Managing Hypertext Dialogue with Recursive Transition Networks

Carol M. Jackway
Dept of Information Management and Library Studies, Faculty of Business, RMIT, Melbourne Vic 3000
Email: S9502270@disinformation.bf.rmit.edu.au.

Jacob L. Cybulski
Dept of Computer Science and Computer Engineering, La Trobe University, Bundoora Vic 3083
Phone: +613 9479 1270, Fax: +613 9479 3060, Email: jacob@cs.latrobe.edu.au.

Abstract

Hypertext, a medium in which text is presented in a non-linear fashion, can provide a flexible and intuitive way of navigating through an information space. Yet this freedom places an extra cognitive load on hypertext readers who can become lost or disoriented. Modern hypertext systems include facilities such as browsers, maps, bookmarks or footprints which aim to alleviate this problem, yet they are limited in effectiveness. We propose another method which could significantly enhance existing navigational mechanisms. The method aims to assist the user during hypertext navigation by reducing the decision space in the process, thus, focusing the user's attention on a specific topic. Our approach is motivated by the assumption that a reader of a hypertext document has a specific objective to reach, thus, the system should provide information in such an order as to lead the user towards this goal. We call this mode of operation a *user-centered* approach to providing hypertext information.

In our approach, all displayed hypertext nodes are used to present certain topics to the user. The user interacts with the presented information with the aim to test his or her knowledge of the topic. The results of this interaction are assessed by the system, and based on this assessment, the system will guide the user to the next node by activating only the most appropriate links. The user finally selects one of the available links to transfer control to another hypertext destination or to stay at the current node. The entire cycle is constantly monitored by the system, so that the current state of user knowledge can be adequately assessed. We view such an interaction with a hypertext system as a dialog which can be described by a pre-determined grammar of possible hyper-navigation events and subsequently analysed with an appropriate parsing mechanism. We also suggest that the user and domain modeling, as well as, the organisation of hypertext nodes may assist in this process.

Our concepts were demonstrated with the prototype system for computer-aided instruction, HyperCAL

Managing Hypertext Dialogue with Recursive Transition Networks

Carol M. Jackway
Dept of Information Management and Library Studies, Faculty of Business, RMIT, Melbourne Vic 3000
Email: S9502270@disinformation.bf.rmit.edu.au.

Jacob L. Cybulski
Dept of Computer Science and Computer Engineering, La Trobe University, Bundoora Vic 3083
Phone: +613 9479 1270, Fax: +613 9479 3060, Email: jacob@cs.latrobe.edu.au.

Abstract

Hypertext, a medium in which text is presented in a non-linear fashion, can provide a flexible and intuitive way of navigating through an information space. Yet this freedom places an extra cognitive load on hypertext readers who can become lost or disoriented. Modern hypertext systems include facilities such as browsers, maps, bookmarks or footprints which aim to alleviate this problem, yet they are limited in effectiveness. We propose an alternative approach to reducing reader disorientation, which provides hypertext devices to limit the space of navigational decisions based on reader goals, interests and abilities. To this end, we utilise an event parser based on recursive transition networks which is capable of monitoring, predicting and guding a hypertext dialogue with the user. The prototype system, HyperCAL, illustrates the main features of such a parser: it integrates hypertext, computer-aided learning and knowledge-based techniques, and consists of a Prolog back end and a HyperCard interface. The system was applied to the problem of delivering tutorial material on phrase structure diagrams.

Keywords

Hypertext, Computer-Aided Learning, Knowledge-Based Systems, Parsing

1. Issues in Hypertext Navigation

Traditional texts are organised in a linear fashion. Usually, readers start at the beginning and read left to right, top to bottom, until they reach the end. Hypertext is a medium in which text is presented in a non-linear fashion. The text is organised into chunks or nodes which have a central theme or discuss a certain concept. Nodes are the fundamental units of a hypertext and can range in size from a few words encourages integrative thought, and facilitates mental associations in the reader by providing physical associations in the text. Furthermore, hypertext aims to provide a user-friendly interface by incorporating elements which are intuitive and familiar to users, eg. text, graphics, buttons and menus.

