In search of Schrödinger's node: dynamic text generation for adaptive hypertext

Maria Milosavljevic
Language Technology Group
Microsoft Research Institute
Macquarie University

Jon Oberlander
Human Communication Research Centre
University of Edinburgh

Robert Dale
Language Technology Group
Microsoft Research Institute
Macquarie University

Abstract

Thanks to the World Wide Web, there are more hypertextual documents in existence than ever before. By its very nature, hypertext opens the door to user interaction with documents. This in turn has resulted in attempts to capitalise on this interaction by building adaptive hypertext systems, where the material presented to the user is in some way tailored on the basis of an existing user model or the user's previous interactions with the system. However, there are limits to the flexibility that current methods afford. We argue for the incorporation of natural language generation techniques into such systems, resulting in what we call DYNAMIC HYPERTEXT. We describe two implemented dynamic hypertext systems, and discuss some of the effects of being dynamic, on both the tailored generation of comparisons, and on what it means to go back, in such environments.

Keywords adaptive hypertext, natural language generation, user modelling, discourse history

Introduction

Technological advances in recent years have given rise to an explosive increase in the quantity of electronic documents; and an increasing proportion of these are hypertextual in nature, being made available via multimedia resources and, more commonly, the World Wide Web.

Hypertext by its very nature opens the door to user interaction with documents. This in turn has resulted in attempts to capitalise on this interaction by building adaptive hypertext systems, where the material presented to the user is in some way tailored on the basis of an existing user model or the user's previous interactions with the system. However, there are limits to the flexibility that current methods afford. We argue for the incorporation of natural language generation techniques into such systems, resulting in what we call DYNAMIC HYPERTEXT.

In this paper, we first briefly summarise the approaches used in adaptive hypertext systems. We then contrast this with dynamic hypertext, which draws on research in natural language generation (NLG)to dynamically create and adapt hypertext documents to the user's needs. We argue that, by making more effective use of a user model and the discourse history, NLG techniques can offer more flexible hypertext documents than traditional adaptive hypertext systems are able to.

The key element that NLG adds is the dynamic construction of textual content on demand, at viewing time. In the famous analogy in quantum mechanics, Schrödinger's cat is neither dead nor alive, so long as it remains unobserved. But as soon as it is observed, the very act of observation causes it to fall into one state or the other [Gribbin 1984]. Dynamic hypertext offers a hypertextual parallel: a link's destination need have no determinate informational contents, until it is visited. As soon as it is visited, the very act of visitation causes a node to contain a determinate content.

We will illustrate the advantages of such 'Schrödinger nodes', and the dynamic hypertext techniques which permit them, through two implemented systems: the ILEX and the PEBA-II text generation systems, which dynamically produce descriptions of entities as World Wide Web pages. We also consider some issues that arise in the dynamic generation of hypertext documents: we describe some of our current work in the production of user-tailored comparisons, and explore what it means to go backwards in such a system.

From adaptive to dynamic hypertext via natural language generation

Adaptive hypertext

ADAPTIVE HYPERTEXT is an area of research which has grown within the User Modelling (UM) and Intelligent Tutoring System (ITS) fields, and which encompasses the goal of adapting hypermedia resources to take account of each individual user's knowledge. Brusilovsky [1996] suggests that an adaptive hypertext system consists of three key components: a fixed network of hypertext nodes, a user model, and the ability to use this model in order to control what is presented to the user at each node, according to her knowledge of the concepts within the document at that node.

Within an adaptive hypertext system, each document may be annotated with the conditions under which particular segments of that document and links to other concepts are considered appropriate given a particular user's knowledge. This enables the system to present different views of the same document to different users.

For a survey of existing adaptive hypertext systems and further elaboration of the concepts involved, see Brusilovsky [1996].

