# Integrating natural language generation and hypertext to produce dynamic documents*

Robert Dale    Jon Oberlander    Maria Milosavljevic
Alistair Knott

**Abstract**

We discuss a task requiring the coherent presentation of heterogeneous information about objects recorded in electronic catalogues. We consider the advantages of combining hypermedia delivery with natural language generation technology, so as to allow us to view a session with such a system as a coherent conversation or dialogue. We describe two prototype systems we have built which make use of these combined techniques, and focus on those aspects of the systems which attempt to provide coherence. Although the techniques themselves are not novel, their combination is relatively recent, and promises to help forge useful tools for accomplishing our specific information retrieval task.

**Keywords** hypermedia, natural language generation, information presentation, discourse coherence, adaptive hypertext

*The support of the Economic and Social Research Council for HCRC is gratefully acknowledged. The ILEX project is funded by UK Engineering and Physical Sciences Research Council, through grant GR/K53321. The second author is supported by an EPSRC Advanced Fellowship. We would like to thank the two anonymous reviewers for their comments.

## 1 Introduction

In this paper, we consider a particular type of task which involves both human–machine interaction and information retrieval: the coherent presentation of heterogeneous information about objects recorded in electronic catalogues. The task is, at root, browsing based, and achievable within a hypermedia framework. In such a framework, the interface presents an index into the underlying database of items of information, with each index entry represented by a hyperlink; following hyperlinks results in the on–demand retrieval of the requested catalogue entry. Each entry may in turn contain links to other items of information that can be requested, thus allowing the user freedom to explore the catalogue in various ways.

Our interest is in maximising the coherence of the texts presented to the reader in such interactive sessions. Our position is that delivering each catalogue entry as a static document is suboptimal in this kind of context; to achieve maximum coherency in the presentation of information, we need to view sessions with such systems as conversations or dialogues, where the content of each entry depends on what has been seen by the reader before. To do this means that we have to provide the system with more communicative sophistication than is possible in a framework where the information returned is in the form of prewritten documents or fragments of prewritten documents.

We take the view that providing systems with the required communicative sophistication requires the addition of natural language generation (NLG) technologies. In this paper, we describe two prototype systems we have built which make use of NLG techniques to add coherence to the presentation of information in hypermedia environments, and we focus on those aspects of the systems which attempt to provide coherence. Neither the hypermedia framework, nor the NLG techniques, should be regarded as novel interaction techniques for information retrieval (IR) in themselves. Nonetheless, their combination is relatively recent, and from them, good tools can be forged for accomplishing our specific IR task.

We begin by presenting brief overviews of hypertext and NLG technologies. Whereas the former is likely to be familiar to—and used daily by—readers of this journal, the latter requires some introduction. Given this background, we define an approach we call DYNAMIC HYPERTEXT, which incorporates techniques drawn from both fields. The principal benefits of the approach are outlined, and the approach is situated with respect to some alternative adaptive approaches. We then consider a family of recent NLG systems which can already be seen as delivering dynamic hypertext, focussing on the two prototype systems we have implemented that generate hypertext documents on the World Wide Web (WWW). Our discussion concentrates on the aspects of these systems that are concerned with the maintenance of coherence. Finally, we conclude by describing some lessons we have learned and some issues that have been raised in the course of this work.

# 2 Background

## 2.1 Hypertext

A hypertext is a network of interlinked pieces of information, or nodes. Someone reading the text (or looking at an image) at a node can access the information at other, related nodes, by following links from their anchor points within the current node, to destinations, which are typically other nodes in the network, but may be locations within those nodes. The WWW, for instance, is a very large, distributed hypertext, used daily by millions of people, who access the information held at nodes within the network by using browsing software, such Netscape Navigator, or Microsoft Internet Explorer.

For a good introduction to issues in hypertext, see Conklin [1987]. For a more up to date discussion, with many references to both research and development, see Nielsen [1995]. The literature is extensive, and there are a number of journals and conferences devoted to topics in the design, evaluation and deployment of hypertext technologies. Since 1987, the Association for Computing Machinery has organised conferences on hypertext, which are now held annually, and whose proceedings contain relevant research papers. Similarly, the Proceedings of the International World Wide Web Conferences are an important resource, focussing on developments concerning the WWW in particular (see, for instance, Genesereth and Patterson [1997]).

## 2.2 Natural language generation

Research in NLG is concerned with the production of coherent natural language text (or speech) from some underlying representation of knowledge. This is in contrast to work in natural language understanding (NLU), where the aim is to go from text to some underlying representation. For many researchers, the representations that serve as input to the generation process are the same as, or similar to, those that might be produced by a language understanding process; but this need not be the case. For example, it is probably true to say that the majority of work in natural language understanding focusses on the construction of representations that are close to formal logical languages (see, for example, the approaches described in Gazdar and Mellish [1989] and Allen [1995]). Although such representations can play an important role in language generation, there are also NLG systems which take as input simple collections of numerical data in tabular format as might be used to represent, for example, stock market data or meteorological data, and produce from these descriptive textual summaries. One interesting difference of emphasis between work in NLU and NLG is that, to date, most work in NLU has focussed on the sentence as a linguistic unit, being primarily concerned with deriving syntactic and semantic representations at that level. Work in NLG, on the other hand, often focusses on larger units of text: multi–sentence paragraphs and multi–paragraph documents. This difference of emphasis means that research in NLG attaches a greater importance to issues of discourse coherence and fluency, which, as we argue below, are precisely the issues that we face in the context of the specific IR task we are addressing.

As suggested by the definition above, NLG is often 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 that we might gloss 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 are specified at a level of abstraction where it is appropriate to realise them by means of natural language utterances.

Figure 1 shows the traditional architecture of an NLG system. These systems generally 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 system's communicative goals, the text planner must decide what is relevant in a particular situation, and then organise this content in a way that allows realisation of a coherent discourse that guides the hearer's inferences. The first task—deciding what should be communicated—is sometimes referred to as CONTENT SELECTION; and the second—deciding how to organise the information for presentation—as TEXT STRUCTURING. The text planning component achieves these two tasks by composing a DISCOURSE PLAN using facts from the knowledge base. Various approaches to this problem have been explored in the literature; McKeown's [1985] schema–based approach, on which our systems are based, stores a number of plan outlines in a plan library and fills in the appropriate

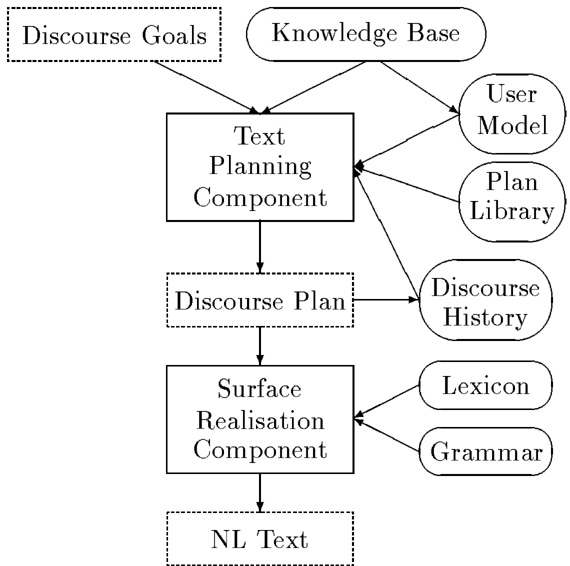



Figure 1: The traditional NLG system architecture

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; see Moore [1995]. We explore these aspects of the process in more detail below.

The elements of the discourse plan can be thought of as semantic specifications for utterances; they are not yet linguistic expressions. Each element in the final discourse plan is realised as a natural language utterance 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. Again, there are a variety of approaches to this task in the literature, the details of which are not important in the context of the current discussion.