Hypertext has been successfully used in applications as diverse as a dictionary, a university information system, a training manual or a software engineering information management system. Balasubramanian (1994) and Conklin (1987) provide excellent surveys of different hypertext concepts, techniques, methods and systems.

Although hypertext is a very powerful tool, it also has certain drawbacks. One of the problems with hypertext is the potential for the user to become 'lost in hyperspace'. This occurs when readers navigating through a hyperdocument by jumping from one node to another, lose track of where they are in the document, or why they came down a particular path. The more complex the network of hyper-links, the greater the chance of this happening. There are several solutions to this problem.

- The first is a map browser, commonly implemented in commercial hypertext systems. A map visualises the interlinking of certain document sections by showing its nodes and links in the form of a graph, with the current node marked or highlighted in some way. This graph enables the readers to locate where they are in the document. Yet, map browsers can become almost incoherent in large, heavily interconnected systems and are not suitable for helping those users who are not visually oriented.
- The second solution is to provide a default path through the document. Because hypertext gives users more control, it also imposes on them more responsibility about their choices. A default path eases this burden on the user, and ensures that all the main information in the document is accessed. The user can stray from this main path if they choose to, but they should be able to come back to it with the press of a button. Default paths are particularly important in educational systems where some linearity must be imposed on the information, for example, when the student must learn some basic topics, before they can understand more complex information or procedures.

There is a range of other navigational mechanisms which aim to alleviate the problem of becoming lost in hyperspace. These include backtracking, query/search facilities, timestamps, footprints, or animation techniques. We propose another method which could significantly enhance existing navigational mechanisms. The method aims to assist the user during hypertext navigation by reducing the decision space in the process, thus, focusing the user's attention on a specific topic. Our approach is motivated by the assumption that a reader of a hypertext document has a specific objective to reach, thus, the system should provide information in such an order to lead the user towards this goal. We call this mode of operation a *user-centered* approach to providing hypertext information. Consider the following examples.

constrain the set of links of interest to the reader, perhaps to a single shelf - a considerable improvement in the number of hyper-navigation possibilities.

- In a tutorial hyper-book, the reader's aim could be to learn a certain range of topics through a series of lectures and interactive exercises. Once the system has established the student's expertise in a certain area of knowledge, it will make no sense to provide the student with hyper-links leading to the exercises in his or her area of expertise, even if such associations may have been planned as part of the course material. Instead the student should be moved into more advanced topics which would build on his or her existing knowledge and strengths.

In the past, focusing reader's attention was achieved mainly by the *information-centered* approach to delivering hypertext information, i.e. by structuring it into a neat hierarchy of concepts or threading it with a mesh of hypertext connections. Such an approach to hypertext management has several obvious advantages. All user decisions are predicted and predefined in advance by hypertext designers. The subsequent selection of navigational options is entirely the user's responsibility. The system does not need to take any initiatives but rather passively responds to the user 'exploring hyperspace' or 'being lost in hyperspace'. As the logic of navigational paths is "wired" into hypertext information and linking, thus, the system does not need to remember previous interactions with the user or to infer the user intentions which drive his or her behaviour. In the information-centered approach, the hypertext model can be kept simple and minimal and the implementation inexpensive. At the same time the system is rigid and unforgiving, as we can see from the following cases.

- Although a travel guide could be structured into a hierarchy of flight destinations, hotels at the destinations, and tours available from these hotels, it will be difficult for the user to arrange his or her holiday by mere browsing of rigidly structured information. Such difficulties stem from the complexity of natural relatioships between browsable objects. Different airlines will fly to similar destinations, hotels chains will have special deals with airlines, tour operators are usually independent of hotels, special packages of flights, hotels and tours may be on offer from travel agencies, etc. A travel information system will, thus, require a powerful query and booking systems to fully support the variety of available information.