Natural language generation

NATURAL LANGUAGE GENERATION aims to produce coherent natural language text from an underlying representation of knowledge. It can be viewed as a goal-driven planning process, involving the formulation of texts that satisfy some communicative goal. Many of the ideas here are borrowed from conventional planning techniques developed within artificial intelligence; so, for example, a top level communicative goal such as 'instruct the user how to operate a telescope' may be decomposed into a number of constituent sub-goals such as 'tell the user what a lens is', 'tell the user where the focusing mechanism is' and so on. This decompositional process iterates until the resulting goals can each be realised by means of a natural language utterance.

Figure 1a shows the traditional architecture of NLG systems. NLG systems embody two main processing components: the text planner and the surface realisation component.

The text planning stage typically encapsulates all those decisions involving choices of *what to say.* Based on the discourse goals, the text planner must decide what is relevant in a particular situation (*content selection*), and then organise this content in a way that allows realisation of a coherent discourse that guides the hearer's inferences. The text planning component achieves this by composing a discourse plan using facts from the knowledge base. For example, McKeown's [1985] schema-based approach stores a number of plan outlines in a plan library and fills in the appropriate information from the knowledge base.

A model of the user's knowledge can be used by an NLG system to tailor the text to the individual user's knowledge; see Paris [1987] for a good example of this approach. In addition, the ongoing discourse with each particular user can also be recorded in the discourse history component to enable the system to adapt future texts to what has been said before.

The discourse plan is realised as natural language utterances by the surface realisation component. This makes use of knowledge of the natural language's grammar and lexicon to produce well-formed utterances that convey the required semantic content.

Dynamic hypertext

DYNAMIC HYPERTEXT is an area of research within NLG which takes advantage of hypertextual interaction to give the user the freedom to perform high-level discourse planning, thereby reducing the burden on the NLG system of having to reason more deeply about her goals. A key element in any dynamic hypertext system is that the hypertext network and the nodes of this network (the documents themselves) are **dynamically** created at run-time when the user requests them; there are no existing hypertext documents, and there may not even be any pre-existing representations of what *could* be documents within the system.

Figure 1b shows the architecture of a dynamic hypertext system; the traditional NLG architecture shown in Figure 1a is augmented with some additional components which are required within a hypertext environment.

A dynamic hypertext system operates in a similar way to traditional NLG systems; a knowledge base contains information about those concepts in the domain, and the system selects which elements of the knowledge base are important for creating the required hypertext node. The surface realisation component of a dynamic hypertext system must encode html tags into the text in order to produce a document which can be viewed using a hypertext interface such as a WWW browser. The hypertext tags represent follow-up questions which the user can ask, and are generally concepts (or other entities) that can be described by the system.[1] In operation, the user can effectively perform the high-level discourse planning for the system, driving the system by selecting hypertext tags. Each hypertext tag indicates a new discourse goal to the system.

Knott *et al* [1996] provides a useful survey and comparison of existing dynamic hypertext systems. In the next section, we look more closely at two particular systems we have been involved with.

[1] Dynamic hypertext systems must decide whether a link is justified; that is, whether there is more to say about the concept, or whether all the useful information about the particular concept has already been included in the current document.

