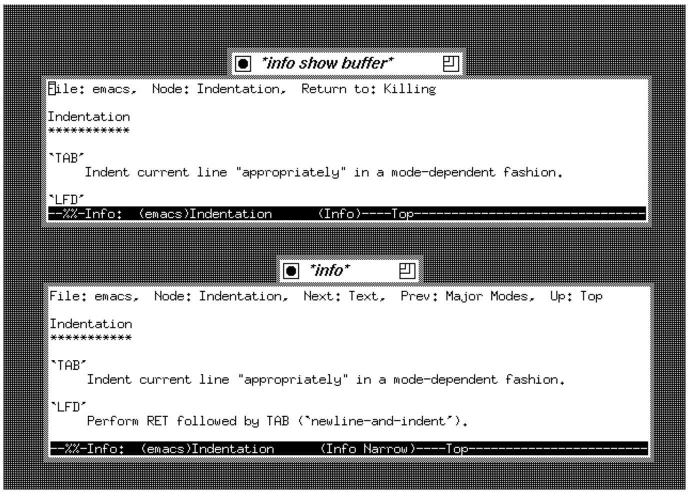

Figure 5: Different headers for the same node. The upper window shows how *DS-Info* has modified the header specified in the document, shown in the lower window, by removing the structural links and adding a `Return to` decoration

point of departure from the parent discourse and adds it as a `Return to` decoration on the target node. This can be seen in Figure 4, which shows how *DS-Info* presents the node shown in Figure 3 *as part of a subordinate discourse segment*. A single keystroke will follow this link and take the user directly back to the page from which they jumped.

The other obvious feature of taking a 'cross-structural' link is that 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—the problem of St. Cuthbert's Follower. *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 cöordinate or super-ordinate nodes are suppressed. This can be seen by comparing the two node headers in Figure 5, where the upper window shows how *DS-Info* removes the structural (`Next`, `Previous` and `Up`) links and adds a `Return to` decoration.

Note that these two issues—being in a subordinate discourse and being out of context—must be handled separately. This is because a node establishes the author's context for its subordinates, so that a user moving to a node by following

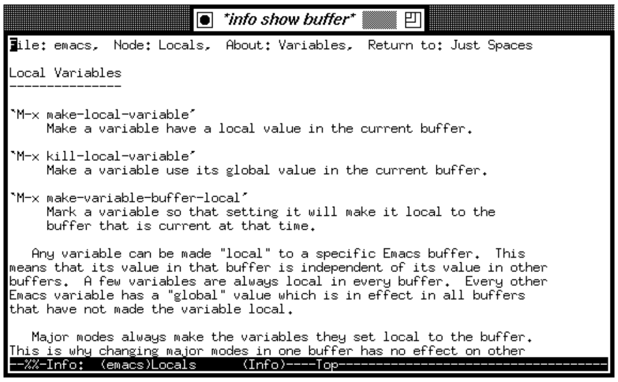

Figure 6: A node showing an **About** decoration. This is the result of using *DS-Info* to follow the link in the text of the node shown in Figure 4: note how the **Return to** decoration now indicates that node.

a '**menu**' pointer will be seeing the target node in the author's context, even though they do so as part of a subordinate discourse segment. As Figure 4 shows, when *DS-Info* presents a node in context, but as part of a subordinate dialogue segment, it adds a **Return to** link, but *leaves* any existing structural links in place.

Using the first version of *DS-Info* quickly showed that although the **Next** and **Previous** links were not missed, the ability to move **Up** the document was. *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. This is, of course, a direct consequence of the fact that hypertext encourages cross-structural jumps. This problem was particularly obvious in the *Emacs* manuals, which were the documents used for evaluation purposes, because they share text and structure with the printed manuals, and thus maximising the independence of nodes was not a high priority. However, as was argued in Section 2, developing an idea in several distinct but inter-dependent units is so natural that avoiding it will greatly increase the difficulty of producing documents. So even though we may try to reduce the extent to which nodes depend on authorial context, we should also ensure that readers can easily access that context if necessary. The context of a node is embodied in its super-ordinate node—the node that is **Up** from it. But notice that although out-of-context readers may want to move to the node that is **Up**, they make the transition for reasons quite different from those of readers who encounter the node in context. Rather than finishing off the study of a group of sub-nodes, they are trying to establish the background to allow one such node to be studied.