- In the case of a shopping guide, it could have been that all of the sales information was split into shops, departments, aisles, shelf units, shelves, and products. At the same time the user would most likely prefer to start with a shopping list, budget, personal preferences as to the make and quality of goods, shopping convenience, etc. Since the likely query structure is much more complex than a simple hierarchical traversal of the information space, the interaction would, thus, have to be interactive and incremental, and the system supporting such a dialogue would have to memorise past user choices and decisions.

not true in a case of a tutoring system. Hence, in all those cases where the complexity of interaction excedes that of simple link following, no amount of structuring of the hypertext information will ever help the user in effective navigation. We need to adopt a *user-centered approach* to information management.

2. User-Centered Model of Hypertext Interaction

We propose to support the *user-centered* hypertext navigation with techniques drawn from the areas of natural language processing (Gazdar & Mellish 1989), computer-aided learning (Burns & Capps, 1988), and knowledge-based systems (Luger & Stubblefield, 1989).

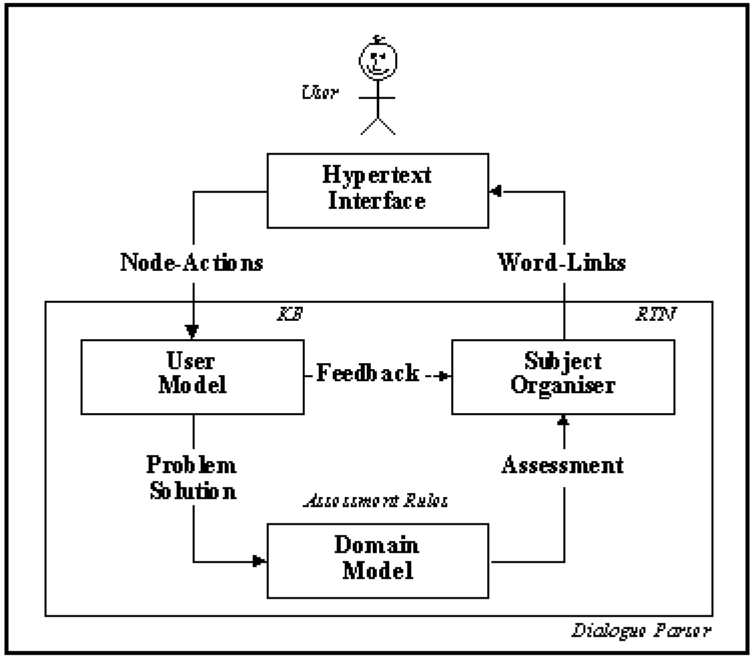

Figure 1 - HyperCAL Architecture

Our method relies on a view of hypertext navigation to be a tutorial dialog between two intelligent agents, the user being a student and the hypertext system being a tutor. The main objective of the student is to acquire maximum information about a certain topic of discourse, while the tutor aims to convey this information in the best possible way, while taking into consideration the student's learning progress. Both of the participants in this dialogue must be capable of understanding and anticipating the other party's objectives, needs and abilities. Only when such an understanding is achieved, may the hypertext system reduce the number of available navigational options to best serve user learning objectives. The proposed model can thus be described in the following terms.

- **Dialogue Constituents.** Interaction with a hypertext system is viewed as a *hyper-dialog* between the user and the machine in which both sides could take conversational initiative. The system displays information nodes, and activates a set of pre-defined *word-links* which may be subsequently selected in the process of navigation. Sequences of word-links constitute *sentence-*