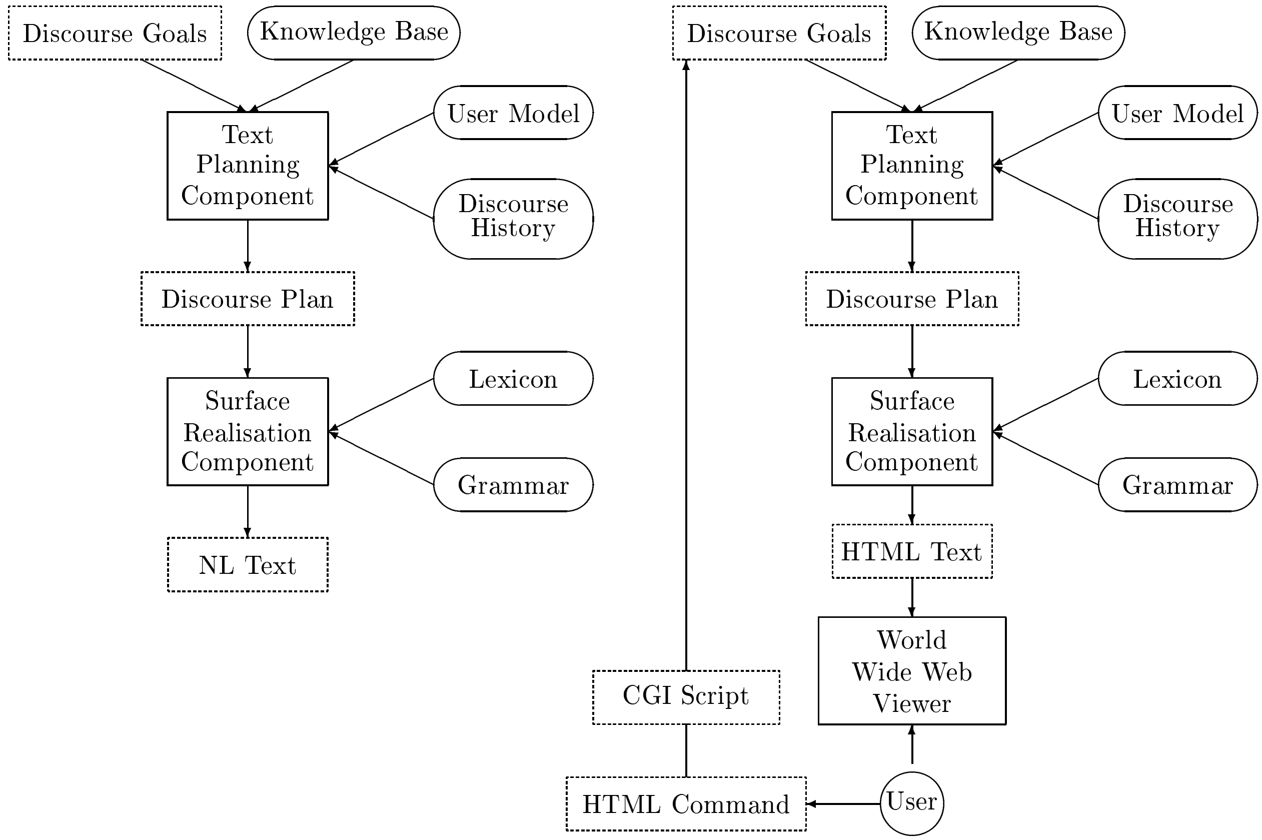

Figure 1: System architectures: (a) traditional NLG; (b) dynamic hypertext.

Two dynamic hypertext systems

The ILEX system

The first phase of the Intelligent Labelling Explorer project has built the ILEX system,[2], which uses NLG technology to generate descriptions of objects displayed in a museum gallery. An overview of ILEX is given here; for more details, see Knott *et al* [1996]. To date, two versions have been implemented (ILEX-0 and -1); both describe objects in the National Museums of Scotland's 20th Century Jewellery Gallery. Figure 2 shows an example description produced by ILEX-0.

ILEX's labels must be: *accurate*; *important*, in the sense of conveying information about the domain that helps educate the visitor more broadly; and *interesting*, in the sense that if the descriptions are boring, the visitor can just walk away. To help meet these criteria, ILEX uses a simple user model, a discourse history, and its own SYSTEM AGENDA of communicative goals. Thus, the user has freedom to explore any object in the gallery at any time, but the descriptions produced are constrained, via the system's agenda of educational goals, which it strives to achieve when the opportunity arises.

Figure 3 shows the architecture of the ILEX system. Comparing it with the general architecture in Figure 1b, it can be seen that the knowledge base has two main parts: information parsed from the museum's (very large) database, and text either entered by hand, or acquired from interviews. Hand-entered information includes type hierarchies for jewels and designers. Interview-based information starts out with canned text STORIES, extracted from interviews with the curator. This is used to represent the *important* messages of the gallery. These stories typically concern individual jewels or classes of jewels, and can be used as

[2] A version of the ILEX system is available on the WWW at URL: `http://cirrus.dai.ed.ac.uk:8000/cgi-bin/jewel-start?start/Ilex1.1`.

part of, among other things, ARGUMENTS and MISCONCEPTION CORRECTIONS. Significantly, however, the canned text used for these stories can be marked up with various kinds of ANNOTATION so that the text can be realised in different ways—pronouns can be used instead of full names, subordinate propositions can be pruned out, and so on, Annotated text has been adopted as a compromise between flexible output and ease of authoring.

