Natural Language Engineering **7** (3): 225–250. Printed in the United Kingdom

© 2001 Cambridge University Press

ILEX: *an architecture for a dynamic hypertext generation system*

M. O'DONNELL, C. MELLISH, J. OBERLANDER

Division of Informatics, University of Edinburgh,
80 South Bridge, Edinburgh, EH1 1HN, Scotland, UK
e-mail: {Mick.ODonnell,C.Mellish,J.Oberlander}@ed.ac.uk

A. KNOTT

Department of Computer Science, Otago University, New Zealand
e-mail: alik@atlas.otago.ac.nz

(*Received 7 July 2000; revised 26 February 2001*)

Abstract

Generating text in a hypermedia environment places different demands on a text generation system than occurs in non-interactive environments. This paper describes some of these demands, then shows how the architecture of one text generation system, ILEX, has been shaped by them. The architecture is described in terms of the levels of linguistic representation used, and the processes which map between them. Particular attention is paid to the processes of content selection and text structuring.

1 Introduction

Hypertext has become an exceptionally popular means of conveying complex information. However, problems arise with this medium because by far the bulk of pages are pre-written: the author cannot predict the exact nature of the reader, and thus must choose one 'typical' reader model to direct the message towards. In addition, the author usually cannot know beforehand in what order the reader will access the pages of a complex document. As a consequence, they often have to provide the same information in several places where it may be necessary for the reader's understanding. The same problem arises with the widely used technique of automatically generating individual pages, or even whole web sites, from databases. In the simple case, these pages are automatic, but they are not context-dependent.

In response to these problems, a number of proposals have been made for the production of *adaptive hypertext* (Brusilovsky 1996). Adaptive Hypertext (AH) systems can be seen as an alternative to Ted Nelson's original notion of 'stretch text'. Stretch text systems allow for the insertion of additional text into the current document, immediately following (or replacing) the selected hyperlink (Kobsa, Muller and Nill 1994). What is displayed at any time is under direct user control. By contrast, AH

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

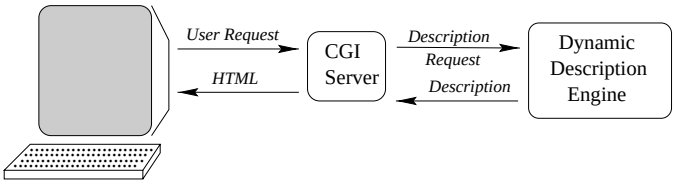


Fig. 1. Generation in a dynamic hypertext environment.

systems automatically manipulate the information that a user receives by varying the content and navigation support according to the circumstances of use. Most AH systems do not actually write text, but rather dynamically link human-written chunks of text. Brusilovsky discusses the use of techniques such as conditionalisation, with the possible text chunks organised according to user models and/or in slots of frames. Some recent papers on AH systems are collected in Brusilovsky, Stock and Strapparava (2000).

We can distinguish AH systems according to the granularity of the information which goes into the system: grains range from whole pages to single words or concepts. The finer the grains, the more options there are for assembling them together, and the greater the scope for context-sensitive adaption in that assembly. Although for most systems the grains are whole sentences or larger, some systems construct the individual sentences of the text. In the simplest case, this is done by template filling, e.g. slotting entity names into partially-complete sentences or paragraphs. At its most complex, with smaller grains still, it may involve full Natural Language Generation (NLG): the assembly of the text word-by-word using knowledge of morphology, syntax, semantics, text structure, and the like. An AH system may even use different grain sizes in different places. For instance, the ALFRESCO system (Carenini, Pianesi, Ponzi and Stock 1993) produces initial dynamically generated pages which then give access to a network of fixed pages.

We follow Dale, Oberlander, Milosavljevic and Knott (1998) in using the term 'dynamic hypertext' to refer to what results when we exploit an NLG system to create documents which are nodes in a hypertextual space, with appropriate links between the nodes; see also Buchanan, Moore, Forsythe, Carenini, Ohlsson and Banks (1995) and Milosavljevic (1999). Crucially, both the hypertext network and the nodes of this network are created *dynamically* at run-time, when the user requests them. Such text is generated on call, not pre-written by a human author. When the user clicks on a link, rather than recalling a static page, the page is constructed by the NLG system, and returned to the browser (figure 1).

Dynamic hypertext can be seen as a kind of multimodal generation. As in other examples of multimodal generation, such as COMET and WIP (McKeown, Elhadad, Fukumoto, Lim, Lombardi, Robin and Smadja 1990; Andre, Finkler, Graf, Rist, Schauder and Wahlster 1993), the integration of NLG and hypertextual technologies can benefit both parts (a point emphasised by Levine, Cawsey, Mellish, Poynter, Reiter, Tyson and Walker 1991). Dynamic Hypertext Generation (DHG) can avoid the problems of static hypertext by taking into account the immediate browsing

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

context. A DHG system can exploit a model of the user to adapt the page to suit the context, choosing what information is provided, and how it should be expressed at a detailed level (down to the choice of individual words). The system can also keep track of what the reader has already been told, and need not repeat such information at later points, thus avoiding the redundancy which often occurs in a statically written suite of pages.

DHG is not without cost however. More data preparation is needed: for truly adaptive text generation, information needs to be stored in some language neutral form. The system needs to be complex enough to know how to express that information in natural language. Also, while the quality of machine-generated text is improving all the time, it is not yet as good as human-written text. So, DHG requires both an investment in preparation, and produces lower quality text. This is the cost of adaptive content.

This paper describes ILEX, a dynamic hypertext system that arose from considering the problem of providing labels for exhibits in a museum gallery. A museum visitor is free to look at whichever objects they choose, in whatever order they like, but this creates problems with the fixed labels that are used to describe the objects. If it may be viewed in isolation, a label must be self-contained. This means that labels may have to repeat background information and may be unable to make educational points that make reference to multiple exhibits. Some museum galleries attempt to solve these problems by restricting visitor movement, but this is not always popular and never works perfectly.

An electronic (e.g. web-based) museum gallery can generate fixed descriptions of the exhibits just as real museum galleries do, but this fails to capitalise on the fact that an electronic gallery has information about what the visitor has already seen. Generating contextually-relevant descriptions requires the generality of a dynamic hypertext system, and ILEX was our attempt to build such a system.

In the following, we will firstly discuss the requirements that a dynamic hypertext environment places on an NLG architecture, using the museum gallery application as a running example. We will then introduce our system and step through it, to give a user's view. We then provide an overview of the system's architecture, and give more explicit details of the major components of this architecture. At each step, we will point out where our architectural choices are responding to the demands of the dynamic generation environment.

2 Requirements for a dynamic hypertext environment

This section introduces a series of requirements which our particular dynamic hypertext system had to satisfy. We argue that these requirements reflect the fact that DHG requires extensions to both static hypertext and NLG technologies.

2.1 Dynamic hypertext extends static hypertext

The ways in which NLG can change the nature of hypertext have been discussed elsewhere (Reiter, Mellish and Levine 1992; Moore 1995; Milosavljevic and Ober-

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

lander 1998); perhaps the most important point is that a discourse history can be maintained and exploited to reduce the user's navigational burden. Thus, dynamic hypertext may be the preferred engineering solution when this is important. Such cases are most likely to arise where: (i) an information resource allows a user wide freedom over their possible paths of exploration; (ii) the order of exploration could influence how an individual node is presented; and (iii) the choice between different presentations of a node makes a material difference to the user's performance with the system. In particular, if a dynamic hypertext system can tailor its information presentation to take account of prior content, it may be possible to alleviate some of the effects of becoming 'lost in hyperspace' (see Conklin (1987) for an early discussion of this problem, and Nielsen (1995) for a survey of ensuing research). If a system uses a discourse history appropriately, then it meets the following requirement:

1. **History-awareness**: Each generated page needs to take into account the other pages which have already been generated. Knowledge of prior pages can be used, for instance,
   - To avoid repeating the same information (e.g. defining a term).
   - To use different strategies for referring to an object depending on whether or not it has already been encountered (e.g. *the Bungweh diamond* vs *a piece called the Bungweh diamond*).
   - To make points that involve re-introducing material from previous pages for specific rhetorical purposes (e.g. making comparisons with an object already encountered).
   - To enhance presentation by referring directly or indirectly to what has already been generated (e.g. using phrases like *also* or *as already mentioned*).

