Natural Language Generation in the Mile System

Paul Piwek, Roger Evans, Lynne Cahill and Neil Tipper

ITRI – University of Brighton
Watts Building, Moulsecoomb, Brighton BN2 4GJ, UK
email: Firstname.Lastname@itri.brighton.ac.uk

Abstract. We describe how Natural Language Generation (NLG) Technology is used in the Mile system. Mile is a web-based system for accessing maritime rules and regulations. We explain how the architecture of the system was derived from a set of user requirements and focus on the role of NLG in this architecture. More specifically, we describe how multilingual generation of answers to queries and the use of the NLG-based WYSIWYM-technology for multilingual query formulation fit into the architecture. The architecture is different from conventional dialogue systems in that it is centered around a dialogue database which allows the user to store, retrieve and manipulate so-called dialogue histories (i.e., language-independent records of the dialogue between the user and the system).

1 Introduction

As part of the Clime (Computerised Legal Information Management and Explanation) project[1], a natural language interface has been developed at ITRI in the University of Brighton for accessing legal and regulatory information via the internet. In particular, an application (Mile; Maritime Information and Legal Explanation) has been built for accessing maritime regulatory information.

In this paper, we describe the place of Natural Language Generation (NLG) Technology in the architecture of the Mile system. We aim to demonstrate how the NLG technology has been put to use in order to satisfy the user requirements of the application. In particular, our goal is to explain how the user requirements led to a dialogue database-oriented approach which makes maximal use of the NLG technology.

The remainder of this paper consists of three sections. In Section 2, we discuss the user requirements and describe how they have been addressed in the Mile system. Section 3 consists of the current state of the application prototype which has been developed in the first two years of the Clime project. We also discuss the planned efforts for the last year of the project which are aimed at further

[1] Clime is funded by the EC Esprit Programme under project number EP 25.414. The partners of the Clime project are British Maritime Technology Ltd., Bureau Veritas, TXT Ingegneria, the University of Amsterdam and the University of Brighton.

improvements of the system on the basis of feedback from the end-users. Finally, Section 4 contains our conclusions.

2 Satisfying the User Requirements

Before addressing the individual requirements for the application, let us characterize its context of use. The intended end-users of the application are surveyors who inspect sea-going vessels on their sea-worthiness. For this purpose, they use a large body of maritime regulations, which are currently available to them on paper and CD-ROM (with simple text retrieval). The MILE system is intended to change this situation. It allows a surveyor to specify the situation on a ship by means of a natural language text. The system is then able to retrieve the rules which are pertinent to the specified situation[2] and present this information by means of a natural language text which includes relevant images (of ships, ship parts, etc.) and HTML-links. Amongst other things, the links provide access to explanatory texts on how the system arrived at its answers.

For instance, a surveyor might want to know which rules apply to the situation described by the following text:

> "An oiltanker is fitted with three bilgepumps. One of them is out of order and another of them is used for firefighting. What are the rules which apply to this situation?"

The idea is that the user can enter this query and that subsequently the system can retrieve the rules which apply to the situation and present them to the user.

The task of a surveyor and the context in which s/he works give rise to a number of more specific requirements on an application of the sort we just described. Within the CLIME project, we formulated a set of such requirements in cooperation with representatives of the industrial partners of the CLIME project. These partners are two organizations which employ surveyors and other professionals who use maritime regulations: Bureau Veritas (one of the largest classification societies with over 100000 clients distributed over 150 countries) and British Maritime Technology, Ltd. (one of the world's leading maritime and engineering consultancies). We will now list the requirements which were thus gathered and describe how these requirements are addressed by the MILE system.