For a good introduction to issues in natural language analysis and natural language processing more generally, see Allen [1995] or Gazdar and Mellish [1989]; the broader literature is extensive, being catered for by a number of international journals and conference series. There are currently no introductory texts in the area of natural language generation, but see Reiter and Dale [to appear]. Much of the published research in natural language generation appears in edited collections of papers presented at workshops in the field; see [Kempen 1987; McDonald and Bolc 1988; Zock and Sabah 1988; Dale *et al* 1990; Paris *et al* 1991; Dale *et al* 1992; Horacek and Zock 1993; Adorni and Zock 1996].

## 2.3 Advantages of natural language generation

Any given document is written with a particular audience and a particular context of use in mind. Presenting the same information content, or some variation of it, to a different audience, or for use in a different context, requires the writing of a new document. NLG techniques make it possible to automatically create a variety of documents from one underlying source of information, thus making information provision more flexible, and even making possible some forms or frequency of document delivery that would otherwise be impractical or impossible. The potential benefits of NLG technology are well–rehearsed elsewhere (for example, see Reiter and Mellish [1993]), so we will only list them briefly here:

**Reduced text construction costs:** Writing can be a very expensive task. If the same information has to be presented in a variety of different documents, perhaps expressed slightly differently in each case, the cost is multiplied, even where much of the same material can be reused. Provided the number of different documents to be created is large enough, the expense of this task can be reduced by using a single knowledge base of facts which can be realised into multiple natural language texts using NLG techniques.

**Variation according to purpose:** Often the same essential information needs to be presented in different ways: for example, a reference manual for a piece of software may contain essentially the same information as a user guide or tutorial document for that piece of software, but that information will be organised differently, and different elements will be presented with different emphases. NLG techniques can leverage off a representation of the common underlying knowledge to produce a variety of documents from the one source.

**Variation according to user characteristics:** Users differ in the nature and extent of the background knowledge they bring to bear on a document. NLG techniques make it possible to countenance the creation of documents that are tailored to individual needs, provided of course some computationally tractable characterisation of the user can be obtained (see McKeown *et al* [1985], Paris [1987] and Sparck Jones [1990] for explorations of this idea).

**Variation according to target language:** NLG techniques make it possible to produce documents in different languages using the same underlying knowledge source. At the expense of creating an abstract underlying representation of the information to be conveyed, this approach avoids the problems inherent in achieving high quality machine translation.

**Variation according to resource boundedness:** Sometimes the most appropriate content of some linguistic interchange will depend on the time and space available. NLG techniques enable the provision of different texts for users in different resource–bound situations.

**Interpretation of unfriendly data:** It is often said that a picture is worth a thousand words; but there are contexts where, faced with a large set of statistical or numerical data, it can be hard for a user to determine what is salient and what is not. In many such situations, graphical means may not help; often what is required is a concise summary of what is important. NLG techniques provide powerful mechanisms for generating natural language summaries of large datasets.

These advantages have already begun to result in the construction of practical NLG systems: there are now systems which can generate textual weather forecasts from representations of graphical weather maps (see, for example, Goldberg, Driedgar, and Kittredge [1994]); summarise statistical data extracted from a database or spreadsheet (Iordanskaja, Kim, Kittredge, Lavoie, and Polguère [1992]); describe a chain of reasoning carried out by an expert system (Swartout [1983]); and produce on-line technical documentation (Rösner and Stede [1994]).

## 3 Dynamic hypertext: the integration of NLG and hypertext

Dynamic hypertext is the name we give to the result of merging NLG techniques with hypertextual information delivery: the NLG system creates documents which are nodes in a hypertextual space, with appropriate links between the nodes. The 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 2 shows the architecture of a dynamic hypertext system which operates in a similar way to traditional NLG systems (see Figure 1); a knowledge base contains information about the concepts in the domain, and the system selects those elements of the knowledge base that are important for creating the required hypertext node. In contrast to a typical NLG system, the surface realisation component of a dynamic hypertext system must encode HTML mark–up tags into the text in order to produce a document which can be viewed using a hypertext interface such as a WWW browser. The resulting hypertext links represent follow–up questions which the user can ask, and are generally concepts (or other entities) that can be described by the system. Note that this means that dynamic hypertext systems must decide whether a hypertext 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. In operation, the user can effectively perform the high–level discourse planning for the system, driving the system by selecting hypertext links in the document. The user's actions are sent to the WWW server which then posts a new discourse goal to the NLG system.

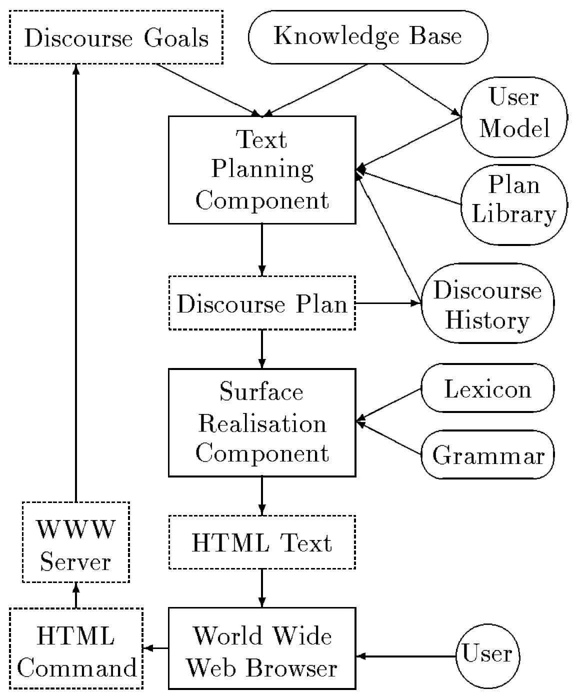



Figure 2: The architecture of a dynamic hypertext system, tailored to dynamically serve Web documents

The result of combining NLG and hypertext is a dynamic environment in which the benefits of each technology is enhanced by the presence of the other.

### NLG benefits hypertext

Hypertext interfaces have been used for many information delivery systems, including help systems and online encyclopædias. The World Wide Web provides a familiar and consistent hypertext interface for a broad audience, and requires relatively little effort to utilise for an information delivery system. Not surprisingly, a substantial amount of online information has been adapted to take up this presentational opportunity.

Employing NLG techniques to automatically generate the textual content of such systems brings a number of advantages, in addition to those that hold for 'conventional' documents. These have been discussed at some length in the previous literature: see in particular Reiter *et al* [1992], Moore [1995] and Milosavljevic *et al* [1996]. For our present purposes, the most significant benefit that NLG brings to hypertext is the scope for using a discourse history to ease the user's navigational burden. In a hypertext environment, we can keep track of the nodes which the user has previously visited, as well as the content which was delivered to the user at those nodes. This allows us to adapt how the user sees the hypertext network and the content of subsequent nodes. By tailoring the presentation of information to take account of the previous context in this way, we may be able to alleviate some of the effects of becoming 'lost in hyperspace' (see Conklin [1987] for an early discussion of this problem).

### Hypertext benefits NLG

Of course, NLG techniques are not without their problems, and many issues in developing and using the technologies remain. It turns out that the medium of hypertext can help overcome some of the outstanding difficulties in NLG. In particular, a big problem in NLG is the task of discourse planning: this involves working out what the structure and content of multi–paragraph texts should be, and often requires access to a detailed model of the user's needs and background. Obtaining and maintaining such a model can be anywhere from very difficult to effectively impossible. Here a real benefit is gained from using hypertext: by presenting her with choices as to what should be viewed next, the user of a dynamic hypertext system can be allowed the freedom of performing high–level discourse planning, thus removing some of this responsibility from the system. The result is an environment where the user has freedom to explore, but this freedom is constrained by the hypertext links which the system provides her.