The scenario of a museum gallery is one where the user expects a great deal of freedom and the examples in this paper show how in this domain dynamic hypertext can lead to texts with natural features that cannot be included in static hypertexts. Our evaluations (Cox, O'Donnell and Oberlander 1999) confirm that this naturalness does indeed help users who are trying to solve certain tasks.

2.2 Dynamic hypertext extends text generation

The hypertextual context alters the nature of the generation task itself. In particular, we can identify: (a) features which distinguish dynamic hypertext generation from more familiar text generation tasks, but may also apply in other areas of NLG; or (b) features which are genuinely unique to dynamic hypertext. We can think of these features as characteristic requirements upon the NLG components in dynamic hypertext environments.

2. **Speed**: for non-interactive NLG, processing time is not important, as documents can be generated off-line. In dynamic hypertext, page composition should occur in reasonably short time, so that the user does not get bored. This means that quality may have to be sacrificed in favour of simplicity, and that implementation approaches such as pre-compilation have to be exploited.

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

3. **Limited planning**: the majority of NLG systems have been designed for non-interactive environments, to produce self-contained, stand-alone documents. Dynamic Hypertext requires *interactive* generation that is carried out in collaboration with the user. In dynamic hypertext, the system is only in control of what it returns on the current page. It is the user that controls the sequence of pages (except to the extent that the system can control what links are offered to the user). A system needs to generate coherent sequences of pages without actually being in control of the sequence. Thus, more traditional approaches that require planning ahead are of limited usefulness.
4. **Prioritisation**: in the worst case of limited planning, not only is the page sequence unknown in advance; the length of the sequence itself is unknown. The user can stop the interaction at any point, and thus the overall actual word-limit for the overall discourse is unpredictable. If a system has an agenda of points it wants to make, it needs to manage this agenda without certainty of how many more pages it will generate. Such a system will be continually faced with decisions as to whether to present information on the first opportunity, or to leave it for some later opportunity which may never arise.

These are (a)-type features: the three requirements distinguish a dynamic hypertext system from a traditional, non-interactive text planning system. However, they are features which an NLG system would be expected to possess within the context of a dialogue system, such as Verbmobil (Wahlster 2000). It is true that not all dialogue systems would require all three features. After all, with simple transition network models of dialogue, the system can maintain the initiative. However, once mixed initiative is permitted, these three requirements become applicable.

There is also a (b)-type feature, which only a dynamic hypertext system would have to possess:

5. **Link-awareness**: hypertexts contain links between pieces of text, or between nodes. On the Web, these are typically realised via *anchors* (pieces of text), and associated links; clicking on an anchor will take the user to the destination of the link. In traditional text generation, one has to decide which information to include and which not to include in a text. In a DH system, however, an additional possibility is to *include a link corresponding to extra text that could be generated*. A DH system will need to be aware of the possibility of generating links and use this effectively. In addition, the system will have to generate pieces of text to act as anchors for those links.

2.3 Practical requirements

Finally, to build a particular working dynamic hypertext generator, certain further requirements can be specified. We do not take these to be essential to the general task, or to any other text generation tasks, but they are necessary for a usable, working system to be built:

6. **Limited output**: Generated pages should conform to the visible page. This restriction thus limits the length of the generated text, regardless of the amount of

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

information available. Such constraints are rarely needed for purely text-based NLG but are relevant in a multimedia context (Reiter, Robertson and Osman 1999).

7. **Limited authoring**: hypertext authoring is a difficult task, and the knowledge base authoring required for an NLG system is difficult in different ways. However, the burden can be reduced if the system can import information stored in existing relational databases, and augment this as appropriate with hand-built taxonomies or lexicons.

2.4 Summary

This section has motivated a number of requirements that ILEX should satisfy. The first (history-awareness) is common to many text generation systems, but not to conventional hypertexts; the next three (speed, limited planning, prioritisation) are not common in text generation, but can be required in relatively sophisticated dialogue generation systems; the fourth (link-awareness) is not required in any text generation system outside the hypertext domain. The final two (limited output, limited authoring) are not essential to all dynamic hypertext systems, but are desirable for practical reasons.

3 ILEX

ILEX is named for the '**I**ntelligent **L**abelling **Ex**plorer', a system for generating labels (descriptions) for sets of objects defined in a database. It was initially applied to a museum domain, delivering descriptions of museum artifacts. It has since been applied to a number of domains, including personnel and computer sales catalogues.

There follows a short example of an interaction in the main ILEX domain (Modern Jewellery). As the user enters the system, they are presented with a selection of jewel images, of which they select one. In this case, the user selects a necklace. The page they receive is shown in figure 2.

Page 1 – Spencer Necklace. This page demonstrates several things about the system. Firstly, it has successfully expressed database fields as syntactically correct English clauses. Secondly, it has managed to select quite natural ways of referring to the entities, changing between common NPs and pronominalisation as appropriate. Thirdly, it has managed to join clauses together to create more fluent text, both in terms of *aggregation* (e.g. *This jewel is a necklace and was made by…*), and also in terms of rhetorical relations, as signalled by *indeed* in the fourth sentence. Finally, it has produced a text which is reasonably coherent, although (most) human authors could do a better job.

According to museum curators, one key pedagogical feature of artifact descriptions is the use of generalisations. ILEX is provided with information about general classes of entities, e.g. that Arts and Crafts jewels tend to include rounded stones, and it has put together the facts that this jewel is in that style, and does indeed use rounded stones. If the jewel did not, a suitable exception would have been stated.

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

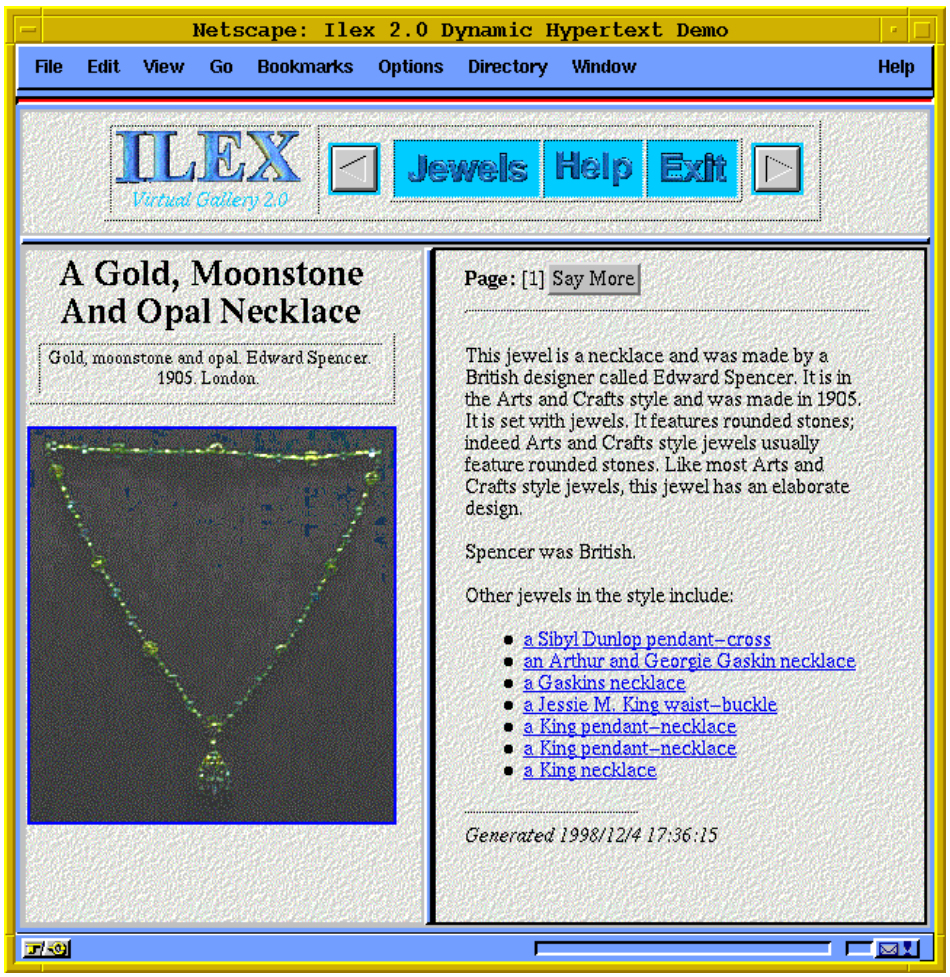


Fig. 2. A first page.