In the light of this experience, *DS-Info* was modified to support another new decoration, leading to information **About** a node that is reached out-of-context. This is illustrated by Figure 6, which shows the result of following the link shown in Figure 4. The destination of this link is the node that was originally **Up** from the current node, but it signals a different purpose within the discourse, and the reader following the link arrives at the superior node out of context.[2]

Finally, note that the **Return to** decoration in Figure 6 indicates the node shown in Figure 4, from which the jump was taken. Because multiple digressions are treated as nested structures, and *DS-Info* maintains a stack of return points, taking this link will produce exactly the display shown in Figure 4: having closed off

[2] *DS-Info* does not supply an **About** link for the node shown in Figure 5, because the node is directly below the 'Top' node of the document, which is recognised as a special case of an uninformative context.

the digression about 'Local Variables', the reader will again be offered the chance to `Return to` 'Killing'.

5 Conclusion

To summarise: we have suggested that theories of discourse structure can be recruited to improve the context-sensitivity of navigation facilities. We raised two problems, concerning St. Cuthbert's Follower, and Topic Resumption, and then showed how features of *DS-Info* addressed the problems. Here, we conclude by (i) orienting our work with respect to related systems; (ii) pointing to questions that have been raised in the course of the work; and (iii) indicating the next steps we intend to take.

First, it should be noted that augmenting `Notes` with a `Return to` facility approximates the functionality of an automatic bookmarking system. Every time a jump is made, the jumping-off node is marked as a point to which a rapid return can be made. Indeed, the *DS-Info* `Return` is similar in nature to Neilsen's structure-oriented 'express return' [18], which 'provided a shortcut through the straight backtrack path' [p309]. However, Neilsen noted that users did not use it, and suggested that this was because the express return was only available in one part of the system, and thus appeared inconsistent with the rest of th system's backtracking behaviour. By contrast, the *DS-Info* `Return to` is uniform.

DS-Info's `About` facility resembles a limited version of Instone et al.'s *Incoming Links* [13]. In their system, *HyperHolmes II*, users were presented with 'a list of other nodes in the document that have links pointing to the current node' [p502]; the authors observe that incoming links resemble Neilsen's bi-directional links [19]. Users seemed to find the facility useful, and tended to 'take a look at the nodes which pointed to the current node and then make a single jump' [p503]. Arguably, *DS-Info*'s `About` facility offers a single, 'standard' incoming link to the new node. This is node that establishes the author's context; whether this is the correct node to offer, however, can only be uncovered by empirical testing.

We stated in the introduction that discourse theory has remained relatively under-exploited within hypertext; work on the *IDAS* system, by Mellish and his colleagues, is a notable exception [23]. However, their natural language generation system uses discourse structure to contextualise the linguistic expressions used *within* nodes, and so discourse structure is being exploited at a rather different level. On the other hand, there is work, such as that by Kaplan et al., which alters node accessibility on the basis of individual interaction history [14]. However, their user model-based approach relies on domain-specific information. By contrast, our discourse theory-based approach is a domain-independent attempt to use the structure of the hypertext document to present a well-structured discourse to the user.

Secondly, our work has raised a number of open questions. We have introduced `About` links, which may be regarded as giving background information. In the relatively coarse-grained Grosz and Sidner theory [6], these would appear to act as subordinations, and this raises two questions. Such subordinations seem different in kind from the elaborative subordinations created by following `notes`. Such differences can only be registered by more fine-grained discourse theories, such as Rhetorical Structure Theory [17], which distinguish Background from Elaboration; such theories may therefore be worth pursuing in the hypertext context. Furthermore, if `Abouts` are subordinations, then they ought to push items onto the discourse stack, and become points to which the user can `Return`; we are not yet sure whether this behaviour is desirable.

Another question concerns the treatment of 'endless digressions'. In particular, a user may initially visit a node via a direct cross-structural link, and subsequently make their way back to the jumping-off point by following structural links (such as *Info*'s `Up` and `Next`). This is analogous to a conversation 'drifting back' to an initial topic, from a digression which was explicitly initiated. Our intuition is that, once the topic has been re-activated, it does not make sense to continue to think of explicitly resuming it; at some point the context stack must be 'tidied'. Interestingly, endless digressions are particularly likely to occur if `About` is treated as a subordination. Following the `About` link is intended to let the user see the current node in context, which is achieved precisely by returning to it using structural links.

A third question arises when a user follows a cross-structural link to a node they have already visited in context. By default, *DS-Info* would alter the node's decorations to remove authorial context, so the same node will bear different decorations on the second visit. This seems undesirable, since it denies access to information of which the user is aware. Instead, *DS-Info* currently presents all contextual information for any node that has been reached in context at any stage of these session, no matter how long ago. This is not ideal; there is obvious appeal to using a time-dependent mechanism such as decaying footprints [18].