(1) *The user should be able to formulate (semantically) relatively complex queries pertaining to the situation of a ship.*

For instance, in the text given above we encounter phenomena such as plurality ("three") and anaphora ("one of them"). Unfortunately, for the purpose of practical applications, natural language understanding is not yet sufficiently

[2] For technical details on the retrieval/legal reasoning functionality of the system see Winkels et al. (1998).

reliable to allow a user to enter such texts freely by means of the keyboard or speech (see, e.g., Dix et al., 1998). Therefore, an alternative approach has been explored which allows the user to construct such queries by directly performing editing operations on the semantic representation underlying the query. The approach is called WYSIWYM, for *What You See Is What You Meant* (Power et al., 1998).

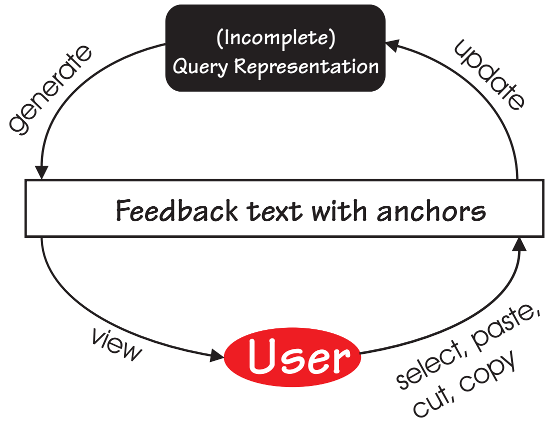

Fig. 1. The editing cycle

The idea, see Figure 1, is simple: a natural language text is generated from a yet to be completed semantic representation of a query. The text contains clickable anchors with pop-up menus. A menu presents the possible extensions of a query representation. On the basis of the extension that the user selects, the representation is updated and a new text is generated on the basis of the updated representation. Additionally, spans of text corresponding to underlying semantic objects can also be selected by means of the mouse. Cut and copy operations are available which allow the user to cut or copy the underlying semantic object into a buffer. Subsequently, such an object can be pasted into a location where the representation is still incomplete.

Consider, for instance a situation in which the following text represents the status of the query: "An oiltanker is fitted with three bilgepumps. **Some equipment** is out of order. **Some states**." Here, bold face indicates where the query is still incomplete. The user can select the span "three bilgepumps", and copy the underlying object (or a subset of it) into the anchor **Some equipment**. Subsequently, the text "An oiltanker is fitted with three bilgepumps. They are out of order. **Some states**." is generated. The underlying representation on which the copy and paste operations take place are object-oriented semantic networks (e.g., Sowa, 1984) which are closely related to Discourse Representation Structures (DRSs; Kamp & Reyle, 1993).

The MILE system uses simple representations which are equivalent to DRSs without logical connectives (such as implication). It does allow for the representation of coreference (Van Deemter & Power, 1998), Plurality (Piwek, to appear) and Speech Act Type information (Piwek et al., 1999).

In summary, the NLG-based WYSIWYM technology has been put to use for the formulation of queries. In this respect, this is a new application of the technology which was originally developed for multilingual document authoring and applied to several domains such as the authoring of software manuals in the DRAFTER II system (e.g., Scott et al., 1998) and more recently the authoring of Patient Information Leaflets in the ICONOCLAST project.[3] ICONOCLAST enables users to formulate logically complex texts. In this respect, there is a difference with MILE. This difference is motivated by the consideration that although the MILE end-users will make frequent use of the technology, the formulation of queries is from their perspective a subsidiary task. On the other hand, for a user of ICONOCLAST the editing of knowledge is the primary task. For such a type of user, the effort of learning how to construct logically complex information is therefore justified. For the average MILE user, this is less evident. We mention this point to draw attention to the tension between the theoretical possibilities of a technology and application specific considerations which can influence which aspects of a technology are made available to end-users.

(2) *The system should be accessible from anywhere in the world.*

This requirement arises out of the working environment of surveyors. Typically, they perform their task by visiting ships, whether it be at a ship yard, in a harbour or at sea. This requirement has given rise to a *web-based multi-agent distributed architecture*, where the interface can be downloaded on the user's computer as a JAVA APPLET which runs in a conventional web browser, whereas the natural language engines (which are written in PROLOG), the Dialogue Manager and the Legal Information Server (written partly in JAVA)[4] can run on high performance (windows NT) machines elsewhere. The latter modules can handle multiple users. In other words, they can handle communications with more than one user interface module.[5] An overview of the different modules and their organization is given in Figure 2.