ILEX begins by selecting *all* the knowledge base entries and stories about the jewel to be described, and *all* the stories about any *classes* to which the jewel belongs. Comparing Figure 3 with Figure 1b, it can be seen that the main complexity in the generation process lies at the content selection stage, which has three sub-phases: preselection, structuring, and pruning. The information items to be presented are placed in a rank ordering, by computing the value of each item in terms of: its interest to the user-type, importance to the system's agenda, extent of existing assimilation, utility in establishing concepts, and groundedness in the current discourse. Thus, for instance, any stories on subjects that the user is not interested in are effectively suppressed. (The user can specify which subjects she is interested in via a simple form-based preferences page.) From here, any information which has already been presented is replaced via back-pointing hyperlinks, potentially realised via comparisons in the resulting text. The material which remains after this process is slotted into the page schema, and passed off to the realisation component.

The PEBA-II system

PEBA-II[3] is an NLG system which produces descriptions and comparisons of entities represented in a taxonomic knowledge base. An overview of PEBA-II will be presented in this section; for more details, see Milosavljevic, Tulloch and Dale [1996]. Figure 4 contains an example description produced by PEBA-II.

The PEBA-II system uses a fairly traditional NLG architecture, and has two types of discourse goals: to describe a single entity or to compare two entities. In realising these goals, the system makes use of both a simple user model and the discourse history. At the beginning of each interaction with the system, the user is permitted to classify herself as either a naïve or an expert user; this choice results in the system taking different views of the structure and content of the knowledge base of animal facts. The discourse history is used to tailor the output to take account of the previous discourse, as described further below.

[3]A version of the PEBA-II system is available on the WWW at URL: `http://www.mri.mq.edu.au/ltg/peba/`.

PEBA-II uses a phrasal lexicon. That is, the mapping from knowledge base elements to surface form varies from being single words (such as "Yak") to short-phrases (such as "lifespan in captivity") to full phrases (such as "has a long shaggy coat which hangs to the ground like a fringe"). In other words, we only decompose those concepts which require linguistic variation. As with ILEX's annotated storieds, our aim here is two-fold: to be able to populate the knowledge base by avoiding difficult representational issues, and to produce a system which is efficient, not rebuilding the same surface form many times.

Sources of flexibility in dynamic hypertext

This section outlines three aspects of NLG which add flexibility to the idea of adaptive hypertext, helping us to build Schrödinger's nodes.

Co-operative text planning

¿From an NLG system's point of view, an obvious benefit of hypertext over conventional textual output is that the system has less need to pursue highly detailed user modelling. The system does not need to do as much guessing as to what to present the user, since the latter has an active say in the discourse planning process. Another benefit, though, of this *mixed-initiative* approach is that the system can pursue its own goals while servicing those posted by the user.

Thus, in producing descriptions of objects which the user requests, the ILEX system ensures that each description contributes maximally towards the user's goals, and also fits into a coherent picture for the individual user. For example, suppose we are describing an item of jewellery which happens to be made of plastic; then (while this may not be important in itself), it allows the system the opportunity to make one of the points on its own agenda: although we now regard plastic as a cheap material (and thus generally use it in mass-produced jewellery), early this century, it was difficult make, and hence used in one-off designer pieces. In this way, the system's own goals are opportunistically satisfied.

User modelling

Hypertext documents should not be viewed simply as 'information dumps', presenting the user with all the information the system has about some topic. The text should be relevant, understandable and tailored to the particular user's needs. NLG systems can provide more adaptive and flexible hypertext in several ways:

Dynamic link creation: Links to other potential nodes from the current document can be created depending upon the user's behaviour, as with most information-retrieval-based hypertext systems. However, instead of providing a simple list of other nodes which can be visited from the current node, an NLG system can weave the links into the current document, perhaps even explaining why they are important in this particular context, or comparing the concepts described at those nodes.