As a particular hypertextual medium, the World Wide Web provides some additional, practical advantages:

- The number of users who can use the system without additional training is very large.
- A separate interface module does not have to be developed; a dynamic hypertext system need only produce the output in HTML format.
- The incorporation of multi–modal information is inexpensive.

In summary, natural language generation techniques provide effective methods for achieving three desirable objectives:

- we can constantly update the text on the basis of an underlying knowledge source;
- we can vary the text on the basis of a model of the user; and
- we can vary the text on the basis of what has been shown to the user before.

In the next section, we compare the dynamic approach outlined above with other adaptive approaches to hypertext recently proposed in the literature. The paper thereafter focusses on particular dynamic hypertext systems; we firstly review the range of systems to have adopted the approach, and then consider in more detail two current prototypes that we have been developing. These prototypes are then used to illustrate the last of the three benefits of NLG techniques just outlined, and that which most directly overcomes some of the limitations of static hypertext: the use of a discourse history.

## 4 Alternatives to dynamic hypertext

Of course, other methods could be pursued to accomplish the IR task, including more traditional approaches to adaptive hypertext, such as dynamic linking (see Brusilovsky [1996] for a useful review). Here we comment briefly on three existing systems: Trellis, HieNET and SIROG.

The various versions of Trellis are based on a Petri–net model of hypertext (Stotts and Furuta [1989]). The link structure of the hypertext can be modified by manipulating or passing tokens around the Petri net structure, depending on which nodes have been visited, or the period of time spent in a sub–part of the database. Thus, the presence of links can be made conditional on user actions, controlling the accessibility of individual nodes, or entire classes of nodes. Although primarily intended as an authoring environment, Trellis networks could be used for the task we have been discussing.

The HieNET system explictly relies on a Saltonian IR approach (see, for example, Chang [1993]). Here, blocks of text (derived from the SGML structure of the document source) are inter–correlated on the basis of their keyword–based vectors, and a link is formed between blocks when a threshold is reached. User–supplied links can be added and stored, and will alter the overall link structure. Thus, different types of user can be supported by switching between different sets of additional links of this kind.

Finally, Simon and Edmann's [1994] SIROG combines a rule–based expert system with a hypertext document, to provide contextualised on–line help for power station operators. The primary function is to present sub–parts of the hyperdocument relevant to a particular situation, and user–entered data can be used to augment the pre–stored help texts with suggested plans of action.

These systems all allow various kinds of contextualisation and are appropriate to a number of tasks; however, there are a number of differences between them and systems of the type we have been discussing. These differences are most visible if we consider the differing perspectives of content author, system implementor, and user.

Consider first the content author, and the effect of dynamic linking on their task. The difference lies not in the use of dynamic linking *per se*: dynamic hypertext systems will suggest different outgoing links from a generated node, depending on the history of the interaction up to that point. Rather, the difference is that, while they can display user–tailored hypertexts, systems such as HieNET do so by selecting a sub–part of a potentially much larger hyperdocument. To accomplish the task we have set out to tackle using *only* this kind of dynamic linking would require an order of magnitude of extra pre–authored textual nodes. Thus, from an author's point of view, the NLG–based approach may have some advantages.

Secondly, however, consider the system implementor: a potential drawback of our approach, as compared to dynamic linking, is the additional cost of knowledge base authoring. The systems we describe below make use of structured knowledge bases from which to generate text. This certainly limits their generality, but it does not stop them being useful, for at least two reasons. First, the constraint does not mean that they cannot be used in conjunction with relatively unstructured text bases. Indeed, both the systems described here make use of phrasal lexicons, where knowledge base elements correspond to multi–word phrases; and an explicit goal of the ILEX project, host to the second prototype we describe, is to permit the generator flexibly to combine both machine–generated text with large fragments of canned, human–generated text. Secondly, of course, the use of robust parsing techniques on existing text bases—for instance, on the Web—may allow us to represent existing texts at a level deep enough to allow NLG techniques to be re–applied to them. While our methods make room for this possibility, they do not presume it.

Finally, note that from the user's point of view, the differing approaches may actually lead to indistinguishable results. However, a potential advantage of a system using dynamic linking is that it may well serve new pages to the user faster than a system that must generate them on demand. The case is arguable, however, and dynamic hypertext systems have the potential countervailing advantage of producing new pages which cohere more effectively with those that have gone before. These should be more easily read or understood by the user; obviously, this is an empirical claim, which should eventually be investigated through experiment.

## 5 A survey of current dynamic hypertext systems

There are already a number of NLG systems which make use of a hypertext interface. The most important of these are briefly described below.

ALFRESCO [Carenini *et al* 1990, 1993] was one of the first NLG systems designed specifically to operate within a hypertext environment. The system is an interface to a multimedia database of Italian art; its text generation component takes a natural language query as input and produces a page of text containing a selection of relevant hyperlinks to a conventional hypertext system.

IDAS [Reiter *et al* 1992, 1995] was another early NLG system with a hypertext interface. The system delivers technical documentation in response to user requests, and allows users to follow up on requests by clicking on hyperlinks incorporated into the generated texts. Unlike ALFRESCO, IDAS's hypertext environment is completely dynamic: hyperlinks from a dynamically generated page act as further calls to the generation system, rather than pointing to prewritten pages. Both systems are interactive, but while ALFRESCO's interactivity is provided through an escape to a natural language query facility, in IDAS, interactivity is provided by the dynamic generation of appropriate hyperlinks.

In PEA (PROGRAM ENHANCEMENT ADVISOR) [Moore 1995] and MIGRAINE [Buchanan *et al* 1995], both forms of interactivity are provided: users can either click on dynamically generated links or enter unconstrained queries in natural language. MIGRAINE operates in the medical domain, generating information from a patient's medical record in response to queries; PEA is an interface to an intelligent tutoring system for teaching Lisp. Both systems place particular importance on a user's follow-up queries, and on how to generate text which responds to them.

PIGLET [Binsted *et al* 1995] operates in the same domain as MIGRAINE, producing personalised explanations of a patient's medical records. The emphasis in this project is to produce a usable dynamic hypertext system with simple template-based NLG technology, in order to minimise the effort of knowledge-base authoring. As in IDAS, user input is constrained to clicks on hyperlinks.

As these descriptions indicate, a family of hypertext-based NLG systems is beginning to emerge. In the next section we present two additions to this family; these provide the context for some general observations about the benefits and difficulties of dynamically generating hypertext.

# 6 Two dynamic hypertext systems

Our two research groups have independently constructed two systems designed to accomplish the specific task of coherently presenting heterogeneous information about objects recorded in electronic catalogues.

## 6.1 Overview

The PEBA–II prototype system aims to explore the idea of an intelligent on–line encyclopædia. This system takes an underlying knowledge base that contains information in our chosen domain (the animal kingdom) and produces from it textual descriptions, similar in content to encyclopædia entries.

The ILEX prototype explores the notion of intelligent labels for artefacts described in museum catalogues. This system's knowledge base contains information on objects in a set of museum galleries, and produces textual descriptions, similar in content to the spoken descriptions produced by a curator showing the objects to a visitor. The domain currently implemented is the 20th Century Jewellery Gallery of the National Museums of Scotland.