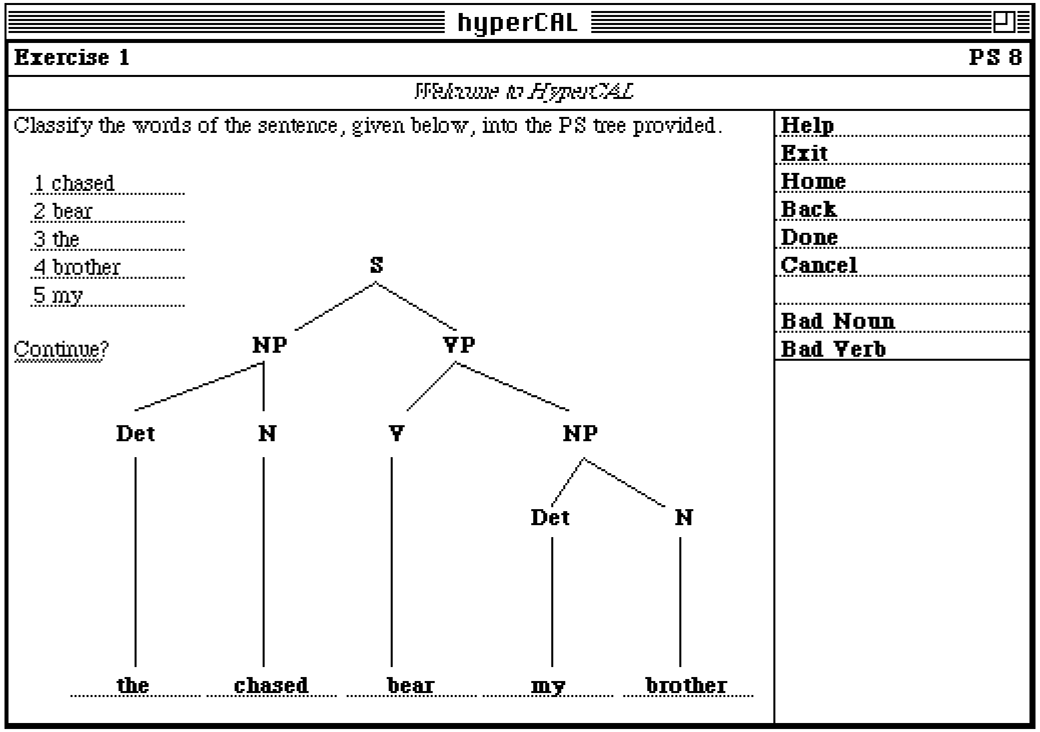

Figure 2 - HyperCAL exercise node with feedback provided

- **Dialogue Management.** While the hypertext system is interacting with the student, it constructs a *user model* of the student's current state of knowledge. This model can be as simple as a list of the presented nodes assessed as fully or partially understood by the user, it could possibly be derived from the trace (e.g. chart) left behind the process of parsing the hyper-dialogue, or it can be as complex as a knowledge-base of user information and experience inferred by an intelligent system. At the same time all knowledge that can be learnt by the user is described in a *domain model*. This model is primarily used in the process of assessing a student's learning performance and in planning educational objectives for that student. All coursework presented to the student is structured by a *subject organiser* into knowledge domains or courses, consisting of subjects, defined in terms of inter-linked topics, which themselves are split into individual issues. The three elements of the user-centered hypertext management system, i.e. the domain and user models plus the subject-organiser, combine into a *dialogue parser* responsible for interpreting user input into the system and producing an appropriate feedback to the user. The communication with the user is conducted via a

HyperCAL Links

```
link(jump,'8','home','1').
link(push,'8','exit','16').
link(push,'8','help','16.1').
link(pop,'8','back',_).
link(jump,'8','redo','8').
link(push,'8',another,'16.2').
link(push,'8',continue,'9').
link(push,'8','well done','9').
link(push,'8','bad det','2.1.5').
link(push,'8','bad noun','2.1.1').
link(push,'8','bad verb','2.1.3').
```

HyperCAL Assessment

```
assess('8', [the, bear, chased, my, brother], 'well done').
assess('8', [Det1, _, _, Det2, _], 'bad det') :-
    not lexicon(8, Det1, det) ; not lexicon(8, Det2, det).
assess('8', [_, N1, _, _, N2], 'bad noun') :-
    not lexicon(8, N1, noun) ; not lexicon(8, N2, noun).
assess('8', [_, _, V, _, _], 'bad verb') :-
    not lexicon(8, V, verb).
```

Figure 3 - Links and assessment rules

information resource system (Nicolson, 1990). Both of the systems rely on the representational strength and inferencing flexibility of a logic language and the elegance and ease of use of a graphical user interface system. However, our main example used to illustrate HyperCAL's functionality was based on a tutorial in phrase structure diagrams, hence, we had a distinct requirement for extremely flexible interaction between the user and the system, in particular a need for responsiveness to the user's skills and abilities. HyperCAL had to employ many attributes not present in USHIR, nor in fact in many other hypertext systems, but instead our system borrowed some concepts and techniques from natural language parsing, user modeling and knowledge representation.