Dynamic node creation: The content of the hypertext node (or document) can be varied in both the text planning stage—deciding what the content of the document should be—and the surface realisation stage—choosing the linguistic realisation for the content. This allows the content to be adapted to the user's knowledge, the current context, and the content of the previous discourse (see below). The content can be made more informative; information which the user already knows or does not need can be omitted; prerequisite information can be incorporated; and the text can build on the user's existing knowledge so as to provide reassurance. As a result, the comprehensibility of the content can be maximised for the current user, saying neither too much nor too little, and using both linguistic and conceptual material that is accessible for the user.

Discourse history

The importance of being able to refer back to previous conversations with a listener, or more specifically, to concepts previously described to a listener, is obvious when we enter a conversation or debate which has been going for some time, or when we try reading a book from the middle, or start watching a movie which is already half way through. Although initially the new environment would be thoroughly confusing, we can often pick up much of what we missed by listening for the use of discourse history in the conversation, debate, book or movie.

The discourse history plays a central role in NLG systems, particularly when constructing anaphoric referring expressions [Dale 1992], or when trying to refer to concepts mentioned in past communication with a user [Moore 1989].

In designing a hypertext document, the path by which a user might arrive at the node is typically not known (but see Mathe and Chen [1994]). It is therefore difficult to know what concepts might be known by a user reading the document. Dynamic hypertext systems are able to make effective use of the discourse history with a particular user by building on the user's growing knowledge. Concepts which have been mentioned in an earlier discourse can be used to describe new concepts, and in particular, comparisons can be drawn with known concepts. Dynamic hypertext systems are thus more able to tailor the text within documents to the specific user's needs.

Being dynamic

In this section we discuss two areas which illustrate the ramifications of dynamic hypertexts: the treatment of comparisons, and the proper approach to navigation.

User-tailored comparison

Comparison is an important part of the description process as well as an effective tool for introducing new concepts to a user. When describing a new concept to the user, a dynamic hypertext system can make use of comparison in a number of ways. We are currently investigating the use of comparisons in electronic encyclopædia articles in order to build the same functionality into our systems. In this subsection, we describe the types of comparisons found in these articles, and present some issues that arise in the automatic generation of comparisons in hypertext descriptions.

The Types of Comparison Milosavljevic and Dale [1996] describe three types of comparisons found in electronic encyclopædia systems:

- A *direct comparison* is usually made when the user specifically asks for a comparison of two entities. Direct comparisons are *bi-focal*: neither of the two entities being compared are more central to the discourse. PEBA-II produces direct comparisons of animals by comparing each of the property types (eg. length, colouring) which the two animals share, in a point-by-point manner.

- A *clarificatory comparison* is produced when, in describing an entity (the *focused entity*), there is another entity which either (i) is a *potential confuser* of the focussed entity and hence needs to be distinguished or (ii) shares a number of salient features with the focused entity and hence makes a good *comparator*. Figure 5 provides an example.

- An *illustrative comparison* is useful when there is a commonly known entity (or an entity specifically known by this user) which shares a particular feature with the focused entity, and hence can be used to illustrate the property. An example illustrative comparison produced by the PEBA-II system is: *The Aye-aye is about the same length as a domestic cat.*

Incorporating the User's Knowledge When describing a new concept to a listener, we often refer to earlier concepts which the user is familiar with, in order to (i) make the description task easier for ourselves and (ii) ease the understanding of the new concept for the listener. However, within a hypertext description, the comparator can be a hypertext link, and hence if the user is interested, she can request further information about the comparator. The important question here is whether the user has to have knowledge of the comparator in order to make a comparison.