The descriptions created by each system are presented in the form of World Wide Web pages. Figure 3 shows an example output page from PEBA–II which describes the echidna and includes a comparison with the African

## The Echidna

The Echidna, also known as the spiny Anteater, is a type of Monotreme that is covered in stiff, sharp spines mixed with long, coarse hairs. Although it is similar in appearance to the African Porcupine it is not closely related. The African Porcupine is a type of Rodent that has long sharp spines, up to 50cm long, which cover its whole back and can be raised by muscles under the skin. Like the African Porcupine, the Echidna has a browny black coat and paler-coloured spines. The African Porcupine is twice the length of the Echidna (80.0 cm vs 47.5 cm). The Echidna has an average weight of 4.5 kg whereas the African Porcupine has an average weight of 25.0 kg. The Echidna is a carnivore and eats ants, termites and earthworms whereas the African Porcupine is a herbivore and eats leaves, roots and fruit.

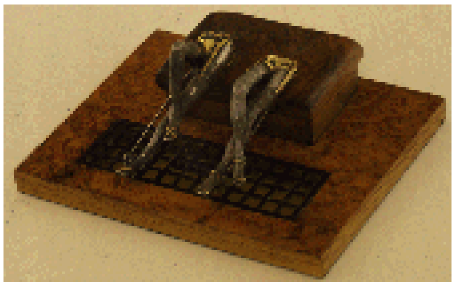
The Echidna has the following subtypes:

- the short-beaked Echidna and
- the long-beaked Echidna.

The Echidna has a small head. It has a prolonged, slender snout. It has no teeth. It uses its extensible, sticky tongue for catching ants, termites and other small insects. It has powerful claws allowing for rapid digging of hard ground. It is found in Australia. It is active at dawn and dusk. It lives by itself. It has an average lifespan in captivity of 50 years.

This text is generated for the novice user level. If you would like the text for the expert user level click here.

Figure 3: A description of the echidna produced by the PEBA–II system.



Pair of brooches on mount
Silver, gold, mahogany, walnut and perspex

This item was made in 1979 and is made of silver, gold, mahogany, walnut and perspex. It was designed by Martin Page who was English. Like the nexklace designed by Flockinger, this item is in the Organic style. Organic jewels tend to be coarsely textured. However, this item has smooth surfaces. With a piece like this, the boundary between 'jewellery' and 'sculpture' or 'art' starts to become quite indistinct. One important theme across 20th Century jewellery has been what to do with a piece of jewellery when it is not being worn. From the 1970s onwards, jewellers have started exploring the idea of turning jewellery into sculpture–so that you can hang it on the wall, or prop it on the mantelpiece when you are not wearing it. This piece works equally well whether it is being worn or being displayed (as at present). Other jewels in the organic style include:

- a pendant necklace designed by Bjorn Weckstrom
- the necklace designed by Flockinger
- a bracelet designed by Flockinger
- a finger ring designed by Frances Beck
- a finger ring designed by Jacqueline Mina
- the previous item
- a finger ring designed by Ernest Blyth

Figure 4: A description produced by ILEX–1.2.

porcupine;[1] Figure 4 shows an example page from ILEX which describes a pendant necklace.[2]

The overall architectures of the two systems are not dissimilar, being essentially that shown in Figure 2. The main difference between them is the fact that ILEX contains one additional component communicating with the text planner: the system agenda. This is ILEX's own set of communicative goals, separate from those posted by the user; we discuss the role of the system agenda further at the end of section 6.3. Beyond this, in each case, the text generator begins with some communicative goal provided by the user (effectively, a directive like 'describe the earrings' or 'compare the echidna and the platypus') and, taking account of the available linguistic resources and contextual constraints, produces a Web page that satisfies this goal.

A document renderer, which carries out the work that is required in order to realise the generated text in some medium, must ultimately play a role in any system which does more than generate disembodied texts (Dale [1992]). In our systems, the document renderer is any Web browser such as Microsoft's Internet Explorer or Netscape, but could equally well be some other component which translates document structuring commands into a visible form.[3] In the current systems the document structuring commands used are a subset of HTML.

In operation, the user guides the system's processing by selecting hypertext

[1] A version of the system is available on the Web, and can be explored with any Web browser via URL `http://www.mri.mq.edu.au/ltg/peba/`.

[2] Two versions of the system are available on the Web, and can be explored with any frames–aware Web browser via URL `http://cirrus.dai.ed.ac.uk:8000`.

[3] Indeed, the commands could instead be translated into audible form, and this is planned for the next major version of ILEX.

links in the document which are then used to indicate new discourse goals; each goal results in the generation of a Web page which contains a number of hypertext links that correspond to a range of further discourse goals the user can choose to pose to the system. This results in a dynamic text planning enterprise where the user decides what information she would like to see on the next page generated. The user can at any time return to the top node of the system, and select from the full set of discourse goals available.

## 6.2 The underlying knowledge representation

The content of the generated texts is derived from a knowledge base of facts. In PEBA–II, these are about animals, and in the current version of the system, the knowledge base has been hand–constructed from an analysis of existing encyclopædia articles. The facts in the knowledge base provide a semantic categorisation of information (see Figure 5) on the basis of which the system can carry out limited reasoning about the knowledge represented, and also provide words or phrases that can be used to realise these facts. In ILEX, the facts are about items of jewellery (and their designers), and the knowledge base has two main sources: information parsed from the museum's (very large) database, and text either entered by hand, or acquired from interviews. Hand–entered information includes hierarchical taxonomies of jewel and designer properties (see Figure 6). Interview–based information starts out with canned text STORIES, extracted from interviews with the curator. These are used to represent the important messages of the gallery, and can be attached at arbitrary levels in the hierarchy.

Both systems use knowledge representations containing taxonomic informa-

```
(distinguishing-characteristic Echidna Monotreme
   (body-covering sharp-spines))
(hasprop Echidna (linnaean-classification Family))
(hasprop Echidna (potential-confusor African-Porcupine))
(hasprop Echidna (nose prolonged-slender-snout))
(hasprop Echidna (length (quantity
                     (lower-limit (unit cm) (number 40))
                     (upper-limit (unit cm) (number 90)))))
(hasprop Echidna (weight (quantity
                     (lower-limit (unit kg) (number 2))
                     (upper-limit (unit kg) (number 7)))))
(hasprop Echidna (geography found-Australia))
(hasprop Echidna (social-living-status lives-by-itself))
(hasprop Echidna (diet eats-ants-termites-earthworms))
(hasprop Echidna (activity-time active-at-dawn-dusk))
(hasprop Echidna (colouring browny-black-coat-paler-coloured-spines))
(hasprop Echidna (captive-lifespan
                     (quantity (average (unit years) (number 50)))))
(hasprop Echidna (nails powerful-claws-rapid-digging))
(hasprop Echidna (head small-head))
(hasprop Echidna (teeth no-teeth))
(hasprop Echidna (tongue extensible-glutinous-tongue-catching-insects))
```

Figure 5: A fragment of PEBA–II's knowledge base

tion. In PEBA–II's case, there is a Linnaean animal taxonomy, where the principal nodes are animal classes and where the arcs, represented using **ako** ('a kind of') links, indicate subset and superset relationships between these classes. A fragment of the knowledge base is shown in Figure 5. For ILEX, the taxonomy is of types of jewels and types of properties of jewels; see Figure 6 for an example of the former classification. In both systems, concepts in the knowledge base are paired with semantic and syntactic structures in a phrasal lexicon, which is used by the linguistic realiser to produce surface natural language expressions. An important aspect of our work here

```
(defsystem
  :name Artefact-system
  :entry-condition artefactual-thing
  :features ((jewellery
                (:require Place-where-worn)
             case))

(defsystem
  :name Jewel-class
  :entry-condition jewellery
  :features ((neck-jewellery
                (:type Place-where-worn 'neck)
                (:require Pendant-Feature)
             wrist-jewellery
                (:type Place-where-worn 'wrist)
             ...)))

(defsystem
  :name Neck-jewellery-class
  :entry-condition neck-jewellery
  :features (necklace
             neckpiece
             collar
             ...))
```