From this point, the user could either ask for more information on this jewel (the "Say More" button above the text), or click on one of the list of other jewels which the system suggested were similar to this one. In this sequence, the user selected the first item in the similar-jewels list.

Page 2 – Dunlop Crucifix (see figure 3). This page demonstrates that the system is adapting the text to the user's path through the system. First, the system indicates its awareness that this jewel is of the same style as the previous item, by inserting the word *also*. In addition, in referring to the third jewel in the list of linked jewels, the phrase *the previous item* is used. Further examples of generalisations are also included.

These examples show that in ILEX the text produced has a very small grain size. This contrasts with systems like HealthDoc (Wanner and Hovy 1996) and Ecran

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

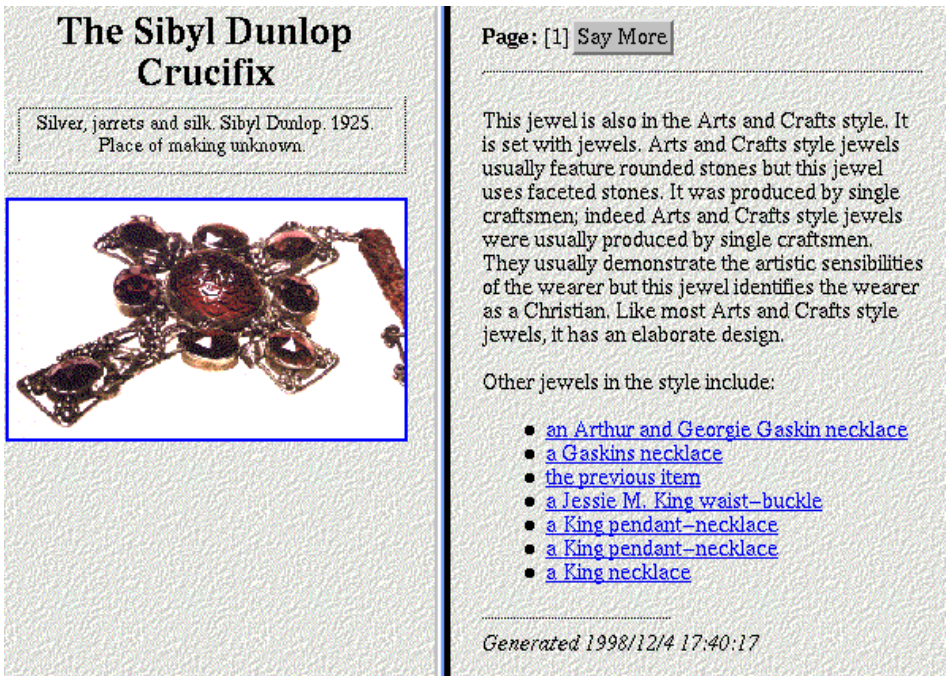


Fig. 3. Jumping to a Related Item

Total (Geldof 1998) which assemble larger text templates or semi-fixed representations from which text will be generated. Such systems, as well as those based on the IRST macronode architecture (e.g. Not, Petrelli, Sarini, Stock, Strapparava and Zancanaro 1998), basically value *speed* and possibly *limited authoring* over *history awareness* and other adaptability, whereas in ILEX a different tradeoff has been aimed for. Systems which accept domain information of similar grain, but express it using templates, include PEBA-2 (Milosavljevic 1999) and POWER (Dale, Green, Milosavljevic, Paris, Verspoor and Williams 1998). Several text generation systems use a similar grain size to ILEX but have chosen rather different architectures (e.g. IDAS (Reiter *et al.* 1992) and PEA (Moore and Paris 1993)).

4 Levels of representation in ILEX

Figure 4 shows the overall architecture of ILEX. Processing can be seen as largely the mapping between levels of representation of information. The rest of this section focusses on the *levels* of representation; the next section describes the *processes* which map between them. At the most abstract level there is information in database form (the domain model), and at the least abstract there is the text produced. There are several layers of representation in between.

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

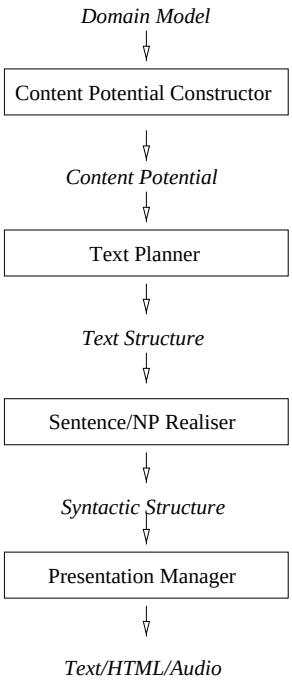


Fig. 4. The ILEX architecture.

4.1 The domain model

The domain model consists of two parts, the *domain database*, which specifies the entities of the domain, and the relations that exist between them, and the *domain semantics*, which specifies how the domain database is to be interpreted, including the taxonomic organisation of entities, relations, etc.

4.1.1 The domain database

An important question in building a text generation system is how the domain database is represented. For our applications, much of the data can be obtained from relational databases (e.g. a museum's artifact database; company product databases; personnel records, etc.), which are *object-centered*, in that the organising entity in the information specification is the object. Eventualities are not specified explicitly, but are implicit in the labelling of relations between entities. For instance, the DESIGNER relation between two database entities hides the process of designing.

We opted for object-centered representation of our domain database, simplifying the task of defining new domains where data is available in relational databases (**limited authoring**). Figure 5 shows a typical object specification from ILEX's Modern Jewellery domain (imported from the museum's database format). For this domain, there are object specifications for items of jewellery, designers, and places. Other objects which are derived from the fillers of the object slots include materials, styles

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

```
(defobject j-240384
  :class jewellery
  :subclass necklace
  :designer king01
  :made-for liberty01
  :date (c. 1905)
  :place "Birmingham"
  :style arts-and-crafts
  :material (gold enamel sapphire)
  :case 1
  :production limited-production
  :qualities (has-festoons has-florals)
  :bib-note "design illustrated in Liberty Pattern Book no 8809"
)
```

Fig. 5. Input specification in ILEX.

and dates. These *defobject* forms provide us with a set of database entities, and between them, a set of database relations (Designer, etc.).

4.1.2 Domain semantics

Before such database records can be used for generation, a *semantics* needs to be provided for the database, in the sense of defining both *how to interpret the domain entities*, and also *how to interpret a relation between two database entities*. We will call these database relations *predicates*, to distinguish them from another type of relation to be introduced later. The domain semantics consists of the following types of information:

- **Taxonomic information**: the domain is specified in terms of a taxonomy of classes, each entity being assigned one of the classes. See figure 6.
- **Predicate definition**: the semantics of each domain predicate is defined using the *defpredicate* structure. See figure 7. It provides information of several types: domain and range restrictions; user-model information (interest, importance, assimilation – see section 4.6); how the predicate should be expressed; and how the predicate can be used for comparison (see section 5.2.2).
- **Generalisations**: for museum education, it is crucial for facts about general types of entities to be asserted, for instance, that Arts and Crafts jewellery tends to be made of enamel. Regardless of their pedagogical importance, generalisations can be used to improve the quality of text, producing object descriptions as in the following:

  This brooch is in the Arts and Crafts style. Arts and Crafts jewels tend to be made of enamel. However, this one is not.