A clarificatory comparison is *important* in a description when there is an entity which could be confused with the focused entity (a potential confuser), and *useful* in a description when there is an entity which shares a number of salient features with the focused entity. Hence, if the focussed entity (for example, the alligator) has a potential confuser (the crocodile) with which the user is unfamiliar, a clarificatory comparison should be utilised regardless. However, in the case where there is a similar entity, it is less obvious whether the comparison should be made. Since the user can select the comparator as a follow-up question, a hypertext environment allows us to make the comparison. In an environment which does not allow follow-up questions, this might not be the case.

Illustrative comparisons are generally made with commonly known entities. Of course, we cannot assume that that every user knows about rabbits, cats and dogs, nor that their common knowledge of these entities is the same universally. In PEBA-II we have a user preferences page which allows the user to specify animals which she is very familiar with. These animals are then used for the illustrative comparisons.

Maximising Coherence through Past Discourse A dynamic hypertext system which is capable of producing comparisons can make considerable use of the discourse history. Within the local textual environment, the system can reinforce the relationship between the previous and current nodes by linking the concepts in each; we refer to this as TEXTUAL COHERENCE. In the more global discourse context, the system can build on those concepts previously described, in a similar way to employing a user model; we refer to this as CONCEPTUAL COHERENCE.

Textual Coherence: As mentioned earlier, the path by which a user might arrive at a hypertext node cannot be predicted in advance, and hence making comparisons with those concepts described in the most recent document will allow the flow from one document to the next to be more coherent. This cannot be achieved by means of the simpler annotations used to mark up documents in straightforward adaptive hypertext systems.

As described in Dale and Milosavljevic [1996], the generation of hypertext documents effectively constitutes an ongoing discourse with the user, and hence the description should provide a connection to the most recently described concept. Both the ILEX and the PEBA-II systems link the focussed entity to the most recently described entity. For example, if the user selects the *Mammal* node when reading the description of the *Monotreme* (as in Figure 4), then the PEBA-II system will produce the following linking text within the resulting description of the Mammal:

> Apart from the Monotreme, the Mammal has the following subtypes: the Placental Mammal and the Marsupial.

The ILEX system currently generates linking sentences such as:

> Like the earlier piece (Gold and enamel pendant necklace), this piece was designed by Jessie M. King (Glasgow) in c. 1905.

ILEX uses these linking sentences to liken the current piece of jewellery to a previously described piece in order to bring the similarities to the user's attention. There are several alternative ways of making the discourse more connected, but the difficulty lies in finding the most relevant relationship between the hypertext nodes. Except where a node has been reached from the top of the system (in ILEX, this is the Cases page), in most instances there should be some characteristic by which to link the two nodes, since the current node must have been listed in the text of the previous node for some reason. It is often this very reason which provides us with some clue as to what to use in the linking sentence.

Conceptual Coherence: A dynamic hypertext system can make use of the longer-term discourse history by making comparisons with those concepts which have been described to the user in the past discourse. For example, in describing the porpoise to a listener, it is often easier to compare it to the dolphin using a clarificatory comparison. If the system has previously described the dolphin to the user then this comparison can easily be made.[4]

Dynamically producing hypertext documents allows the production of shorter texts, since the system can

[4] Of course, it is not entirely clear which aspects of the concepts in the past discourse will be remembered by the user.

include links to those concepts it computes to be relevant, but not relevant enough to place within the current textual node. If the user requires further information about any of those concepts, then she can ask follow-up questions by selecting the links. This helps to alleviate the problem of overwhelming the user with too much text, and increases the likelihood that the user will read the whole text on the node.[5]

In the PEBA-II system, it is assumed that the user reads all the text displayed and, additionally, is a perfect learner—she will remember everything. By contrast, in ILEX, some pieces of information are reiterated; how often depends both on the item's assimilation score, and on the user-type's assumed assimilation rate. It remains to be seen which strategy will prove to be the more effective; it is possible that different strategies will be appropriate in different domains.

