

# Demonstration of ILEX 3.0

Michael O'Donnell†(micko@dai.ed.ac.uk),  
Alistair Knott‡(alick@hermes.otago.ac.nz),  
Jon Oberlander†(jon@cogsci.ed.ac.uk),  
Chris Mellish†(chrism@dai.ed.ac.uk)

† Division of Informatics, University of Edinburgh.  
‡ Department of Computer Science, Otago University.

## Abstract

We will demonstrate the ILEX system, a system which dynamically generates descriptions of database objects for the web, adapting the description to the discourse context and user type. Among other improvements in version 3, the system now generates from *relational databases*, and this demonstration will focus on this ability. We will also show how incremental extensions to the domain semantics improve the quality of the text produced.

## 1 Introduction

ILEX is a tool for *dynamic browsing* of database-defined information: it allows a user to browse through the information in a database using hypertext. ILEX generates descriptions of a database object on the fly, taking into account the user's context of browsing. For more information on ILEX, see Knott et al. (1997) and Mellish et al. (1998).

The demonstration will consist of generating a series of texts, in each case adding in additional components of the domain semantics. This short paper should be read in conjunction with the full paper elsewhere in this volume.

## 2 Generating from Bare Data

We start initially with a relational database, as defined by a set of tab-delimited database files, plus some minimal semantics. As discussed in the paper, we use assume a relational database to consist of two types of files:

1. *Entity Files*: each of which provides data for a particular entity type. Each row (or record) defines the attributes of a different entity. See figure 1.
2. *Link Files*: where a particular attribute may have multiple fillers, we use link files to define the entity-entity relations. See figure 2.

To generate from these files, the domain-editor needs to provide two additional resources:

1. *Data-type specification* for each entity-file, a specification of what data-type the values in the

| Entity | Material |
|--------|----------|
| J-997  | silver   |
| J-997  | enamel   |
| J-997  | gold     |

Figure 2: A Sample from a Link file

column are, e.g., string, entity-id, domain type, etc.

2. *Domain Taxonomy*: detailing the taxonomic organisation of the various classes of the entities.
3. *Mapping Domain taxonomy onto Upper Model*: ILEX uses an Upper Model (a domain-independent semantic taxonomy, see Bateman (1990)), which supports the grammatical expression of entities, e.g., selection of pronoun, differentiation between mass and count entities, between things and qualities, etc. We require that the basic types in the domain taxonomy are mapped onto the upper model, to allow the entities to be grammaticalised and lexicalised appropriately.

With just this semantics, we can generate texts, although impoverished texts, such as:

*The class of J-997 is necklace. It's designer is Jessie M. King. It's date is 1905.*  
...

Several tricks are needed to generate without a specified domain semantics:

- *Use of standard clause templates*: lacking any knowledge of how different attributes are to be expressed, the system can only generate each attribute using a standard template structures, such as *the X of Y is Z* or *It's X is Z*. The attribute names, e.g., *Designer*, *Style*, etc. can be assumed to work as the lexical head of the Subject. This ploy sometimes goes wrong, but in general works. (this approach borrowed from Dale et al. (1998)).

| ID    | Class    | Designer | Date | Style       | Place  | Sponsor   |
|-------|----------|----------|------|-------------|--------|-----------|
| J-997 | brooch   | King01   | 1905 | Art-Deco    | London | Liberty01 |
| J-998 | necklace | King01   | 1906 | Art-Deco    | London |           |
| J-999 | necklace | Chanel01 | 1910 | Art-Nouveau | Paris  |           |
| etc.  |          |          |      |             |        |           |

Figure 1: A Sample from an Entity file

- *Referring to Entities*: there are a number of strategies open for referring to entities. If the *Name* attribute is supplied (a defined attribute within the ILEX system), then the system can use this for referring. Lacking a name, it is possible for the system to form nominal references using the *Class* attribute of the entity (all entities in ILEX databases are required to have this attribute provided). We could thus generate indefinite references such as *a brooch* as first mentions, and on subsequent mentions, generate forms such as *the brooch* or *the brooch whose designer is Jessie M. King*. Without specification of which entities should be considered part of the general knowledge of the reader, we must assume all entities are initially unknown.
- *Fact Annotations*: ILEX was designed to work with various extra information known about facts, such as the assumed level of *interest* to the current reader model, the *importance* of the fact to the system’s educational agenda, and the assumed *assimilation* of the information (how well does the system believe the reader to already understand it). See the main paper for more details.