  These generalisations are defined using *defeasible implication* – similar to the usual implication, but expressing a default, rather than a hard principle.

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

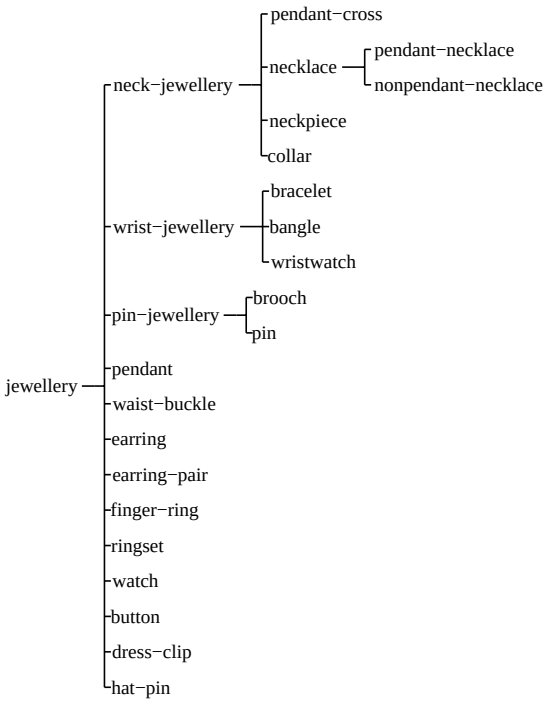


Fig. 6. A Sample entity taxonomy.

Figure 8 shows an example of how a defeasible rule is defined (this could give rise to a sentence like *Most Arts and Crafts jewels are made of enamel*). The rule has a left hand side (antecedent) and a right-hand side (consequent).

4.2 The content potential

To reason about what content to convey to a given request, ILEX re-represents the domain database internally in the form of a directed acyclic graph, representing the entities of the domain and the predicate relationships between them. Each domain object is represented by a node in the graph, called an *entity-node*. Entities may be of two kinds: *specific* – individual entities, such as a specific jewel or person; and *generic* – an entity representing some class of entities, such as Scottish jewellers, or Art-Deco brooches.

It would be most true to the relational database metaphor to represent predicates as arcs between entity-nodes. However, each instance of a predicate (e.g. the fact that the designer King made item J-999) is represented as a node itself, so that there is a way of associating extra information with this small piece of content. Each of these *fact-nodes* is then linked to the entity-nodes via two arcs, *Arg1*, and *Arg2*, representing the domain and range of the relationship respectively (see, for instance, FN-2 in figure 9). The reliance on two-place predicates, though obviously

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

```
(def-predicate Designer
  :arg1 jewellery
  :arg2 person
  :importance ((expert 10) (default 1) (child 3))
  :interest ((expert 10) (default 6) (child 6))
  :assimilation ((expert 0) (default 0) (child 0))
  :assim-rate ((expert 1) (default 1) (child 1))
  :expression (:verb design-verb
               :tense past
               :voice passive)
  :comparison (:variation (string 1)
               :scale nominal)
)
```

Fig. 7. Specifying predicate fillers.

```
(def-defeasible-rule
 :qv ($jewel jewellery)
 :lhs (some ($X (style $jewel $X))(arts-and-crafts $X)))
 :rhs (some ($X (made-of $jewel $X)) (enamel $X)))
```

Fig. 8. Specifying generalisations.

limiting, has advantages both for *limited authoring* and *speed* (since text planning and realisation have to consider fewer cases). Within a fact-node, various other fields exist which detail the polarity, defeasibility, interest, importance and assimilation of the fact. Some of these are inherited from the predicate (see figure 7).

These two kinds of nodes are supplemented with *relation-nodes* which link facts together. These are modelled on rhetorical relations (Mann and Thompson 1988), but are intended to be at a purely informational level (that is, they express things which are true in the domain, rather than having to do with the intentional goals of the speaker (Moore and Paris 1993)). Relations used in ILEX include EXAMPLE, CONCESSION, AMPLIFICATION, SIMILARITY, CONTRAST and RESTATEMENT. As in Mann and Thompson's Rhetorical Structure Theory (RST), each relation-node either has a *nucleus* and a *satellite*, or two nuclei. In a departure from RST, each relation-node has a (possibly empty) set of *preconditions* which are facts that should be assimilated by the reader before the relation can be successfully conveyed.

Since some entities may occur in several object records, the graphs for each object record merge into a large interconnected graph, which we call the *content potential* (see figure 9). Most of the content potential is precompiled before being accessed by a hypertext user (for **speed**), though some aspects of the network change dynamically and have to be computed on demand (e.g. relation-nodes and fact-nodes involved in comparisons).

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

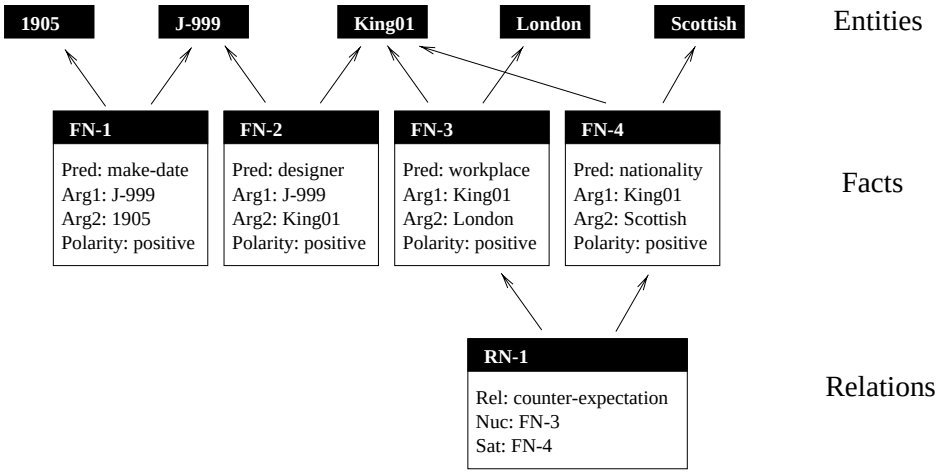


Fig. 9. A small content potential.

4.3 Text structure

When a particular entity is selected as the focus of a description, the system produces a *text-structure*, which is an abstract representation of the text describing the object.

The text structure is a hierarchical tree of *nodes*. The nodes at the leaves of this tree are *text-nodes*, each corresponding to a clause or phrase of the text, and non-terminal nodes are *rst-nodes*.

4.3.1 Text nodes

Text-nodes are the leaves of the text structure. Text-nodes do not have any internal structuring of their own, but rather are defined through pointers to elements at other linguistic levels. These pointers are defined by three slots, similar to those used in Head-driven Phrase Structure Grammar (Pollard and Sag 1994):

- *Sem*: a pointer to the conceptual entity which the text-node is realising.
- *Syn*: a pointer to the syntactic unit which realises the text-node.
- *Orth*: a pointer to the orthographic representation of the text-node: a sequence of characters, punctuation, font, etc.

In generating from a text-node, the end requirement is a completely specified orthography. The process starts with any one of the slots specified and works towards producing the *Orth* slot. It might start with a specified semantics, and generate the corresponding syntactic structure, and extract from this the orthography. If a completely lexified syntactic structure is provided, this can be used to produce the orthographic string. If generating from semantics, then partial syntactic information may be supplied as well, to constrain the syntactic structure used (e.g. specifying that one participant should be realised in the Subject slot). See figure 10.

In some cases, a text-node will have an orthographic string pre-specified, in which case this string is used without need of semantics or syntax. This is the mechanism

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

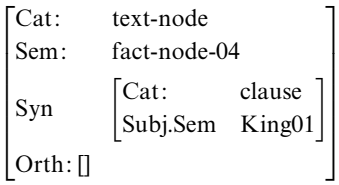


Fig. 10. Partial specification of a text-node.