Navigational metaphors in dynamic hypertext

Classic approaches to hypertext are based on a spatial metaphor: we visit nodes, navigate around, explore, get lost in hyperspace, and so on. These fit very well the idea that a document is a fixed artefact, whose parts (in this case, hypertext nodes) look the same from whatever direction they are approached. Part of the interest in hypertexts comes precisely from the fact that a given part can have many paths leading from and to it. Given all this, it is appropriate to require that the appearance of any given part of the document not vary from time to time, except perhaps through some wear-and-tear mechanism which indicates how popular the place is—as in visit-counters on Web pages.

Now, recall Schrödinger's node: a link's destination need have no determinate informational contents, until it is visited. Once visited, the very act of visitation causes the node to contain a determinate content. This represents a 'non-classical' attempt to save the spatial, object-based metaphor, in the face of dynamic hypertexts, which are no longer classical, fixed artefacts.

However, perhaps the real point is that dynamic hypertext stretches the spatial model beyond reasonable bounds, so that we must consider a more radical switch to a temporal model, which takes seriously the idea that dynamic hypertexts enable a mixed-initiative dialogue, a conversation between the user and the system.

Such a switch from the spatial metaphor has corollaries for the proper approach to 'navigation' in dynamic hypertext. Both the standard navigational metaphors and the fixed-look assumptions adopted in most hypertext systems seem inappropriate. This becomes most obvious when we look at what happens when we return to an earlier topic of conversation: what does it mean to re-accomplish a discourse goal that has already been satisfied?

As mentioned earlier, a major aim in dynamic hypertext is to allow the user freedom to ask follow-up questions about the concepts in a description. However, suppose that the user did request further information about a concept in a node D_i (and perhaps even some concepts in the resulting node D_{i+1}, and so on), and then returned to the original node D_i. One example of this would be where the user viewed the description of the alligator in Figure 5, requested a description of the crocodile, and then returned to the description of the alligator by selecting the *back* button on the WWW browser. We are faced with a question: should the description of the alligator remain just as it was before, or should it now be revised in light of the user's new knowledge?

Going back to a previous node means, more or less, re-accomplishing a goal that has already been accomplished. We can identify two distinct but related issues here:

- On re-accomplishing a goal that has been accomplished before, should we generate the same text as we produced before, or should we produce a different text that takes account of what has happened in the interim?

- Once a goal has been re-accomplished, should we continue from that point as if it is the first time the goal has been accomplished, or should we make use of discourse history accumulated between the two accomplishments?

In both cases the question concerns what we do with the INTERIM DISCOURSE HISTORY: that part of the discourse history accumulated between two accomplishments of the same goal. We can distinguish various options in the following terms. First, there are two ways we can look at the re-accomplishment of a goal.

Restatement: Posting the same goal again will lead to a new realisation. This is a continuation of the discourse: the interim discourse history figures in how the goal is accomplished this second time around, and we add monotonically to the conversational record. This is what generally happens in human–human conversation when we are asked to re-explain something we have already explained, as in a response to 'Tell me about that again': so, we might explain concept A, find that the hearer has

[5] Many users of hypertext systems not only browse through documents; they skip through the paragraphs of text within them.

a problem understanding, so clarify by introducing a concept B, then return to redescribe concept A using the hearer's new assumed knowledge of B.

Repetition: The system can be asked to recall the way it realised the goal last time (or the first time). This is a verbatim repetition of the utterance made last time, the kind of utterance prompted by a request like 'No, say it exactly as you did before'. In such a circumstance, the reaccomplishment of the goal does not take account of the interim discourse history. The caching mechanisms in Web browsers would cause this to be the default behaviour—which might be appropriate some applications.

There are also two ways we can handle the discourse continuation subsequent to the re-accomplishment:

Resumption: Having re-accomplished the goal, we continue assuming that the interim discourse history is part of the conversational record. In effect, the system knows it has said the same thing twice. Even if the user now follows an unexplored link off the repeated page, all the material presented between these two accomplishments of the goal in question is recalled.

Restoration: A more violent view of repetition is that it is a destructive operation. Having recalled the past, we can take new options from it—and the way these are pursued will not acknowledge that material was presented between the two events. So this is not a continuation of the discourse; instead we cut off a branch of the conversational record, and start to grow the conversation again from an earlier point.

