WYSIWYM: knowledge editing with natural language feedback

Richard Power and Donia Scott
Information Technology Research Institute
University of Brighton
Brighton BN2 4GJ
email: {firstname.lastname}@itri.bton.ac.uk

Introduction

WYSIWYM (What You See Is What You Meant) is a user interface technique which allows an author to create and edit in a natural and simple way the knowledge contained in a generated document. More generally, WYSIWYM editing provides a self-documenting, multi-lingual approach to maintaining knowledge bases.

We demonstrate here the use of WYSIWYM knowledge editing in the DRAFTER-II system. DRAFTER-II is an interactive software tool designed to assist the production of technical documents in several languages at once. The prototype system allows a technical author or domain expert to create software manual instructions in English, French and Italian. Our interactive 'Symbolic Authoring' approach avoids the difficulty and cost of translation between languages and also the problems of attempting fully automatic generation of documents.

Symbolic Authoring

Symbolic Authoring is a document authoring method in which the author generates language-neutral 'symbolic' representations of the content of a document, from which documents in each target language are generated automatically, using NLG technology. A number of commercial or near commercial systems now exist (see Scott, Power and Evans (1998) for an overview). Many of these systems take their input from some external data source. The basic idea of Symbolic Authoring is to allow the user to specify the generator input directly.

To achieve this, it is clear that a key requirement of a Symbolic Authoring system is an effective user interface — one which enables the author to construct the knowledge base without assuming prior expertise in knowledge representation languages or in computational linguistics. This is a significant problem, which different systems have addressed in different ways. The WYSIWYM technique proposes a new solution to this problem.

WYSIWYM editing

WYSIWYM is a technique for creating and maintaining complex data objects such as typically found in knowledge bases, but presenting them to the author or knowledge editor as natural language texts. Clearly, a completely specified object in a knowledge base can be described in natural language by a suitable NLG system. But during the creation of a knowledge base object, some parts of it are incomplete — i.e., unspecified (or at least underspecified). If the NLG system is extended to be able to describe such incomplete objects, and also the range of options available to make an object more complete, the entire knowledge editing process can be presented to the user in terms of natural language.

This is the basic idea of WYSIWYM editing: a special kind of natural language text is generated in order to present the current (possibly incomplete) state of a data object in the knowledge base. This text includes generic phrases (or 'anchors') which mark attributes that have no value. The anchors serve as the locations where new objects may be added. By opening a pop-up menu on an anchor, the user obtains a list of short phrases describing the types of objects that are permissible values of the attribute; selecting one of the options, a new object of the specified type is added to the semantic network. A new text is then generated to present the modified configuration, including the attributes of the new object.

As more information is added about a new object, it will be represented by longer spans of text, comprising whole sentences, or perhaps even several paragraphs. The generator automatically reorganises the text, making new sentences, paragraphs, even subsections, maintaining anaphoric cross-references etc. to present even very complex and incomplete data structures as readable natural language texts. During knowledge editing with WYSIWYM , the user appears to be creating text, but she is doing this only indirectly by creating the underlying knowledge base. Whereas WYSIWYG editors (e.g., Microsoft Word, FRAMEMAKER and INTERLEAF) present the user with text as it will appear on the printed page, WYSIWYM editors present a text that reflects only what the user meant. In symbolic authoring applications, the final generator may choose to present the output differently (adopting different text styles, for example).

In the accompanying demonstration, we will show how a domain specialist or technical writer could use WYSIWYM to author a multilingual instructional manual in the domain of software products. Detailed examples of the process of authoring instructional texts with WYSIWYM are also provided in Scott, Power and Evans (1998), Power and Scott (1998), and Power, Scott and Evans (1998).

Architecture

Figure 1 shows the basic architecture of a WYSIWYM editing system, including the following features:

- The only thing presented to the user is a text generated from the current domain model.
- The user can choose between input (feedback) and output modality; the former presents the current state of the knowledge base and the latter the recommended text(s) (given a potentially complete knowledge base) for inclusion in the final document being created.
- The only way in which the user can edit the domain model is by selecting from pop-up menus on an feedback text.

The text is completely regenerated every time the user changes the domain model or switches the modality. So far we have developed two experimental systems with this architecture. In DRAFTER-II, which we will demonstrate here, the domain model and the generator are implemented in Prolog, while the interface is implemented in CLIM (CLIM 1994). In the other system, PILLS[1], the Prolog generator produces HTML source files which can be read by a web browser. In both applications, texts several paragraphs long can be generated very quickly, so that whenever the model is changed the text seems to be updated instantaneously.

[1] PILLS generates pharmaceutical insert leaflets in English, Dutch, French, German, Italian and Portuguese

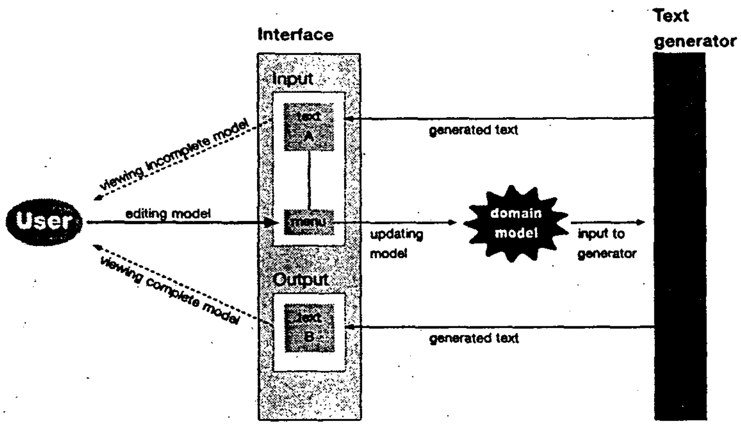

Figure 1: The WYSIWYM architecture

Switching from one language to another results in the immediate re-generation of the currently viewed text in the new chosen language. This takes no longer than the generation of a new text when the model is expanded during editing.

Conclusion

A key motivation for adopting a Symbolic Authoring approach to document production and management is multilinguality. In conjunction with WYSIWYM, however, this aspect takes on a new significance: in WYSIWYM, the language generator is used to drive the user interface. This means that interface localisation is no longer a problem, since the interface is automatically available in any language for which there is a NL generator available for producing the output text. It also facilitates the collaborative authoring of knowledge bases and their documentation – even multilingually. For example, the original author may construct a document in English, later modifications may be made in Italian, a French manager may want to review the work in her own language, etc.

We have not yet produced a formal evaluation of WYSIWYM , but our experience so far is very encouraging. Many visitors to our laboratory have used WYSIWYM in one or more of our prototype systems to author multilingual documents. Those who are familiar with using a mouse and menus are able to use WYSIWYM efficiently within two to three minutes.

References

Power, R., and Scott, D. (1998). What You See Is What You Meant: direct knowledge editing with natural language Feedback. *Proceedings of the 13th Biennial European Conference on Artificial Intelligence*, (ECAI'98). Brighton, UK.

Power, R., Scott, D. and Evans, R. (1998). Multilingual authoring using feedback texts. *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics* (COLING/ACL'98). Montreal, Canada.

Scott, D., Power, R. and Evans, R. (1998). Generation as a Solution to Its Own Problem. *Proceedings of the 9th International Natural Language Generation Workshop*, (INLG'98). Niagara-on-the-Lake, Canada. August, 1998.