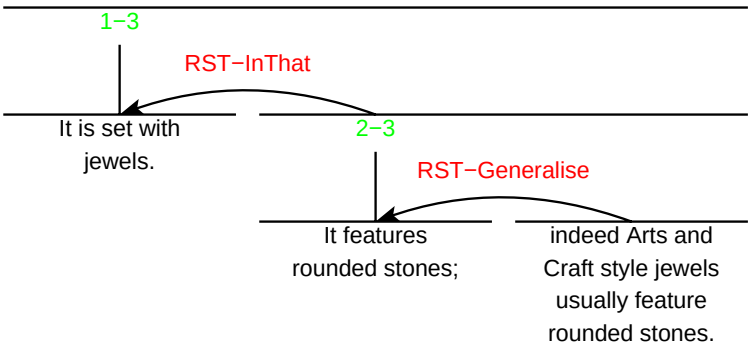


Fig. 11. An RST structure.

through which canned text is implemented in the ILEX system. The ability to use canned text in some places is useful in many applied NLG systems (*limited authoring*, *speed*).

4.3.2 Rhetorical structure

Text-nodes are organised into a dependency structure, using *rhetorical relations*, after RST (Mann and Thompson 1988) – see figure 11. RST allows two ways of linking nodes (whether text-nodes or rst-nodes) together:

- *Nucleus-Satellite relations*: one node is taken to be dependent on the other. The dependent is the *satellite*, the dominant is the *nucleus*. A nucleus may have several dependent nodes, and a satellite may have its own satellites. Typical relations of this kind are ELABORATION, EXAMPLE, CONTRAST, etc.
- *Multi-Nuclear relations*: several nodes are seen to work together, and are of equal status. Examples of multi-nuclear relations are CONJUNCTION, JOINT, SEQUENCE, etc.

In ILEX, the rhetorical relations that can exist between text-nodes all correspond to relation-nodes linking facts in the content potential. The rhetorical relations are thus all informational in nature.

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

4.4 Syntactic structure

The leaves of the text structure are text-nodes. Each text-node is normally associated with a *grammatical-unit*, which is either a clause or phrase. Each grammatical-unit details the constituency of that node, e.g. the syntactic slots present such as Subject, Object, etc. We use a Systemic Functional approach to syntax (Halliday 1994; Martin 1992). The particular grammar is due to O'Donnell (1996).

4.5 Presentational forms

After the text-structure is produced, and each text-node resolved in terms of syntactic structure, the text-structure is mapped onto the final output form. This may involve its expression in terms of HTML, for delivery to a browser. Alternatively, for spoken interaction systems, the text-structure can be expressed in terms of the SABLE speech synthesis markup standard, which is then fed to the Festival synthesis system (Taylor, Black and Caley 1998), which converts text to speech. A plain-text format is also available, for use of the system without an HTML browser.

Text marked-up with SABLE allows the semantic structuring of the text to influence the speech synthesis process, for instance, using intonational contours which distinguish examples from contrastives (see Hitzeman, Black, Taylor, Mellish and Oberlander 1998).

4.6 Representation of context

Apart from these levels of representation of the text, ILEX maintains two types of contextual information:

1. *Informational context*: for *history awareness*, ILEX maintains a representation of what it has delivered:
   - *Discourse History*: a record of which facts have been delivered in this interaction, and which entities have been introduced. These aspects are used in determining how entities and facts are expressed (e.g. *the previous item*, *as already stated*...).
   - *Local and Global focus*: the system keeps a record of the object of focus of the last generated sentence (local focus), and of the focal object of the page (global focus). Among other things, local and global focus affect pronominalisation, focal entities being more open to pronominalisation (Grosz, Joshi and Weinstein 1995).

   Note that the notion of order captured in the system is very simple, distinguishing between past and present and giving a special status only to the immediately previously described entity.
2. *User Model*: ILEX models the user in terms of their relation to information:
   (a) The (assumed) *interest* of the information to the user, e.g. jewellery made of plastic or paper is considered interesting, because such materials are unusual in jewellery.

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

(b) The *importance* of the information to the system's educational goals for this user type, e.g. the system considers it important to educate on stylistic development, so facts about styles are rated highly.
(c) The level of *assimilation* of the information – the degree to which we can assume the reader already knows the information. This could be because the information is part of the reader's assumed world knowledge, or because the reader has previously been told the information. Assimilation of a fact varies between 0.0 (totally unknown) and 1.0 (fully assimilated).

Values for interest and importance are provided for each predicate type in the domain specification (see figure 7). These values are inherited by each fact, and stay constant throughout an interaction. An initial value for assimilation is also inherited from the *def-predicate* form, but the assimilation level of a fact will rise on each delivery. These three values are important in the content selection process, and allow that some facts are, in themselves, more worthy of including in the text. It is only by considering this information that the system can address the problems of *limited planning* and *prioritisation.*

With these simple annotations, ILEX allows different user models to be provided. We have no special theory about where the initial values of the user variables come from: user surveys or curator intuition are possibilities. It is assumed that in general the values are precomputed (possibly from a deeper representation) for speed, but there is nothing to prevent the values being changed at run-time by a system that is tracking the characteristics of a changing user (as already happens with assimilation). As it is, changes to values do not require expensive inference at run-time (*speed*). The result is an approach to user modelling which provides significant short-term session-based modelling, on the basis of simplistic long-term modelling.

5 Processing in ILEX

As shown in figure 4, processing in ILEX can be seen as a simple translation between representations, starting from content potential, and ending with the visual or audio presentation form. On the surface, ILEX takes a pipeline approach to text generation. However, in many places, this model is violated. For instance, the text planner can request addition to the content potential (adding automatically derived comparisons). Also the Sentence Realiser and NP Realiser are interleaved in their calls to each other, in that the sentence realiser will call the NP Realiser to realise an embedded NP, and vice versa.

The first process in the ILEX system takes place off-line. This involves the construction of the content potential from domain information (*speed*). The remaining processes occur when the user requests an object description, and involve the typical two stages of content determination and text structuring. Once the tree of text-nodes has been created, the program creates syntactic units for each node at the leaves of this tree. Syntactic realisation is performed using the sentence generator from O'Donnell's (1996) Workbench for Analysis and Generation. There are procedures to map from the created syntactic structure onto various presentational forms.

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

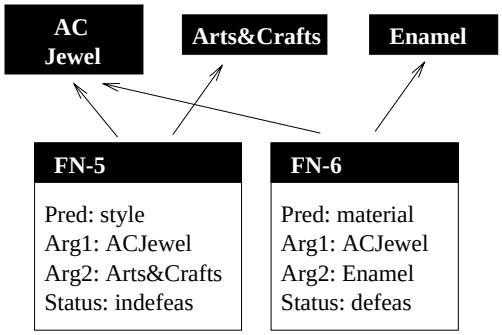


Fig. 12. Asserting generalisations.

5.1 Building the content potential

The first processing stage in ILEX is the construction of the content potential, the representation of the content which is available for expression. This involves:

1. Processing individual *defobject* forms;
2. Processing generalisations;
3. Creating other relations.

Object definitions are broken down into a set of predicate relations between entities, and asserted into the content potential as entity-nodes and fact-nodes. A node is created for each database entity (e.g. one for King01, one for J-999, etc.). Then a node is created for each fact, with arcs pointing from the fact to its two arguments. See figure 9.

The generalisations mentioned in section 4.1.2 represent statements about generic entities, for instance, that *Arts and Crafts jewellery tends to be made of enamel* (Knott, O'Donnell, Oberlander and Mellish 1997). Such a rule is built by first making a generic entity to represent the class of jewels which match the LHS element (Arts and Crafts jewels). A fact (annotated as defeasible) is then created to represent the relationship as a whole. See figure 12.

Some other information is derivable from these rules, by comparing the generalisation to instances. Where a particular jewel matches the generalisation, a number of relations can be asserted, such as EXEMPLIFICATION and AMPLIFICATION. And for those jewels which actively conflict with the generalisation, other relations such as CONCESSION can be asserted. Also, as well as moving from the generalisation to the instance, it is possible to move from the instance to the generalisation, asserting a GENERALISE relation. See figure 13.

Relations are also derived from other sources in the domain model, for instance by detecting subsumption relationships between facts and by detecting predicates that could be related by comparisons (see section 5.2.2).