There is a natural pairing between repetition and restoration on the one hand, and restatement and resumption on the other. We are still exploring the most appropriate courses of action in our current implementations.

In the case of PEBA-II, suppose we ask for a description of the alligator; this will contain a reference to the crocodile as a similar animal. If we then request from this page a description of the crocodile, the resulting text will also mention the alligator; if we then go on to request a re-accomplishment of the goal of describing the alligator, this subsequent description will not contain the clarificatory comparison with the crocodile that was included on the first accomplishment, since the user is assumed to now know the difference between these two entities. In ILEX-1, such restatements not only suppress elements of the original description; they also present new information. This behaviour is mediated by the provision of an explicit 'back' button, provided in a separate frame (see Figure 2), rather than by the browser's own back button. The next version of ILEX-1 will also support a 'verbatim' button for any re-visited object.

Conclusion

To search for Schrödinger's node is to seek ways of adding flexibility in the creation of adaptive documents. In this paper, we have argued that adding natural language generation techniques to the construction of hypertextual interactions gives us important leverage. We term the result DYNAMIC HYPERTEXT. We have identified a number of the key characteristics of dynamic hypertext that deliver this flexibility, and described two implemented systems which make use of these ideas. We have also explored in some detail two specific research issues that have arisen in this context: the generation of user-tailored comparisons, and the question of what it means to go back in a dynamic hypertext environment.

References

Brusilovsky, P [1996] Methods and techniques of adaptive hypermedia. *User Modeling and User Adapted Interaction* 6(2-3).

Dale, R [1992] *Generating Referring Expressions.* MIT Press.

Dale, R and Milosavljevic, M [1996] Authoring on demand: natural language generation of hypermedia documents. In Proceedings of *the First Australian Document Computing Symposium* (ADCS'96). Melbourne, Australia.

Gribbin, J [1984] *In Search of Schrödinger's Cat.* London: Wild Wood House. (2nd edition, 1991, London: Black Swan.)

Knott, A; Mellish, C; Oberlander, J and O'Donnell, M [1996] Sources of flexibility in dynamic hypertext generation. In *Proceedings of the 8th International Workshop on Natural Language Generation*, Herstmonceux, Sussex, UK.

Levine, J; Cawsey, A; Mellish, C; Poynter, L; Reiter, E; Tyson, P and Walker, J [1991] IDAS: combining hypertext and natural language generation. In *Proceedings of the Third European Workshop on Natural Language Generation*, pp. 55-62. Innsbruck, Austria.

Mathe, N and Chen, J [1994] Adaptive dynamic hypertext based on paths of traversal. Presented at at the workshop on adaptive hypertext held in conjunction with the Fourth International Conference on User Modeling. Hyannis, Ma.

McKeown, K R [1985] Discourse strategies for generating natural-language text. *Artificial Intelligence* 27:1-41.

Milosavljevic, M and Dale, R [1996] Strategies for comparison in encyclopædia descriptions. In *Proceedings of the 8th International Workshop on Natural Language Generation*, Herstmonceux, Sussex, UK.

Milosavljevic, M; Tulloch, A and Dale, R [1996] Text generation in a dynamic hypertext environment. In *Proceedings of the Nineteenth Australasian Computer Science Conference*, Melbourne, Australia.

Moore, J D [1989] A Reactive Approach to Explanation in Expert and Advice-Giving Systems. *PhD Thesis*, University of California, Los Angeles.

Moore, J D and Swartout, W R [1990] Pointing: a way toward explanation dialogue. In *Proceedings of AAAI90.*

Paris, C L [1987] The Use of Explicit User Models in Text Generation: Tailoring to a User's Level of Expertise. *PhD Thesis*, Columbia University.

Figure 2: A description produced by ILEX-0

Figure 3: The ILEX system architecture

Figure 4: A description of the Monotreme produced by PEBA-II

Figure 5: A description of the Alligator containing a clarificatory comparison with the Crocodile produced by PEBA-II