This figure contains a screen dump of a special module (intended primarily for demonstration purposes), the "behind module", which provides the user with feedback on the system activity. It highlights the modules which are currently activated (in this case the query and response interface –that is, the user interface module– on the left at the bottom).

[3] See http://www.itri.brighton.ac.uk/research.html#ICONOCLAST

[4] The legal information server matches the situation which the user specified against the body of regulations

[5] Because the main functionality of the system is run by modules which can communicate with multiple user interfaces and the former modules can share user databases, it becomes also possible for users to share (e.g., look at) each other's query databases.

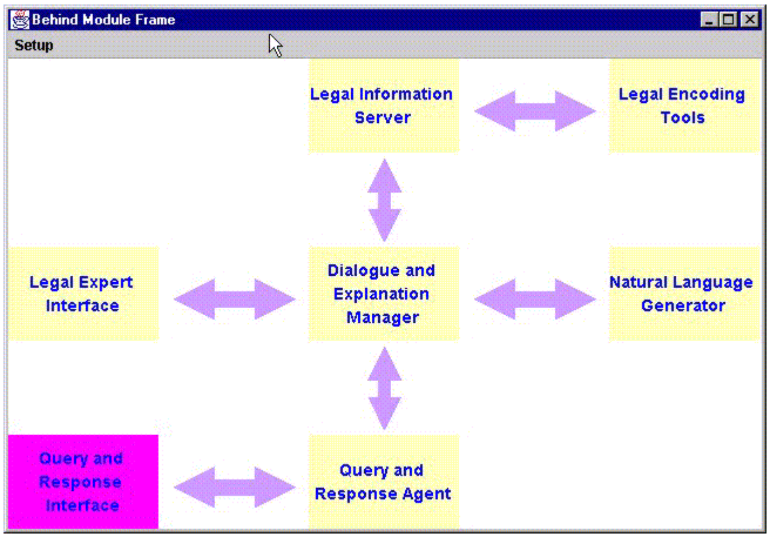

Fig. 2. The system status and architecture as it is conveyed to the user by the behind module

The relation of the Legal Expert Interface (which allows the user to direct a query to a human expert in case the system fails to provide an answer) and the Query and Response Interface (which allows the user to browse a database with queries and answers and construct, submit and manipulate queries) with the Dialogue and Explanation Manager (which manages the database of queries and answers and can generate explanatory information) and the Query Response Agent (which implements the WYSIWYM technology) is a client-server one. Communications on the server side of the architecture (the Dialogue and Explanation Manager, the Query and Response Agent, the Legal Information Server and the Natural Language Generator) are based on the CORBA (Common Object Request Broker Architecture) standard.

The Legal Encoding Tools module has a somewhat different role in the architecture. It is not part of the end-user system. There is a second version of the system for developers which includes this module. The module has been developed at the University of Amsterdam and provides an environment for encoding domain knowledge. Furthermore, in the future some tools which have been developed at the University of Brighton for semi-automatically generating linguistic resources from the domain knowledge will be integrated into this module

(3) *When the system is computing an answer to the user's query, the user should be able to direct his or her attention to other tasks (including the formulation of further queries) and be able to modify and resubmit queries which were posed earlier.*

These considerations have led to a *database-oriented dialogue model* analogous to conventional email systems. Such an architecture allows for asynchronous communication between the user and the system, e.g., the user can formulate and submit new queries before s/he has received the answers to previous queries. A simplified representation of the system architecture is depicted in Figure 3., where the arrows 1. and 6. involve the NLG technology.