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

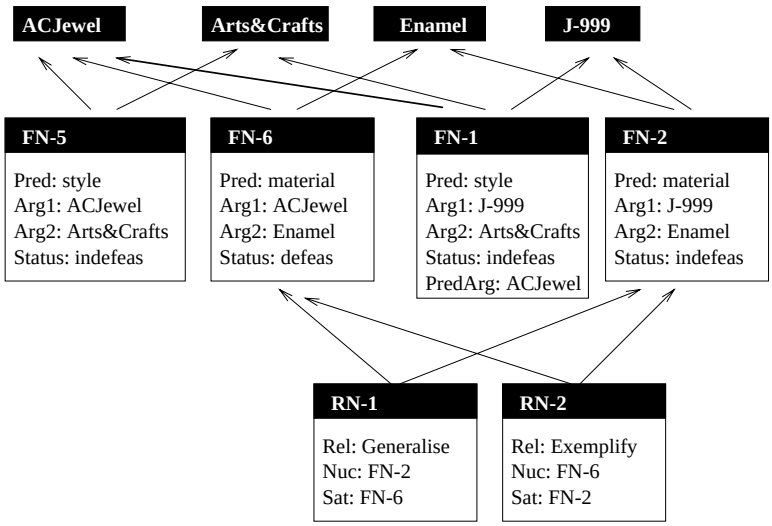


Fig. 13. Relations arising from generalisations.

5.2 Text planning

To compose a text, one must decide both what to say (content selection), and also how to organise it (text structuring). This section first outlines the motivations for selecting our general approach to text planning, and then gives details of the content selection and text structuring processes in ILEX.

5.2.1 An opportunistic approach to text planning

Our requirements create problems for existing approaches to text planning. Existing plan-based generation systems, such as the Moore-Paris planner (Moore and Paris 1993), require a unitary *goal* for a text, and it is planning backwards to achieve this goal that in the end justifies all the content of the text. An alternative text planning method is *schemas*, as in McKeown's early work (McKeown 1985), or as used in the KOMET project (see Bateman and Teich 1995). Schemas produce texts that are reasonably fixed in structure, which is excellent when there is a stereotyped way of organising texts in some domain.

Neither of these approaches addresses the idea that ILEX, when asked to describe an object, must decide what can be said that will address the user's interest, the curator's notion of importance (*prioritisation*), which will exploit the history of interaction (*history awareness*) and which can nevertheless be said coherently within the space available (*limited output*), without the luxury of planning ahead the whole dialogue (*limited planning*). In such a situation, the system must plan to exploit the different *opportunities* that are presented to it without having any means of predicting when or in what form these opportunities will appear. The solution to this problem is one of the more distinctive aspects of ILEX.

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

5.2.2 Content selection

The content selection algorithm is based on the idea that information included in the text must be in some way *relevant* to the focal object of the text.

Calculation of Relevance. Relevance can be defined in terms of the content potential: A fact is relevant to the entity of description if:

1. it directly describes the entity, and is thus linked directly to it.
2. it provides background on an introduced participant, i.e. is linked to a relevant fact via a shared participant, e.g. when describing a brooch by naming its designer, facts providing background on the designer also become relevant.
3. it provides background on the fact itself, i.e. the fact is linked to a relevant fact via an informational relation. The facts linked to a fact via CONTRAST, GENERALISE, SIMILARITY, etc. in our content potential all shine some more light on the relevant fact, and thus perhaps on the entity in focus.

Given the recursive application of these criteria, and the high degree of interconnectedness of the content potential, it might be the case that *all* facts in the content potential are relevant. For this reason, we need to take into account *degrees of relevance* – the further away from the focal entity we move, the less relevant the information becomes. In addition, some types of connections between facts preserve relevance better than others. It may be more desirable to include a fact connected via a GENERALISE relation than one connected via a common entity. For this reason, we assign each relation type a level of relevance-maintenance, ranging from 1.0 (no lessening of relevance) to 0.0 (no preservation of relevance). If GENERALISE is assigned a high level of relevance-maintenance, then facts connected by this relation will be included in the text before facts connected by weaker connections.

The implicit links between facts by virtue of shared entities also have levels of relevance-maintenance. The link between two facts which share (only) a common *Arg2* value is, however, given a relevance-maintenance level of 0 because the transition between such facts usually disrupts coherence. This follows from the fact that for a given predicate, there are in general many *Arg1* values that correspond to any given *Arg2*. So after *This item was designed by J. G. Burghardt* it is strange to say that another item is also designed by him unless that item is special in some way. The other items with the same designer can be (and are) pointed out to the reader via the forward-pointing links at the end of the page. This is a case where the decision of what is included in the text is influenced by what can easily be made available via hyper-links (*link awareness*).

The structural relevance of any fact is derived by multiplying together the relevance-maintenance levels of the links back to the entity being described. For instance, if a GENERALISE relation has a level of 0.9 and a link through a common entity has a level of 0.5, then a generalisation on a fact linked through a common entity would have structural relevance of 0.45. If multiple paths exist, the highest score is assumed. Although we have given exact arithmetic details of the calculations here for concreteness, this is best seen as an example of how calculations could

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

be done and we do not wish to claim that there is anything optimal about this particular approach. See also Rino and Scott (1996) for a related approach.

Intrinsic Score. To account for the fact that some facts are of more interest and importance to the user in themselves, the interest, importance and (1 – assimilation) of each fact are multiplied together to calculate the *intrinsic score* of each fact. This represents how valuable the fact is to say, regardless of what object is currently being described. To find the overall value of including the fact in the text, the node's intrinsic score needs to be combined with its structural relevance. At present this is done via a straight multiplication.

The Algorithm. Content selection is an iterative process of working out from the focal object, calculating the relevance of each fact encountered and returning the best *n* facts. Since relevance decreases with distance, the search can follow the most promising close facts first and abandon searching further from a fact once *n* facts with greater relevance have been found. Since an indirectly related fact will only be chosen if the intermediate (and thus higher-scoring) facts on the path to it are also included, the resulting text will have a way of relating each fact chosen to the focal object. The algorithm also has the properties of allowing digressions if there is not much material on the focal object, and of including facts that are (almost) assimilated if these give access to more remote, but highly valued, facts.

'Barrel Scraping'. If the content selection process cannot find a sufficient number of facts, it enters a 'barrel-scraping' phase, to locate new content. This process may introduce two types of new content to the content potential:

1. *Negative polarity facts*: expressed for instance, as *This item is not made of enamel*, are not pre-stored, but are generated as needed from existing positive polarity facts only in contexts where the reader has grounds for believing the proposition which is negated (because they have assimilated a defeasible rule).
2. *Comparisons*: comparisons to items related to the focal object may be made. The choice of appropriate item to compare to is determined via a measure of the similarity between pairs of fact-nodes. The metric used for similarity is that developed by Milosavljevic (1999). Facts of different types require different notions of similarity to be invoked. As well as storing information about individual facts, the system also stores information about fact types, which is used to determine what facts can be compared, and when it is instructive to compare them. Each fact type is associated with two fields: *variation* and *scale* (provided by the def-predicate form, see figure 7). The former is a number (between 0 and an unspecified upper bound) representing how much variation there is between items with this attribute. *Scale*, meanwhile, represents the type of scale that the attribute is associated with (nominal, interval, ratio or ordinal). For some scales, the variation can be given in advance, while for others it must be calculated by looking at the values for the collection of objects that the system knows about. This calculation is carried out off-line,

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

when the content potential is built. The similarities between the facts associated with candidate objects are determined by checking the values they take on the various scales, and the amount of variation possible on those scales. The most similar object will be found, together with the facts that make it similar to, and different from, the current object.

5.2.3 *Text structuring*

After the relevant content is selected, it is necessary to construct a text structure which represents a coherent, contextually appropriate, expression of the content. This process has four steps.

1. Map each fact-node to be generated onto a text-node;
2. Organise these text-nodes into a valid text structure (linking text-nodes via rhetorical relations);
3. Linearise the text structure: assign order to segments at each level of structure;
4. Generate forward pointers to related items to the one being described, e.g. *Other items by Jessie King include*....