Lacking this information, the system assumes an average value for interest and importance, and a 0 value for assimilation (totally unknown).

With only default values, the system cannot customise the text to the particular user. It may provide information already well known by the user, and thus risking boring them. Also, there can be no *selection* of information to ensure that the more interesting and important information is provided on earlier pages (the reader may not bother to look at later pages).

Other information (defeasible rules), which allows us to organise the material into complex rhetorical structure, is also missing.

So, these tricks allow us to generate simple texts, consisting of a list of template-formatted clauses.

### 3 Adding Expression information

In the next step, we will add in information about how the various attributes should be expressed. This includes three main resources:

1. *Syntactic expression of attributes*: for each attribute, we provide a specification of how the attribute should be expressed syntactically.
2. *Lexicalisation of domain types*: by providing a lexicon, which maps domain types to lexical items, we avoid problems of using the domain type itself as the spelling. The lexical information allows correct generation of inflectional forms (e.g., of the plural for nouns, comparative or superlative forms for adjectives).
3. *Restrictive modifiers for referring expressions*: In choosing restrictive modifiers for forming referring expressions, some facts work better than others. For instance, *the brooch designed by King* is more likely to refer adequately than *the brooch which was 3 inches long*. ILEX allows the user to state the preferential order for choosing restrictive modifiers.

The addition of these resources will result in improved expression within the clauses, but not affect the text structure itself, which are still a list of clauses in random order.

### 4 Adding User Annotations

In the next step, we add in the user model, which provides, for each attribute type, predicted user interest, importance for the system, and expected user assimilation.

Using these values, ILEX can start to organise the text, placing important/interesting information on earlier pages, and avoiding information already known by the user.

### 5 Adding Defeasible Rules, Stories

As a final step, we add in various resources which improve the texture of the text.

- *Defeasible Rules*: ILEX allows the assertion of generalisations like *most Art Deco jewels use enamel*. These rules allow the generation of complex rhetorical structures which include Generalisation, Exemplification and Concession. The use of these relations improves the quality of the text generated.
- *Stories*: much of the information obtainable about the domain is in natural language. Often, the information is specific to a particular

entity, and as such, it would be a waste of time to reduce the information into ILEX's Pred-Arg knowledge structure, just to regenerate the text. Because of this, ILEX allows the association of canned text with a database entity (e.g., J-999), or type of entity (e.g., jewels designed for Liberty). The text can then be included in the text when the entity or type of entity is mentioned.

The intermixing of generated and canned text improves the quality of generated texts by providing more variety of structures, and allowing anecdotes, which would be difficult to model in terms of the knowledge representation system.

## 6 Conclusion

By showing incremental addition of domain specification within the ILEX system, we have demonstrated that it is a system which can function with varying degrees of information. This allows domain developers to rapidly prototype a working system, after which they can concentrate on improving the quality of text in the directions they favour.

## References

- John Bateman. 1990. Upper modeling: organizing knowledge for natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Generation*, Pittsburgh, June.
- Robert Dale, Stephen J Green, Maria Milosavljevic, Cécile Paris, Cornelia Verspoor, and Sandra Williams. 1998. The realities of generating natural language from databases. In *Proceedings of the 11th Australian Joint Conference on Artificial Intelligence*, Brisbane, Australia, 13-17 July.
- Alistair Knott, Michael O'Donnell, Jon Oberlander, and Chris Mellish. 1997. Defeasible rules in content selection and text structuring. In *Proceedings of the 6th European Workshop on Natural Language Generation*, Gerhard-Mercator University, Duisburg, Germany, March 24 - 26.
- Chris Mellish, Mick O'Donnell, Jon Oberlander, and Alistair Knott. 1998. An architecture for opportunistic text generation. In *Proceedings of the Ninth International Workshop on Natural Language Generation*, Niagara-on-the-Lake, Ontario, Canada.