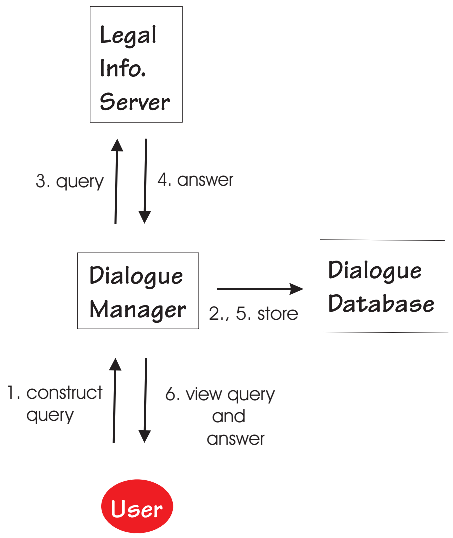

Fig. 3. The system architecture

The idea is that 1. the user constructs a query using WYSIWYM. 2. This query is stored in the Dialogue Database. More specifically, both the natural language text (in fact, several texts: one for each of languages which the system supports) and the formal representation of the query are stored in different fields of one and the same query record. This record carries a unique identifier. 3. The query (representation) is submitted to the Legal Information Server. 4. The Legal Information Server returns an answer in the form of a set of rules and a set of concepts which are pertinent to the users query and a set of properties of and relations between rules and concepts. 5. This information is stored in the Dialogue Database together with natural language texts for the answer which are produced by the NLG on the basis of the answer representation of the Legal Information Server. 6. The user is notified that the Dialogue Database has been updated and can now view the text of the question and its answer (in the language which is appropriate for her or him).

Let us now discuss the processing of a query from the user's perspective. After the user has logged in, a window with two frames appears, containing the browser interface to the MILE system, see Figure 4. The frame on the lefthand contains the applet which controls the interface. It includes a choice panel which displays a list of queries (and, if available, their answers) which the user has constructed on previous occassions. The user can (re)name these queries, and if required organize them in folders. On the righthand side, there is a view panel which displays the text of the query/answer which the user has selected in the choice panel.

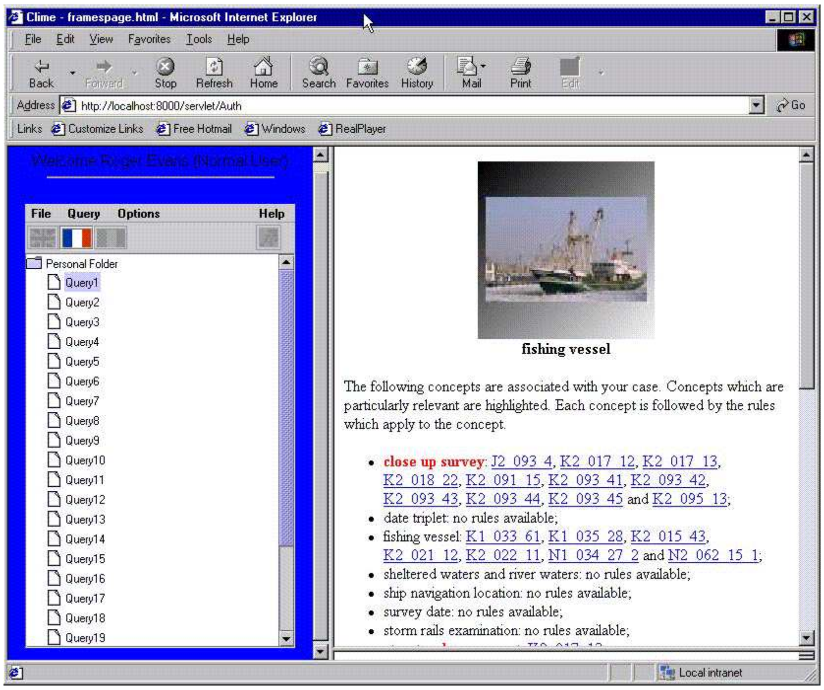

Fig. 4. The Main User Interface

In order to construct a new query, the user selects "Query" and then the option "new". This causes a query-editing window to pop up. In this window, the user can then formulate his or her query using the WYSIWYM technology. See Figure 5. for a query window with a WYSIWYM-constructed query.[6] Alternatively, the user can also access old queries which are stored in the dialogue database, alter them, and then resubmit them.