The process of mapping each fact-node onto a text-node is simple: a text-node is created for each fact-node, whose *Sem* field points back to the fact-node. The syntactic realisation of that text-node is more complex and will happen later. For organising the text-nodes into a rhetorical structure, three main text-planners have been implemented within the ILEX system. Here we describe the one which is used in the demonstration system.

In theory, the text structurer could be designed around a module that constructs all possible Rhetorical Structure (RS) trees from a given set of facts, applies an evaluation function to each tree, and selects the highest-scoring tree to generate. A number of other text planners have used this approach, in particular Marcu (1997). One of its main problems is complexity: the number of trees grows at least as fast as the factorial as the number of facts.

Heuristics must be applied to prune the space of possible trees. In practice, the ILEX text structurer overcomes the complexity problem by building a series of local trees, and using a separate, simpler algorithm to link these trees together. The local trees are constructed by looking for all possible trees except those that involve the RST relation OBJECT-ATTRIBUTE ELABORATION. This relation applies between two facts when one fact mentions an object and the other describes an attribute of this object. The looseness of this definition is responsible for a great deal of the combinatorial explosion of possible trees. Moreover, there are good reasons for distinguishing OBJECT-ATTRIBUTE ELABORATION from other RST relations. The relation only holds between two propositions indirectly, as a result of the objects which the propositions are about; consequently it is better thought of as expressing constraints on how the focus of a text can move from one entity to another. We argue that these constraints apply at a global, rather than a local level.

At the global level, we define a coherent text as a linear sequence of focus spaces or *entity-chains*, each of which focuses on a single entity. The only requirement for

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

coherence is that each entity-chain except for the first must be 'introduced' by an entity-chain somewhere to its left. An entity-chain C_2 is introduced by another entity-chain C_1 if the entity that C_2 is about is mentioned by a fact-node somewhere within C_1. The relation of 'introducing' is thus analogous to the relation of ELABORATION; however, it applies between sets of fact-nodes, rather than individual ones. Note that it does not matter where in C_1 the entity is mentioned, or whether there are any entity-chains intervening between C_1 and C_2. So the relation of 'introducing' is in fact weaker than ELABORATION. Even allowing for this, the task of searching for a legal ordering of entity-chains is much simpler than that of finding all possible trees if ELABORATION is included as a relation. See Knott, Oberlander, O'Donnell and Mellish (to appear) for details of this model of descriptive text.

The planner begins by taking the set of fact-nodes delivered by the content selection module, and extracting a subgraph from the content potential containing all the entity-nodes pointed to by these fact-nodes, together with all relation-nodes which link pairs of fact-nodes, regardless of which paths were traversed by the content selection module. It then proceeds in two independent directions. Firstly it builds a set of entity-chains, which are sets of fact-nodes which have the same *Arg1*. Secondly, it searches for the best rhetorical structure tree (minus ELABORATION) that can be created from the complete set of fact-nodes, regardless of which entities they are about. It then extracts the individual fact-nodes that make up the tree from their entity-chains, and inserts the whole tree into an appropriate entity-chain (the one associated with the fact-node which is reached by following a chain of nuclei from its root). It then checks whether a legal sequence of entity-chains exists; if it does not, it tries the same procedure on the next-best tree. The whole process is then iterated, and the set of trees containing facts not yet incorporated into a tree is produced. The algorithm finishes when no more trees can be added to the entity-chains structure.

At this point, two post-processing operations are performed. The content-selection module is actually set up to find more facts than are asked for by the user. The extra facts are removed when structuring is complete; thus, the usefulness of a fact in generating interesting structure is an additional factor in the selection of that fact. The fact-removing mechanism can be set up either to remove whole entity-chains (beginning with the rightmost one), or to remove individual fact-nodes within entity-chains. The latter strategy guarantees the most densely-structured 'first page', but means that subsequent pages might not contain so much interesting structure. The former strategy reserves some interesting structures for later pages, but runs the risk that these structures might never be seen. The choice of strategy, together with the number of extra facts, are user-configurable parameters.

The second post-processing stage is the aggregation of individual fact-nodes using sentence conjoining, and the embedding of fact-nodes within NPs; see Cheng (1998) for details.

At the end of each description, ILEX generates a separate block of forward pointers to a limited number of other items with the same designer. The decision has been made to have descriptions of these available only via hypertext links (*link awareness*). The hypertext links are anchored with NPs produced by the NP

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

planner (see below). This means, for instance, that items already seen are likely to be indicated by definite phrases and new items by indefinite ones.

5.3 Text realisation

The next processing step involves producing syntactic structures corresponding to the text structure. This mapping happens at three levels:

1. *Multi-Clausal structures*: rhetorical relations in the text structure are realised via clause complexes, the aggregation of clauses with common subjects, and discourse connectives between sentences.
2. *Clauses*: text-nodes which correspond to facts in the content potential are realised via clause structures. The *def-predicate* form provides details of how a predicate is mapped onto a clause structure – see O'Donnell, Knott, Oberlander and Mellish (2000) for details. Variation in syntactic form includes tense (past/present/future), voice (active/passive), mood (declarative/imperative), finite vs. nonfinite forms, negation, addition of adjuncts, and adverbials, etc. This latter includes the resources for indicating *history-awareness* (*as already mentioned*, *also*) and generalisation (*usually*).
3. *Noun Phrases*: text-nodes which correspond to entities in the content potential are expressed as (potentially complex) noun phrases. The NP Planner takes into account both what is known about the entity and the context of reference to produce an appropriate referring expression, including the embedding of clauses as a form of aggregation (see O'Donnell, Cheng and Hitzeman 1998).

Whenever an NP is generated that refers to a museum exhibit that ILEX would be able to describe, there is the potential to create a hypertext link anchored by that NP. In fact, links are only created in the block of forward pointers and for references in the text to objects previously seen. Creating hypertext links for references to the current object would be redundant: the user can get more information by clicking on a 'say more' button, as seen in figures 2 and 3.

The realisation mechanisms in ILEX have been designed to be domain independent, and simple enough that no extensions (except for new lexis) are needed for new domains (*limited authoring*). They seem to provide sufficient expressive power to generate reasonable quality text in the domains ILEX has been applied to.

6 Summary

We started by setting out the special requirements for dynamic hypertext systems, and on ILEX in particular. We conclude by summarising the consequent architectural choices, and the extent to which the requirements have been met.

Requirements imposed or required by NLG

History Awareness – This is basically made possible by the context model, which includes the interaction history, user knowledge (assimilation values), local

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

and global focus. This is used to achieve effects such as appropriate referring expressions, comparisons between items, limited repetition of material and explicit acknowledgements of repeated material.

Requirements imposed by Hypertext

Speed – Speed is enhanced by: having the generation of the content potential performed at compile time; the simple content determination procedure (involving numerical manipulations for stereotyped user types with changing assimilation); the limited grammar; and the selective ability to use canned text. ILEX typically responds to a request for a description in 6–8 seconds.

Limited Planning and Prioritisation – These are handled mainly by the opportunistic content determination procedure. This is efficient but assumes reliable numerical annotations for guidance.

Link Awareness – Addressed in the generation of NPs for forward links and in decisions about what to exclude in content determination.

Further practical requirements

Limited Output – This is also addressed by the content determination procedure and also by the ability of the text structuring algorithm to work flexibly within length limits. The actual length in screen inches cannot be guaranteed exactly by these methods, but they seem to do well enough in practice.

Limited Authoring – This is addressed by starting with a simple relational database model for domain information, orienting the defeasible rule format to language generation rather than inference and allowing canned text to eliminate complex knowledge representation. Although there are only limited tools for authoring ILEX domains, the domain knowledge required by the system is relatively intuitive and is being further simplified (O'Donnell *et. al.* 2000).