Finally, we are considering how we may go about evaluating and improving the *DS-Info* system. In the first instance, we must arrive at some provisional answers to the open questions raised above. Only then can we proceed to the kind of empirical interface comparisons advocated by Instone et al. [13]. Our hypothesis is that giving the system an understanding of the ways people structure discourses will help make navigation more more context-sensitive and less distracting. We believe that discourse theory can contribute useful methods for capturing relevant aspects of context. On the basis of our experience so far, we will be developing a new version of *DS-Info*, and we invite other researchers who are interested in evaluating this software to contact us.

6 Acknowledgements

The support of the Economic and Social Research Council for HCRC is gratefully acknowledged. The first author is supported by UK Joint Councils Initiative grant G9018050, (Signal). The second author is supported by an Advanced Fellowship from the Engineering and Physical Sciences Research Council.

References

[1] Apple Computer *HyperCard Script Language Guide: The HyperTalk Language*. Reading, MA: Addison-Wesley.

[2] Conklin, J. Hypertext: An Introduction and Survey. *IEEE Computer* **20**, 17–41, 1987.

[3] Edwards, D. M and Hardman, L. 'Lost in hyperspace': cognitive mapping and navigation in a hypertext environment. Presented at the Hypertext I Conference, Aberdeen, 1988. Available in McAleese, R. (ed.) *Hypertext: Theory into Practice*, pp90–105, 1993.

[4] Elm, W. C. and Woods, D. D. Getting lost: a case study in interface design. *Proceedings of the Human Factors Society*, pp927–931, 1985.

[5] Encarnation, E. O. and Boyle, C. A User Model Based Hypertext Documentation System. *User Modelling and User-Adapted Interaction*, **3**, 1993.

[6] Grosz, B. and Sidner C. L. Attention, Intentions, and the Structure of Discourse. *Computational Linguistics*, **12**, 175–204, 1986.

[7] Halasz, F. G., Moran, T. P. and Trigg, R. H. NoteCards in a nutshell. In *Proceedings of the ACM CHI+GI '87 Conference on Human Factors in Computing Systems and Graphics Interface*, pp45–52, 1987.

[8] Halasz, F. G. Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems *Communications of the ACM*, **31**, 836–852, 1988.

[9] Halasz, F. G. "Seven Issues" Revisited. Talk given at *Hypertext '91*, reported in Lynda Hardman's trip report, 1992.

[10] Hardman, L. Hypertext '91 Trip Report *SIGCHI Bulletin*, **24**, 30–39, 1992.

[11] Hobbs, J. Coherence and Coreference. *Cognitive Science*, **3**, 67–90, 1979.

[12] Hobbs, J. On the Coherence and Structure of Discourse. Report No. CSLI-85-37, Center for the Study of Language and Information, 1985.

[13] Instone, K., Teasley, B. M. and Leventhal, L. M. Empirically-based Re-design of a Hypertext Encyclopedia. In *Proceedings of InterCHI '93*, pp500–506, 1993.

[14] Kaplan, C., Fenwick, J. and Chen, J. Adaptive Hypertext Navigation Based On User Goals and Context. *User Modelling and User-Adapted Interaction*, **3**, 193–220, 1993.

[15] Lascarides, A. and Oberlander, J. Temporal Coherence and Defeasible Knowledge. *Theoretical Linguistics*, **19**, 1–37, 1993.

[16] McKeown, K. R. *Text Generation*. Cambridge: Cambridge University Press, 1985.

[17] Mann, W. C. and Thompson, S. A. Rhetorical Structure Theory: A theory of text organisation. Technical Report ISI/RS-87-190, USC/ISI, June, 1987.

[18] Neilsen, J. The art of navigating through hypertext. *Communications of the ACM*, **33**, 296–310, 1990.

[19] Neilsen, J. *Hypertext and Hypermedia*. New York: Academic Press, 1990.

[20] Partee, B. H. Nominal and temporal anaphora. *Linguistics and Philosophy*, **7**, 243–286, 1984.

[21] Polanyi, L. A Formal Model of the Structure of Discourse. *Journal of Pragmatics*, **12**, 601-638, 1988.

[22] Reichman, R. Conversational Coherency. *Cognitive Science*, **2**, 283–327, 1978.

[23] Reiter, E., Mellish, C. and Levine, J. Automatic Generation of On-Line Documentation in the IDAS Project. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pp64–71, 1992.

[24] Richard M. Stallman *The GNU Emacs Manual*. Fifth Edition, Emacs Version 18. Cambridge, MA: The Free Software Foundation, 1986.

[25] Wright, P. and Lickorish, A. An empirical study of two navigation systems for two hypertexts. Presented at the Hypertext II Conference, York, 1989. Available in McAleese, R. (ed.) *Hypertext: State of the Art*, pp84–93, 1990.