Requirements Definition, Validation, Verification and Evaluation of the CLIME Interface & Language Processing Technology

Paul Piwek

ITRI – University of Brighton
Watts Building, Moulsecoomb, Brighton BN2 4GJ, UK
Email: Paul.Piwek@itri.brighton.ac.uk

Contents

1 Introduction

This document provides an overview of the requirements analysis, validation/verification and evaluation activities which were carried out during the construction of the MILE system. MILE stands for Maritime Information and Legal Explanation. The MILE system is an application of the generic CLIME architecture, where CLIME stands for Computerized Legal Information Management and Explanation. Both CLIME and MILE were developed as part of the CLIME project. CLIME was funded by the European Commission ESPRIT programme, project number EP 25.414. The partners in the project were British Maritime Technology Ltd., Bureau Veritas, TxT E-Solutions SPA, the University of Amsterdam and the Information Technology Research Institute (ITRI) of the University of Brighton.[1] The project ran for three years (1998 - 2001).

The focus of this paper is on those aspects of the development of MILE system which involved the natural language processing components and the user interface technology. Both of these were conceived and implemented at the ITRI in Brighton.

---

[1] At ITRI the CLIME team was led by Roger Evans and consisted of Lynne Cahill, Paul Piwek and Neil Tipper.

The paper proceeds as follows. In Section 2 we briefly discuss the architecture of the MILE system. Section 3 describes how the user requirements for the system were arrived at. Next, Section 4 describes how validation and verification were addressed during the development of the system. Section 5 describes the efforts which were made to evaluate the system. Finally, in Section 6 we list some conclusions which we derived from our experiences in developing the MILE system.

2   The MILE **system**

The MILE system (see Piwek et al., 2000) was built during the course of the CLIME project. This project started in February 1998 and ran for three years until February 2001. During this period several prototypes of the MILE system were built. The architecture of the MILE system was developed during the first year of the project. This architecture is shown in Figure 1.
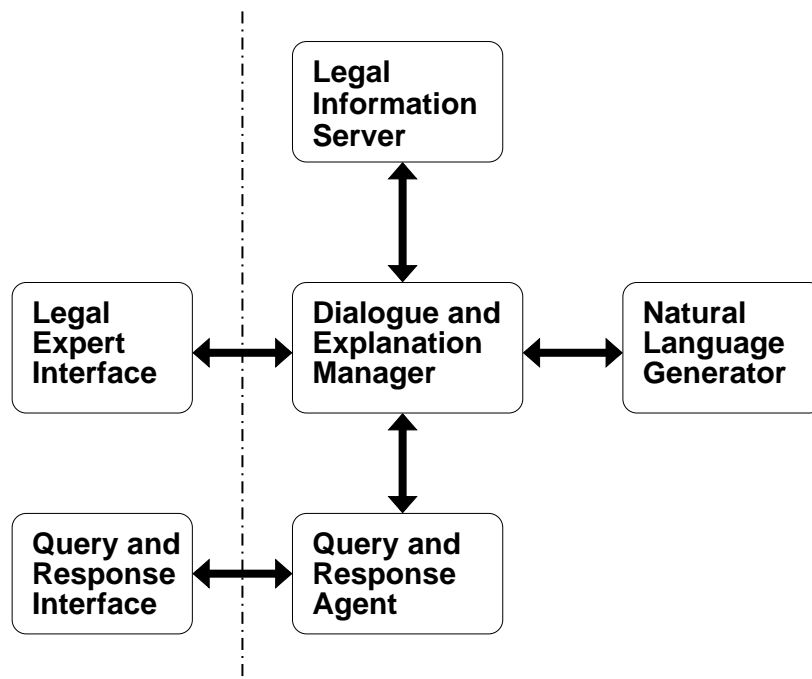

Figure 1: The CLIME architecture

The modules in the left-hand column are web-delivered JAVA applets. The other modules make up the CLIME server. This diagram does not show the additional off-line modules of the CLIME architecture, notably the **Legal Encoding Tools** for the encoding of regulations as formal rules.