Figure 6: A fragment of ILEX–1.2's knowledge base

is that we make use of a hybrid knowledge representation whereby we can represent lexical, syntactic and semantic information at various 'degrees of cannedness': we only use complex representations where there is benefit to be gained from doing so, and make use of precompiled structures where appropriate. For example, in the PEBA–II system, the lexical entry for the verb phrase corresponding to the semantic element **40cm-long** may be represented as:

$$
\begin{bmatrix}
\text{sem: 40cm-long} \\
\text{orth: "is 40cm long"} \\
\text{syn:} \begin{bmatrix} \text{cat: vp} \\ \text{agr:} \begin{bmatrix} \text{number: singular} \end{bmatrix} \end{bmatrix}
\end{bmatrix}
$$

However, this does not allow the system to reason about this element when making a comparison. As a result, we use complex representations for properties such as length (see Figure 5) when it is important for the system to reason more deeply about the semantics of the property value. This allows the system to produce sentences such as “The African Porcupine is twice the length of the Echidna” and “The Echidna has an average weight of 4.5 kg” as shown in Figure 3.

The type hierarchies used in both systems allow us to infer relationships between instances and classes—for instance, animals and animal classes, items of jewellery and types, or materials and material classes—and to describe these. It also permits inheritance of features so that, for example in the jewellery domain, we may assume that all the instances of Arts–and–Crafts jewellery are elaborate (unless there is information to the contrary). In PEBA–II, the hierarchy also currently forms the main backbone for hypertext generation, as will be seen later.

Each node in the hierarchy serves as a location off which properties of the entity in question can be hung. In PEBA–II, there are two types of properties in the knowledge base. The distinguishing-characteristic clauses single out the important property that indicates how one subtype of a node is distinguished from others (and thus justifies the taxonomic distinction); for example, from

Figure 5, the characteristic that distinguishes the Echidna from all other Monotremes is that it has sharp spines. The hasprop (for 'has property') clauses enumerate the known properties of an entity.

PEBA–II's current knowledge base contains 1137 clauses describing 401 classes; ILEX's current knowledge base contains around 200 classes for the jewellery domain (built on top of a more general hierarchy of domain–independent classes), and around 150 objects, including jewels, designers and sponsors. Another strand of our research efforts is looking at the automatic extraction of this kind of representation from existing texts.

### 6.3 Co–operative text planning

The focus of the research described in the remainder of this paper is in determining what the content of a text should be, and in determining how that information should be organised into a coherent text. This is a major issue for NLG, and it is widely accepted that a model of the user plays a crucial role in determining both the content and organisation of a text. Constructing suitable user models automatically turns out to be very difficult. In light of this, our current work attempts to take advantage of the fact that hypertext navigation of a document source has characteristics that are more akin to dialogues than monologues: the interactivity effectively permits the user to ask questions at any point and to select from a variety of lines of inquiry, restricted only by the set of links made available by the document author. The result is an interesting symbiosis, where the user has the initiative and at the same time is constrained by the set of possibilities offered; the overall character is of a dialogue where the set of questions the user can ask at any given point is restricted. An additional benefit of this mode of interaction is that the system's capabilities are explicit, so the user is less likely to become frustrated than she would by a system which accepts free–form natural language queries but often responds with 'Query not understood' messages.[4]

This symbiosis has been used in the development of a number of other NLG projects which have married text generation with the hypertextual presentation of information: see, in particular, [Moore 1995] and [Reiter *et al* 1992, 1995] described in Section 5. Moore's PEA (Program Enhancement Advisor) system permits the user to ask followup questions by clicking on component words of earlier messages generated by the system as it critiques the user's Lisp code; Reiter *et al*'s IDAS system allows the user to interrogate the system for more information on specific concepts mentioned in on–line documentation. Our systems are closest in spirit to the IDAS system, and share some intellectual heritage (see Reiter and Dale [1992]). However, there are two differences worth noting: first, the fragments of text produced in the systems described here are generally larger, with each corresponding to an entire Web page, whereas in IDAS each contribution from the system is what in some hypertextual systems might be thought of as pop–up glossary items. The larger scale of our systems' discourse contributions allows us to focus on discourse–level issues of the kind discussed in Section 7.

A second difference from earlier systems is that, as we mentioned earlier, in the case of ILEX, there is a separate source providing goals to be satisfied during the interaction. ILEX's labels must of course be *accurate* as well as

[4] Tennant's [1983] NLMenu database interface used a similar idea to overcome problems of this kind for natural language database query tasks.

*interesting*, in the sense that if the descriptions are boring, the visitor can just walk away. But they must also be *important*, in the sense of conveying information about the domain that helps educate the visitor more broadly. To help meet this final criterion, ILEX uses its own SYSTEM AGENDA of communicative goals. Thus, the user still picks the objects to be described, but the system modifies the output descriptions, by including important messages (as determined by the curator), whenever the given object offers a reasonable opportunity to do so.

## 6.4 Using schemas to generate text

The range of texts we are interested in generating is sufficiently invariant that the schema–based approach to text generation, introduced by McKeown [1985], makes most sense. Schemas essentially provide paragraph templates of pre–defined structure, content and order. For example, we can formulate a standard way to describe an animal which includes giving information about its name and taxonomy, distinguishing features, habitat, size and weight, followed perhaps by an example. For an item of jewellery (and for artefacts more generally), the high–level schema comprises a title, a short description, a central paragraph of text called the BODY, and finally a list of pointers to other relevant jewels. In the general case such techniques are too rigid for fluent text production; however, some variation comes from the differing kinds of information available on any given item, and the remaining elements of uniformity themselves have some value in an instructional context.

Each schema provides a set of ordering constraints over a pattern of RHETORICAL PREDICATES in such a way that the resulting text is fluent and coherent; each rhetorical predicate is effectively a representation of a SPEECH ACT TYPE, defined so as to provide an interface to the underlying knowledge representation.[5] In ILEX, the predicates are grouped within the body of the high–level schema. A number of rhetorical devices are used, such as EXEMPLIFY, CONCESSION, COMPARISON and CONTRAST; selection and configuration of these is determined by the particular circumstances in which a page is generated. PEBA–II makes use of two schemas, called IDENTIFY and COMPARE AND CONTRAST. The COMPARE AND CONTRAST schema is represented by the following 'text grammar' rules, where each terminal symbol in the grammar corresponds to a rhetorical predicate:

- CompareAndContrast $\longrightarrow$
  LinnaeanRelationship CompareProperties

  CompareProperties $\longrightarrow$
  CompareProperty CompareProperties

  CompareProperties $\longrightarrow \phi$

The LinnaeanRelationship rhetorical predicate describes how the animals are related according to the Linnaean animal taxonomy. This relationship is determined by traversing the hierarchy upwards from each animal until the lowest common ancestor is found.

We use an underlying corpus–derived categorisation of properties into a hierarchy to permit appropriate comparisons to be drawn; this allows us

[5] Speech act theory (see, for example, Austin [1962]) is a widely–accepted approach to the analysis of language use that takes the view that utterances are actions, and can be reasoned about as such. In the linguistics literature, a distinction is often drawn between the major speech act types of *command*, *request* and *assertion*; most implemented systems further refine and subcategorise speech acts in a fashion similar to that adopted here.

to determine that, for example, height and length are both measurements of size and so can be usefully mentioned together. Using this hierarchy of properties, the CompareProperty rhetorical predicate searches for related properties for the selected animals.