---

[6] For a walk through of the WYSIWYM construction process see, for instance, Scott et al. (1998) and Piwek et al. (1999). The former concerns the construction of software manuals, whereas the latter describes the process of query construction in the domain of MILE.

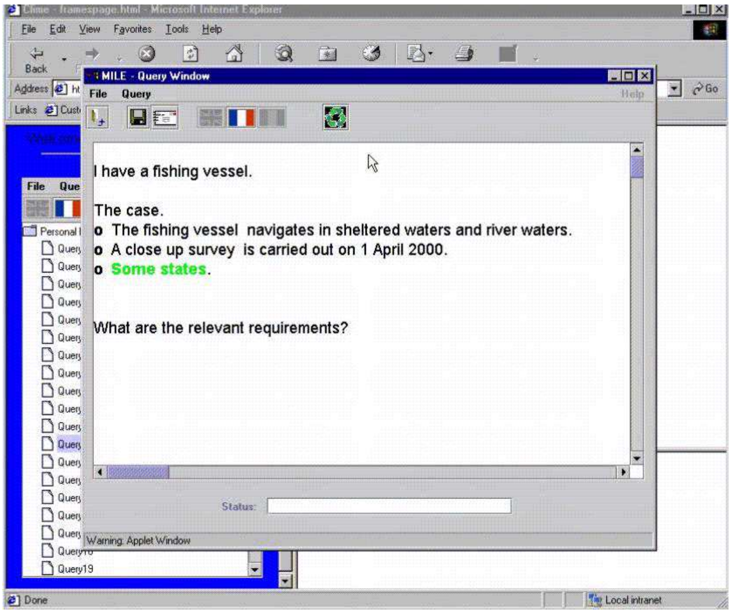

Fig. 5. Query Construction with English Feedback

(4) *The system is used in a world-wide operating company, which means that it should be adaptable to the language of the local users.*

Currently, MILE supports English and French (the lexica cover 300 domain specific concepts).[7] See Figure 6. for the French text of the query which is also depicted in Figure 5. The system uses separate generators (using a pipe-line architecture; cf. Reiter & Dale, 1997) for query formulation and answer generation, although these generators do share the lexical resources. The query formulation generator is based on a unification grammar which allows for the mixing of proper grammar rules and rules for fixed phrases. The input for the generator is the semantic network which the user constructs using the WYSIWYM technology. For the (also multilingual) answer generation, a less complex generator is used.[8] This generator is tailored to quick generation of HTML documents on the basis of the output of the Legal Information Server. A data format has been developed which is particularly suited for generation in legal domains, where the answer consists of a set of rules marked up with explanatory and background information. Bacically, this format is specified as a set of sets: a set of rules, a set of concepts and a set of properties of/relations between rules and concepts.

[7] An Italian Grammar is under construction.

[8] The choice for this generator was driven by practical considerations. Specifically, the general purpose generator which we use is less suited for incorporation of open ended concept classes, such as the set of rule names.

Rules show up in the answer text as HTML links. Clicking on these rules evokes a helper servlet which retrieves the text of the rule and depicts it in a separate window.