HyperCAL nodes denote either information pages or exercises. All nodes are presented to the user (cf. figure 2) in a window with a menu depicting hypertext links to some standard destinations (e.g. help, exit or home pages), and to other nodes as determined by the Subject Organiser (e.g. previous node, tutorial on nouns or verbs, next exercise, etc.). The connections between the HyperCAL nodes, are modelled in a form of a hyper-dialogue grammar described with recursive-transition networks (RTNs). HyperCAL RTNs have links of four types (jump, push, pop and quit) and encode transitions between pages of information triggered upon receiving of word-link input from the user selecting a menu item. This process of transition from one hypertext node to another is equivalent to the continuous RTN parsing of the user input. RTN links for a single hypertext node (node 8 - "Exercise 1") are shown in figure 3.

As described in the previous section, all HyperCAL nodes require some degree of processing by the user, e.g. all our exercises prompt the user to fill-in a form of data fields which represent the elements of done'. An incorrect input [the, chased, bear, my, brother] will match two possible responses, i.e. 'bad noun' and 'bad verb'. The matched responses are then merged with linking information and displayed to the user in a form of a menu allowing him or her further navigation. The subsequent selection of 'well done' from the menu will allow HyperCAL to go to the next exercise ('push' node '9') on the correct answer, or to further explore nouns or verbs ('push' nodes '2.1.1' or '2.1.3') after the system detected a bad noun or verb in the user input.

This simple example shows that the user-centered model of hypertext dialogue is capable of focusing user attention on only those elements of the information hyperspace which are most relevant to the user needs and expertise. Such a method can be used in the context of a tutoring system to hilite user mistakes and to guide him or her to additional reading material explaining the issues surrounding an error or misconception. It could also be used to fast-track the user through the hyperspace whenever the answers are in agreement with system predictions. It should be noted, though, that our prototype did not make extensive use of the User Model, which at this point in our experimentation was kept to a minimum, simply recording the user successes and failures at decision points.

4. Conclusions and Future Work

This paper introduced the concept of the user-centered model of hypertext dialogue. It was shown that by analysing user interaction with a hypertext system it is possible to reduce a decision space confronting the user at any information node. Our prototype system, HyperCAL, consists of the hypertext interface and a dialogue parser supported by the user and domain models, and the subject organiser. HyperCAL was used to demonstrate how domain model assessment rules and a subject organiser linking information could be used to constrain the navigational space in a hypertext-based tutoring system.

Being only a concept demonstrator, HyperCAL suffers certain conceptual and implementation deficiencies. One of the problems in the current implementation of HyperCAL is its inefficiency, the other more serious, is the lack of an elaborate user model resembling those used in intelligent tutoring systems. To this end, we are currently extending HyperCAL parser with a chart that could be used as a vehicle for modeling user's knowledge. Further information about HyperCAL and its comparison with other hypertext and intelligent tutoring systems may be found in the thesis by Jackway (1994).

Bibliography

Balasubramanian, V. (1994). *State-of-the-art review of hypermedia issues and applications (draft).* Newark, NJ: Rutgers University.

Luger, G. F., & Stubblefield, W. A. (1989). *Artificial intelligence and the design of expert systems.* Redwood City, CA: Benjamin/Cummings.

Nicolson, R. I. (1990). Towards the third generation: the case for IKBH (intelligent knowledge-based hypermedia) environments. In R. McAleese and C. Green (Eds.), *Hypertext: State of the art.* Oxford, UK: Intellect, 166-173.

Nielsen, J. (1990). The art of navigating through hypertext. *Communications of the ACM, 33*(3), 296-310.