Note here another benefit of using NLG techniques: by representing separately the information about 100 animals, we have the resources available to dynamically generate 4950 different comparisons. This is clearly more space–efficient than storing even a high–probability subset of these documents in canned form. It also makes it much easier to deal with updates to the knowledge represented; changes need only be made in one place, and all subsequent documents generated will reflect this change.

## 6.5 Discourse plans and sentence plans

The effect of instantiating a schema with respect to some portion of the knowledge base is a discourse plan, which specifies the propositional content of the speech acts that make up the text to be generated. Figure 7 shows an example discourse plan. Each speech act specification results in the generation of a surface sentence.

Each speech act may contain references to conceptual entities which are marked as active hypertext links. In the current version of PEBA–II, only entities corresponding to nodes in the animal taxonomy are marked in this way, although in principle any concepts introduced in the text (such as, for example, *spines* in the description of the Echidna in Figure 3) could be so marked. Similarly, while ILEX marks designers and styles as hypertext links, the main outgoing references from a page are collected together in

```
((schema-type compare-and-contrast)
 (constituents
  (((speech-act-type realise-name-entity)
    (content
     ((primary-name ((cat np) (sem alligator) (name-type true-name)))
      (supertype
       ((cat np) (sem family-crocodylidae) (name-type linnaean-name)))
      (relationship ((sem is-a-member-of)))
      (distinguishing-characteristic property
       ((cat vp)
        (property-realisation predicative)
        (property-type nose) (property-value broad-flat-rounded-snout))))))
   ((speech-act-type realise-name-entity)
    (content
     ((primary-name ((cat np) (sem crocodile) (name-type true-name)))
      (supertype
       ((cat np) (sem family-crocodylidae) (name-type linnaean-name)))
      (relationship ((sem is-a-member-of)))
      (distinguishing-characteristic property
       ((cat vp)
        (property-realisation predicative)
        (property-type nose) (property-value narrow-snout))))))
   ((speech-act-type realise-compare-properties)
    (content
     (((speech-act-type realise-animal2-much-more)
       (content
        ((argument1
          ((name ((cat np) (sem alligator) (name-type true-name)))
           (property
            ((cat vp) (property-realisation quantitative)
             (property-type length)
             (property-value
              (quantity (lower-limit (unit m) (number 1.5))
               (upper-limit (unit m) (number 6))))))))
         (argument2
          ((name ((cat np) (sem crocodile) (name-type true-name)))
           (property
            ((cat vp) (property-realisation quantitative)
             (property-type length)
             (property-value
              (quantity (lower-limit (unit m) (number 1.5))
               (upper-limit (unit m) (number 9))))))))))
      ...
```

Figure 7: Initial fragment of the discourse plan for a Compare and Contrast page

categorised lists of related items, at the foot of the page. It is precisely this capability that allows what amounts to shared discourse planning: the underlying system uses schemas to determine what information should be included, but the user can choose to interrogate further at any point. Reiter *et al*'s [1995] IDAS system takes advantage of this observation to limit the amount of information presented at any one time.

## 7 The virtues of combining NLG and hypertext

Our main argument in this paper is that the use of NLG techniques helps makes the hypertexts we create more coherent than they would otherwise be; more generally, this approach provides a better approximation of the 'human–human' conversational task. In this section, we demonstrate this by looking at particular phenomena in both PEBA–II and ILEX.

One property that distinguishes a dialogue from a random sequence of utterances is that, in a dialogue, each conversational participant's contribution is related to the previous utterance by the other conversational participant. This observation is what underlies the notions of question–answer pairs and turn–taking, and other related concepts, in work on conversational analysis within sociolinguistics.

Much of this essential property comes down to COHERENCE: the way that the dialogue contributions flow together and connect to make a seamless whole. A dialogue is not jerky or discontinuous; it is fluent. For a hypertext conversation to be coherent, a number of properties are required:

- Each 'utterance' should be coherently related to the previous one.

Within a monologic context, a considerable amount of research in NLG has looked at the role of rhetorical relations in ensuring that this property holds (see, in particular Mann and Thompson [1988]). Within a hypertextual dialogue, this is controlled by only offering valid rhetorical moves.

- The conversation should flow in terms of thematic structure and focus–shifting. Ideas like those developed in Grosz, Joshi and Weinstein's [1983] centering theory may be useful here: we can think of the set of hypertext links from a web page as offering forward–looking centers, any one of which the user can choose to accept.
- Previously unseen concepts should be introduced appropriately, so that we do not assume knowledge that the user does not have.

In a dynamic hypertext system, the resource that makes coherence possible is a discourse history: a record of the content of the utterances the system has provided to the user. In our current systems, for example, we model this by allocating a unique user ID to each user at the start of a session, then including this as a parameter in all subsequent requests the user poses to the system. This enables the system to maintain an arbitrary number of discourse histories simultaneously. On the basis of this information, the system can construct subsequent texts taking into account the information that has already been presented to any given user.

We can distinguish two kinds of coherence in text:

**Textual Coherence:** a textual contribution should use syntactic resources that are consistent with the thematic development of the text.

**Conceptual Coherence:** the content of a text should not make reference to concepts that are unknown to the reader except in order to introduce them to the reader.

These types of coherence matter both within web pages, and between them. Below, we show how these principles can be embodied in a hypertext system using NLG.

## 7.1 Textual coherence between pages

First, let us consider an aspect of *between–page coherence*. PEBA–II allows the user to explore the Linnaean taxonomy of animals. When we describe an animal, the text schema we use always relates a class to its superordinate class and its subordinates. This means that any node in the taxonomy may be reached from one of two directions: either from the node immediately underneath, or from the node above. The user may also reach a node from a document describing a similar entity which has a hypertext link to the node, or she may commence interaction with the system at any given node, coming on from nowhere, so to speak. Different texts are produced in each case; see Dale and Milosavljevic [1996] for a detailed discussion of this process.

To take the last of these three possibilities first: suppose we are at the beginning of a session with the system where no previous nodes have been visited, and the user requests a description of the marsupial class; we might then generate the following text:[6]

The Marsupial is a type of Mammal that carries its young in a

[6] Underlined words in these examples correspond to hypertext links.

pouch. The Marsupial has the following subtypes ...

However, suppose the user reaches the **marsupials** node from the **mammals** node, as would be the case after reading the following text and then clicking on the **marsupials** link:

> There are three kinds of mammals:
>
> - the monotremes
> - the placental mammals
> - the marsupials

This would result in the following text:

> The Marsupial differs from other Mammals in that it carries its young in a pouch. It has the following subtypes ...

On the other hand, suppose we reach the **marsupial** node from viewing a description of the **kangaroo** class; i.e., after reading the following text:

> The kangaroo is a kind of marsupial which has a powerful tail and back legs.

Here, if we click on **marsupial**, then we reach the **marsupials** node from below, with the resulting text as shown below:

> Apart from the Kangaroo, the class of Marsupials also contains the following subtypes ...

By varying the way in which the information is presented in this way, we can generate more fluent texts. In each case, the way in which the queried class is introduced is determined by how best this fits into the ongoing discourse.

We have only begun to explore the potential here for varying the text to take account of the discourse history: in the example just given, this is analogous to the practice of modifying the introductory section of a document for different audiences but leaving the body of the text substantially the same.

## 7.2 Textual coherence within pages

We now turn to three aspects of *within–page coherence*. Consider the following two alternative fragments of text (from which hyperlinks are omitted) in the ILEX domain; each starts describing the same necklace:

1. This item was made around 1905. This item is made of gold, enamel and sapphire. This item was designed by Jessie M. King. Jessie M. King worked in London. Jessie M. King was Scottish. ...