The solutions to the different requirements are surprisingly interwoven in the architecture. For instance, the mechanisms for history awareness allow the system to ameliorate the effects of limited planning by ensuring coherence at least with respect to the past. Link awareness helps to reduce the problems of limited output (by limiting the content considered) and also seeds opportunities which can then be exploited by history awareness (referring back to the object just seen).

Acknowledgements

The ILEX project was supported by EPSRC grant GR/K53321 and was carried out in collaboration with the National Museums of Scotland, with the support of Interactive Information Ltd. and VIS Interactive Media Ltd. We thank various other researchers who have contributed significantly to the project, including Janet Hitzeman (anaphora, speech); Hua Cheng (aggregation); Maria Milosavljevic (comparisons); Caroline Sporleder (focus control), Adriana Morales (Spanish language generation), Scott Nowson (personnel domain), and Richard Cox (evaluation).

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

References

Andre, E., Finkler, W., Graf, W., Rist, T., Schauder, A. and Wahlster, W. (1993) WIP: The automatic synthesis of multimodal presentations. In: M. Maybury (ed.), *Intelligent Multimedia Interfaces*, pp. 75–93. Menlo Park, CA, AAAI Press.

Bateman, J. A. and Teich, E. (1995) Selective information presentation in an integrated publication system: an application of genre-driven text generation. *Information Processing & Management* **31**(5): 753–768.

Brusilovsky, P. (1996) Methods and techniques of adaptive hypermedia. *User Modelling & User Adapted Interaction* **6**: 87–129.

Brusilovsky, P., Stock, O. and Strapparava, C. (eds.) (2000) *Adaptive Hypermedia and Adaptive Web-Based Systems*. Berlin, Springer-Verlag.

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

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

Cheng, H. (1998) Embedding new information into referring expressions. In *Proceedings of COLING-ACL'98 (Student Session)*, pp. 1478–1480. Montreal, Canada.

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

Cox, R., O'Donnell, M. and Oberlander, J. (1999) Dynamic versus static hypermedia in museum education: an evaluation of ILEX, the intelligent labelling explorer. *Proc. Artificial Intelligence in Education Conference (AI-ED99)*, pp. 181–188. Le Mans, France.

Dale, R., Green, S., Milosavljevic, M. Paris C., Verspoor, C. and Williams S. (1998) The realities of generating natural language from databases. *Proc. 11th Australian Joint Conference on Artificial Intelligence*, Brisbane, Australia.

Dale, R., Oberlander, J., Milosavljevic, M. and Knott, A. (1998) Integrating natural language generation with hypertext to produce dynamic documents. *Interacting with Computers* **11**: 109–135.

Geldof, S. (1998) Con-textual navigation support. *New Review of Hypermedia and Multimedia* **4**: 47–65.

Grosz, B. J., Joshi, A. K. and Weinstein, S. (1995) Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics* **21**: 202–225.

Halliday, M. A. K. (1994) *Introduction to Functional Grammar*. 2nd ed. Edward Arnold, London.

Hitzeman, J., Black, A., Taylor, P., Mellish, C. and Oberlander, J. (1998) On the use of automatically generated discourse-level information in a concept-to-speech synthesis system. In *Proc. 5th International Conference on Speech and Language Processing*, pp. 2763–2766. Sydney, Australia.

Knott, A., O'Donnell, M., Oberlander, J. and Mellish, C. (1997) Defeasible rules in content selection and text structuring. *Proc. 6th European Workshop on Natural Language Generation*, Duisburg, Germany.

Knott, A., Oberlander, J., O'Donnell, M. and Mellish, C. (to appear) Beyond elaboration: The interaction of relations and focus in coherent text. In: T. Sanders, J. Schilperoord and W. Spooren (eds.), *Text Representation: Linguistic and Psycholinguistic Aspects*. Benjamins, Amsterdam. (In press.)

Kobsa, A., Muller, D. and Nill, A. (1994) KN-AHS: An adaptive hypertext client of the user modeling system BGP-MS. *Proc. 4th International Conference on User Modelling*, pp. 99–105.

Levine, J., Cawsey, A., Mellish, C., Poynter, L., Reiter, E., Tyson, P. and Walker, J. (1991) IDAS: Combining hypertext and natural language generation. *Proc. 3rd European Workshop on Natural Language Generation*, Judenstein, Austria.

Mann, W. C. and Thompson, S. (1988) Rhetorical Structure Theory: A theory of text organization. *Text* **8**: 243–281.

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press

Marcu, D. (1997) From local to global coherence: A bottom-up approach to text planning. *Proceedings of AAAI-97*.
Martin, J. R. (1992) *English Text: System and Structure*. Benjamins, Amsterdam.
McKeown, K. R. (1985) *Text Generation*. Cambridge University Press.
McKeown, K., Elhadad, M., Fukumoto, Y., Lim, J., Lombardi, C., Robin, J. and Smadja, F. (1990) Natural language generation in COMET. In R. Dale, C. Mellish and M. Zock (eds.), *Current Research in Natural Language Generation*, pp. 103–139. Academic Press, London.
Milosavljevic, M. (1999) *Maximising the Coherence of Descriptions via Comparison*. PhD Thesis, Macquarie University, Sydney, Australia.
Milosavljevic, M. and Oberlander, J. (1998) Dynamic electronic catalogues: helping users to help themselves. *Proc. ACM Hypertext 98*, pp. 123–131. Pittsburgh, PA.
Moore, J. D. (1995) *Participating in Explanatory Dialogues*. MIT Press.
Moore, J. D. and Paris, C. L. (1993) Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics* **19**: 651–694.
Nielsen, J. (1995) *Multimedia and Hypertext: The Internet and Beyond*. Academic Press, London.
Not, E., Petrelli, D., Sarini, M., Stock, O., Strapparava, C. and Zancanaro, M. (1998) Hypernavigation in the physical space: Adapting presentations to the user and to the situational context. *New Review of Hypermedia and Multimedia* **4**: 33–45.
O'Donnell, M. (1996) Input specification in the WAG sentence generation system. *Proc. 8th International Workshop on Natural Language Generation*. Herstmonceux Castle, UK.
O'Donnell, M., Cheng, H. and Hitzeman, J. (1998) Integrating referring and informing in NP planning. *Proc. Coling-ACL '98 Workshop on the Computational Treatment of Nominals*, pp. 46–55. Montreal, Canada.
O'Donnell, M., Knott, A., Oberlander, J. and Mellish, C. (2000) Optimising text quality in generation from relational databases. *Proc. 1st International Natural Language Generation Conference*, pp. 133–140. Mitzpe Ramon, Israel.
Pollard, C. and Sag, I. A. (1994) *Head-Driven Phrase Structure Grammar*. Chicago, University of Chicago Press and Stanford, CSLI Publications.
Reiter, E., Mellish, C. and Levine, J. (1992) Automatic generation of on–line documentation in the IDAS project. *Proc. 3rd Conference on Applied Natural Language Processing*, pp. 64–71. Trento, Italy.
Reiter, E., Robertson, R. and Osman, L. (1999) Types of knowledge required to personalise smoking cessation letters. In: W. Horn *et al.* (eds.) *Artificial Intelligence in Medicine: Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making (AIMDM'99)*, pp. 389–399. Springer-Verlag, Berlin.
Rino, L. H. M. and Scott, D. R. (1996) A discourse model for gist preservation. In: D. L. Borges and C. A. A. Kaestner (eds.), *Advances in Artificial Intelligence (Proceedings of the 13th Brazilian Symposium on Artificial Intelligence)*, pp. 131–140. Springer-Verlag, Berlin.
Taylor, P., Black, A. and Caley, R. (1998) The architecture of the Festival Speech Synthesis System. *Proc. 3rd ESCA Workshop on Speech Synthesis*, pp. 147–151. Jenolan Caves, Australia.
Wahlster, W. (ed.) (2000) *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer-Verlag, Berlin.
Wanner, L. and Hovy, E. (1996) The HealthDoc sentence planner. In *Proceedings of the Eighth International Natural Language Generation Workshop*, pp. 1–10. Herstmonceux Castle, Sussex, UK.

https://doi.org/10.1017/S1351324901002698 Published online by Cambridge University Press