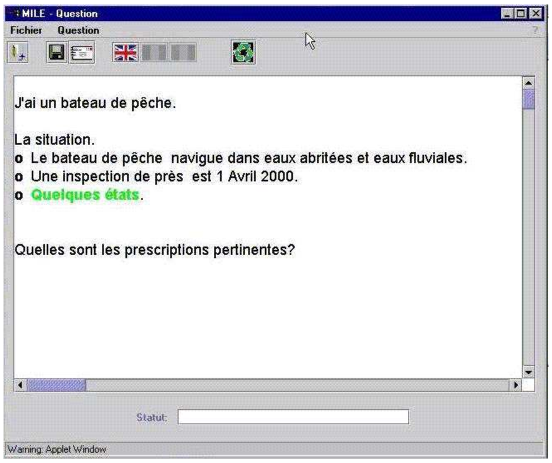

Fig. 6. Query Construction with French Feedback

3 Current State of the Project and Future Developments

Currently (April 2000) the project has just entered its third and last year. Version 3 of the system was demonstrated in November 1999 at the EC Information Society Technologies (IST) Exhibition in Helsinki. Subsequently, the system was distributed amongst the CLIME partners. Feedback from the end-user partners has led to a number of changes to the system. For instance, the layout of the answer texts has been changed. At the same time, we developed a more general representation of the data format which is passed from the Legal Information Server to the NLG. This format has been designed to make generation of different layout as easy as possible. This was achieved by keeping the data structures as "flat" as possible. The idea behind this is that extracting a piece of information from complex recursive structures requires more programming effort and can lead to less robust software.

Furthermore, the development of a demonstrator for a new domain (environmental regulations) has been initiated and should provide us with more information on the reusability of the software.

4 Conclusions

To conclude, let us summarize how we tried to address the two central questions of the workshop in this paper. Firstly, *I have invented a new technique for* NLG*!*

– *What is its impact on applications?* We have indicated how a new technology based on NLG, i.e., WYSIWYM, makes it possible for users to formulate queries of a complexity (e.g., anaphora and plurals) which is not achievable with current technology for free text interpretation. As for the question *I have built a new* NLG *application! – What is its impact on the technology?*, we have tried to show that the demands of the application suggests a new context for the use of NLG, i.e., as part of an application for asynchronous multilingual human-system interaction. Finally, our experiences of building an NLG module which has to interface with other modules (which were developed in parallel) has led us to the adaptation of flat datastructures which allow for less complicated information extraction for NLG.

Acknowledgements Thanks are due to Alessandra Bagnato for kindly letting us use a screen dump of the behind module (Fig. 2) which was implemented by her at TXT Ingegneria. The research reported in this paper was carried out as part of the CLIME project. CLIME is funded by the EC Esprit Programme under project number EP 25.414. The partners of the CLIME project are British Maritime Technology Ltd., Bureau Veritas, TXT Ingegneria, the University of Amsterdam and the University of Brighton.

References

Dix, A., J. Finlay, G. Abowd, R. Beale (1998), *Human-Computer Interaction*, Prentice-Hall, Hertfordshire.

Kamp, H. & U. Reyle (1993), 'From Discourse to Logic', Kluwer Academic Publishers, Dordrecht.

Piwek, P., R. Evans and R. Power (1999), 'Editing Speech Acts: A Practical Approach to Human-Machine Dialogues', *Proceedings of AMSTELOGUE '99: Workshop on the Semantics and Pragmatics of Dialogue*. University of Amsterdam.

Piwek, P. (to appear), 'A Formal Semantics for Generating and Editing Plurals', *Proceedings of COLING 2000*, Saarbruecken.

Power, R., D. Scott and R. Evans (1998), 'What You See Is What You Meant: direct knowledge editing with natural language feedback', *Proceedings of ECAI-98*, Brighton, UK, 1998, 180–197.

Reiter, E. & R. Dale (1997), 'Building Applied Natural Language Generation Systems', Natural Language Engineering 3 (1): 57-87.

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

Sowa, J. (1984) Conceptual Structures, Addison Wesly, Reading, Massachusetts.

Van Deemter, K. and R. Power (1998), 'Coreference in knowledge editing', *Proceedings of the COLING-ACL workshop on Computational Treatment of Nominals*, Montreal, Canada, 1998, 56–60.

Winkels, R., A. Boer, J. Breuker & D. Bosscher (1998), 'Assessment Based Legal Information Serving and Co-operative Dialogue in CLIME'*Proceedings of JURIX-98*, GNI, Nijmegen, Netherlands 131–146.