2. This item was made around 1905 and is made of gold, enamel and sapphire. It was designed by Jessie M. King who worked in London and was Scottish. ...

The first fragment is highly repetitive and arguably incoherent, because it uses the expressions *Jessie M. King* and *this item* three times each. The second—which is what ILEX 1.2 actually generates—is obviously easier to read. ILEX recruits two mechanisms to improve coherence. First, it accomplishes what is generally referred to within NLG as aggregation—the joining together of information that might otherwise be presented in separate sentences. Secondly, the realiser generates the correct referring expressions for the entities under discussion, distinguishing initial from subsequent references: here the necklace is first referred to as *The item*, and then as *It*; subsequent reference to the designer would be realised as *King* or *She*. These mechanisms can be used because (2) is, in fact, a fully–generated text, in the sense that all the information in it is explicitly coded in the knowledge base. However, even canned 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. Such annotated text has been adopted as a compromise between flexible output and ease of authoring. The currently implemented algorithm for generating referring expressions is based on Grosz, Joshi and Weinstein's [1983] centering theory (see Hitzeman, Mellish and Oberlander [1997]); aggregation is the work of Hua Cheng.

Turning to the third method of improving intra-page coherence, consider the following alternative ILEX texts, each of which describes the same necklace:

1. This necklace is in the arts–and–crafts style. ... It has very elaborate festoons. It has faceted stones.

2. This necklace is in the arts–and–crafts style. ... Arts–and–crafts jewels tend to be intricately worked; for instance, this piece has very elaborate festoons. Arts–and–crafts jewels also tend to have cabochon stones. However, this piece has faceted stones.

Again, neither text is ideal. Aggregration of sentences would certainly improve (1), and in either case, the need to establish conceptual coherence (see below) might also require definitions for terms like *festoons* and *cabochon stones*. Nonetheless, (2) is the kind of text ILEX can produce, and the use of generalisations in it helps, in part because without them, the propositions describing the properties of the selected jewel do not in fact stand in obvious relationships to each other. Indeed, exploiting aggregation in (1) as its stands would tend to make matters worse.

To generate text like (2), ILEX's content selection algorithm collects all the simple facts involving the selected jewel. For each fact, a search is made of the generalisations which can be expressed in connection with these facts. For a generalisation to be expressible, the general class it relates to must be introduced by a simple fact (here, *This necklace is in the arts–and–crafts style*); this prevents the expression of the rule from acting as an unexpected topic–shift. The introducing fact is then linked to the generalisation (here, *Arts–and–crafts jewels tend to be intricately worked*) via the coherence relation DEFINITION. The generalisation is then linked back to another simple fact about the jewel, by an appropriate relation. Here, with *for instance, this piece has very elaborate festoons*, it is EXEMPLIFICATION, since the next fact accords with the rule. Note that facts about an individual jewel can also conflict with a generalisation, in which case they are linked to the generalisation by the relation of CONCESSION signalled by *however*, as illustrated in the final two sentences of (2). Other aspects of (2), and the wider use of generalisations to improve textual coherence, are discussed in Knott, O'Donnell, Oberlander and Mellish [1997].

## 7.3 Conceptual coherence

More radical variations in the text are possible if we also strive for CONCEPTUAL COHERENCE. One way of doing this is to 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.

Milosavljevic and Dale [1996] describe the types of comparisons which are found in encyclopædia descriptions of entities. These comparison types are employed in the PEBA–II system in order to produce descriptions which aid the user's conceptualisation of the entities being described. The goal here is to describe new concepts by comparing them to concepts with which the user is already familiar (see [Milosavljevic 1997]). There are two main types of comparisons which are used in order to achieve this goal:

**Clarificatory comparison:** If there is another entity which shares several salient features with the entity being described (the FOCUSSED ENTITY), and if the user is familiar with that similar entity, then a clarificatory comparison of the two entities will aid the user's understanding of the focussed entity. For example, the description of the echidna in Figure 3 includes a clarificatory comparison with the African porcupine. If the user already has knowledge of the African porcupine, then this comparison would aid her understanding of the echidna. Of course, there are other reasons why clarificatory comparisons might be made, such as to distinguish potential confusors.

**Illustrative comparison:** If the focussed entity has a property which is shared by a commonly known or a recently described entity, then an illustrative comparison can be made with that entity in order to allow the user to more easily understand the property and also to make an association between the two concepts. For example, the PEBA–II system produces sentences such as *The platypus is about the same length as the domestic cat.*

The ILEX system also produces illustrative comparisons with those items described in the discourse history. For example, the following sentence is produced in order to highlight the similarities of the focussed entity to a recently described entity:

> Like the previous item, this piece was designed by Jessie M. King (Glasgow) in c. 1905.

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 re–iterated; how often depends both on the item's assimilation score (which is broadly speaking a measure of its 'abstractness'), 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.

## 8 Summary and open questions

Among other things, this special issue called for papers discussing:

- novel interaction techniques for information retrieval;

- conversational modelling of information retrieval tasks; and
- browsing based information retrieval (including hypermedia).

We have described our approach to the task of presenting coherent assemblages of heterogeneous information stored about objects indexed in electronic catalogues. While the techniques are not independently particularly novel, their combination is relatively recent, and seems well–suited to the task. More generally, the combination provides a particular way of extending browsing–based IR in a conversationally appropriate direction.

We have suggested that a shift of metaphors—from navigation to conversation—allows us to think about interactions with a hypertextual system in different ways; and we have argued that natural language generation technologies help to make this shift. We have focussed in particular on what is required to give hypertextual interactions a dialogic character, and we have suggested that a key notion is that of coherence. We have shown how the notions of textual and conceptual coherence play a role in two functioning Web–based natural language generation systems.

It should also be noted that there are some respects in which the metaphor of hypertext–as–dialogue does not quite fit as comfortably as we might hope. In particular:

- In a real conversation or dialogue between two human participants, we can usually assume that the other party is listening to what we are saying, and asking questions when they don't understand what has been said. The situation is somewhat different on the hypertextual systems like the Web: from introspection alone, it is obvious that people do not read Web pages from beginning to end. Rather, they skip around, skim a page, very quickly scan the material to see if there is something of interest. This suggests that, in generating text for the Web, we should assume skimmers rather than good listeners. There is a fine balance to be achieved here: on the one hand, NLG techniques give us a way of avoiding saying things we have said before on some previous page; on the other hand, we have no way of knowing whether the reader even noticed that something was said on a previous page.

- Real conversations only go forwards. Hypertextual navigation aids, however, allow us to go backwards. This raises the question of what should happen when the user asks a second time for a description of some entity. In an NLG–based system, this corresponds to re–accomplishing a goal that has already been accomplished; and so the the system has two choices, which we term RESTATEMENT and REPETITION. In restatement, the reposting of a goal leads to a new realisation of the content, where the interim discourse history—all the things that have been said between the first realisation of this goal and the second realisation—makes a difference. In the case of repetition, we have what amounts to a request for verbatim re–realisation, so that the interim discourse history is effectively ignored for this realisation. Which is the best strategy? This is not clear: on the one hand, users expect things to be pretty much as they were last time they saw them, thus favouring repetition; on the other hand, restatement is closer to what happens in 'real' conversation.

Ultimately, the answers to these open questions depend on the expectations that users develop with regard to this new medium. The use of NLG techniques allow an escape from the restricted interaction of static hypertext, but it is too early to determine how this freedom should be balanced with the responsibility of minimising the disorientation experienced by users.

## References


Adorni, G. and Zock, M. (eds.) (1996) *Trends in Natural Language Generation.* Lecture Notes in Artificial Intelligence. Berlin: Springer.

Allen, J. (1995) *Natural Language Understanding.* Benjamin Cummings.

Austin, J. L. (1962) *How To Do Things With Words.* Clarendon Press.

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

Buchanan, B. G, Moore, J., Forsythe, D. E., Carenini, G., Banks G. and Ohlsson, S. (1995) An intelligent interactive system for delivering individualized information to patients. *Artificial Intelligence in Medicine* **7**(2):117–154.

Carenini, G, Ponzi, M and Stock, O (1990). Combining natural language and hypermedia as new means for information access. *Proceedings of the 5th European Conference of Cognitive Ergonomic (ECCE '90)*, Urbino, Italy.

Carenini, G, Pianesi, F, Ponzi, M and Stock, O (1993). Natural language generation and hypertext access. *Applied Artificial Intelligence* **7**:135–164.

Cawsey, A., Binsted, K., and Jones, R. B. (1995) Personalised explanations for patient education. In K de Smedt, C Mellish and H Novak (eds) *Proceedings of the 5th European Workshop on Natural Language Generation*, Leiden, pp 59–74.

Chang, D. (1993). Hienet: A user–centered approach to automatic link creation. In *Proceedings of the Fifth ACM Conference on Hypertext*, pp145–158.

Conklin, J. (1987). Hypertext: An introduction and survey. *IEEE Computer* **20**:17–41.

Dale, R. (1992). Visible language: Multimodal constraints in information presentation. In *Aspects of Automated Natural Language Generation: Proceedings of the Sixth International Natural Language Generation Workshop*, edited by R. Dale, E. Hovy, D. Rösner and O. Stock. Springer–Verlag.

Dale, R., Mellish, C., and Zock, M. (eds.) (1990) *Current Research in Natural Language Generation.* London: Academic Press.

Dale, R., Hovy, E., Rösner, D., and Stock, O. (eds.) (1992) *Aspects of Automated Natural Language Generation.* Lecture Notes in Artificial Intelligence. Berlin: Springer.

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.

Dale, R. and Reiter, E. (1995). Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science* **19**:233–263.

Gazdar, G. and Mellish, C. (1989) *Natural Language Processing in LISP: An Introduction to Computational Linguistics.* Addison–Wesley.

Genesereth, M. R. and Patterson, A. (eds.) (1997) *Proceedings of the Sixth International World Wide Web Conference*, Santa Clara, Ca., April 7–11 1997.

Goldberg, E., Driedgar, N. and Kittredge, R. (1994). Using natural-language processing to produce weather forecasts. *IEEE Expert* **9**: 45–53.

Grosz, B. J., Joshi, A. K., and Weinstein, S. (1983). Providing a unified account of definite noun phrases in discourse. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, pp44–50.

Hitzeman, J., Mellish, C. and Oberlander, J. (1997). Dynamic generation of museum web pages: the intelligent labelling explorer. Presented at the International Conference on Museums and the Web, Los Angeles, March 1997.

Horacek, H. and Zock, M. (eds.) (1993) *New Concepts in Natural Language Generation.* London: Pinter.

Iordanskaja, L., Kim, M., Kittredge, R., Lavoie, B. and Polguère, A. (1992) Generation of extended bilingual statistical reports. In *Proceedings of the Fifteenth International Conference on Computational Linguistics (COLING–92)*, Volume 3, pp1019–1023.

Kempen, G. (ed.) [1987] *Natural Language Generation.* Martinus Nijhoff

Publishers.

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*, pp151–160. Herstmonceux, Sussex, UK.

Knott, A., O'Donnell, M., Oberlander, J. and Mellish, C. (1997). Defeasible rules in content selection and text structuring. In *Proceedings of the European Natural Language Generation Workshop*.

McDonald, D. and Bolc, L. (eds.) (1988) *Natural Language Generation Systems*. Berlin: Springer.

McKeown, K. R., Wish, M. and Matthews, K. (1985) Tailoring explanations for the user. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp794–798.

McKeown, K. R. (1985). Discourse strategies for generating natural–language text. *Artificial Intelligence* **27**:1–41.

Mann, W. and Thompson, S. (1988) Rhetorical structure theory: toward a functional theory of text organization. *Text* **3**: 243–281.

Milosavljevic, M. and Dale, R. (1996). Strategies for comparison in encyclopædia descriptions. In *Proceedings of the 8th International Workshop on Natural Language Generation*, pp161–170. Herstmonceux, Sussex, UK.

Milosavljevic, M., Tulloch, A. and Dale, R. (1996) Text generation in a dynamic hypertext environment. In *Proceedings of the 19th Australasian Computer Science Conference*, pp417–426. Melbourne, Australia, 31 January–2

February 1996.

Milosavljevic, M. (1997) Augmenting the user's knowledge via comparison. In *Proceedings of the 6th International Conference on User Modelling.* Sardinia, Italy, 2–5 June 1997.

Moore, J. (1995) *Participating in Explanatory Dialogues.* MIT Press, Cambridge, MA.

Nielsen, J. (1995) *Multimedia and Hypertext: The Internet and Beyond.* Academic Press, London.

Paris, C. L. (1987). The Use of Explicit User Models in Text Generation: Tailoring to a User's Level of Expertise. Ph.D. diss., Columbia University.

Paris, C., Swartout, W., and Mann, W. (eds.) (1991) *Natural Language Generation in Artificial Intelligence and Computational Linguistics.* Kluwer Academic Press.

Reiter, E., Mellish, C. and Levine, J. (1992). Automatic generation of on–line documentation in the IDAS project. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pp64–71. Trento, Italy.

Reiter, E. and Dale, R. (1992) A fast algorithm for the generation of referring expressions. In *Proceedings of Coling–92*, Nantes, France, July 1992.

Reiter, E. and Dale, R. (to appear) Building applied natural language generation systems. *Journal of Natural Language Engineering.*

Reiter, E. and Mellish, C. (1993) Optimizing the costs and benefits of natural language generation. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pp1164–1169. Chambery, France, 28

August – 3 September 1993.

Reiter, E., Mellish, C. and Levine, J. (1995) Automatic generation of technical documentation. *Applied Artificial Intelligence*, 9:259–287.

Rösner, D. and Stede, M. (1994) Generating multilingual documents from a knowledge base: the TECHDOC project. In *Proceedings of the 15th International Conference on Computational Linguistics (*COLING '94*)*, Vol. I, pp339–346. Kyoto, Japan.

Simon, L. and Edmann, J. (1994). SIROG: a responsive hypertext manual. In *Proceedings of the ACM European Conference on Hypermedia Technology*, pp108–116, Edinburgh.

Sparck Jones, K [1990] Tailoring output to the user: What does user modeling in generation mean? In *Natural Language in Artificial Intelligence and Computational Linguistics*, edited by Paris, C. L., Swartout, W. R. and Mann, W. C. pp201–225. Kluwer Academic Publishers, Boston.

Stotts, D. and Furuta, R. (1989). Petri–net based hypertext: document structure with browsing semantics. *ACM Transactions on Information Systems* **7**:3–29.

Swartout, W. (1983) XPLAIN: A system for creating and explaining expert consulting systems. *Artificial Intelligence* **21**(3):285–325.

Tennant, H., Ross, K., Saenz, R., Thompson, C. and Miller, J. (1983) Menu–based natural language understanding. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, pp151–158. MIT, 15–17 June 1983.

Zock, M. and Sabah, G. (eds.) (1988) *Advances in Natural Language Generation: An Interdisciplinary Perspective* (two volumes). London: Pinter.