The **Query and Response Interface** (QRI) is a JAVA applet for query construction (using WYSIWYM, see Figure 2) and query management (browsing, filing, submitting) (See Figure 3). As an applet, it is relatively lightweight, and relies on a server-side module, the **Query and Response Agent** (QRA) for the heavier processing, notably WYSIWYM natural language feedback generation (Power et al., 1998). Thus the QRI is really just a client-side presentation manager for the QRA. The **Dialogue and Explanation Manager** (DEM) provides the persistent database storage of queries and answers, manages the interactions with the user, and between the server modules, and (somewhat surprisingly, for historical reasons) provides explanation functionality. The **Legal Information Server** (LIS) is the engine that actually provides answers to questions, by reference to its knowledge base of formally
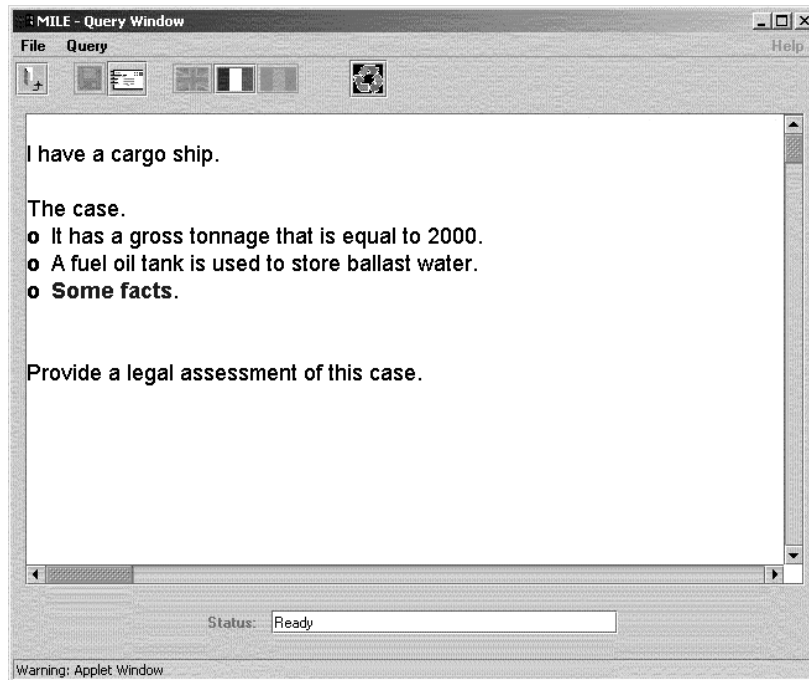
Figure 2: The CLIME query interface

encoded legal regulations. The **Natural Language Generator** (NLG) is responsible for rendering the answers in natural language, potentially including explanations in a readable form. Finally the **Legal Expert Interface** (LEI) is a secondary web-based interface to the system, allowing a legal expert to manually browse and insert answers into the system database if the system is unable to provide the answer itself.

The overall operational scenario of the system is as follows: the user manipulates the interface provided by the QRI, supported by the QRA, to construct a query. When the user submits the query, it is passed to the DEM, which stores it in its persistent database, and also passes it to the LIS for legal processing. The LIS returns a response to the DEM, which processes the response to maximise relevance and incorporate explanatory material, and then passes it to the NLG. The NLG generates text, which it returns to the DEM. The DEM then notifies the QRA, and hence the QRI, that an answer is available, and the user accesses it whenever they wish. If the LIS is unable to deliver an answer, the query is emailed to a human expert. This expert then connects to the system using the LEI to insert a response into the database manually. The DEM's database is used to store all the intermediate results involved in the processing of a query: the query as WYSIWYM data, as LIS input data and as text, the LIS response and the NLG output texts (HTML).

In this document, the focus will be on the functionality and interfaces for the following modules: the QRI, the QRA and the NLG.

3   User Requirements

In this section we discuss the user requirements for the MILE system; in particular, the requirements pertaining to the QRI, QRA and NLG. The user requirements were based on three different sources of information:
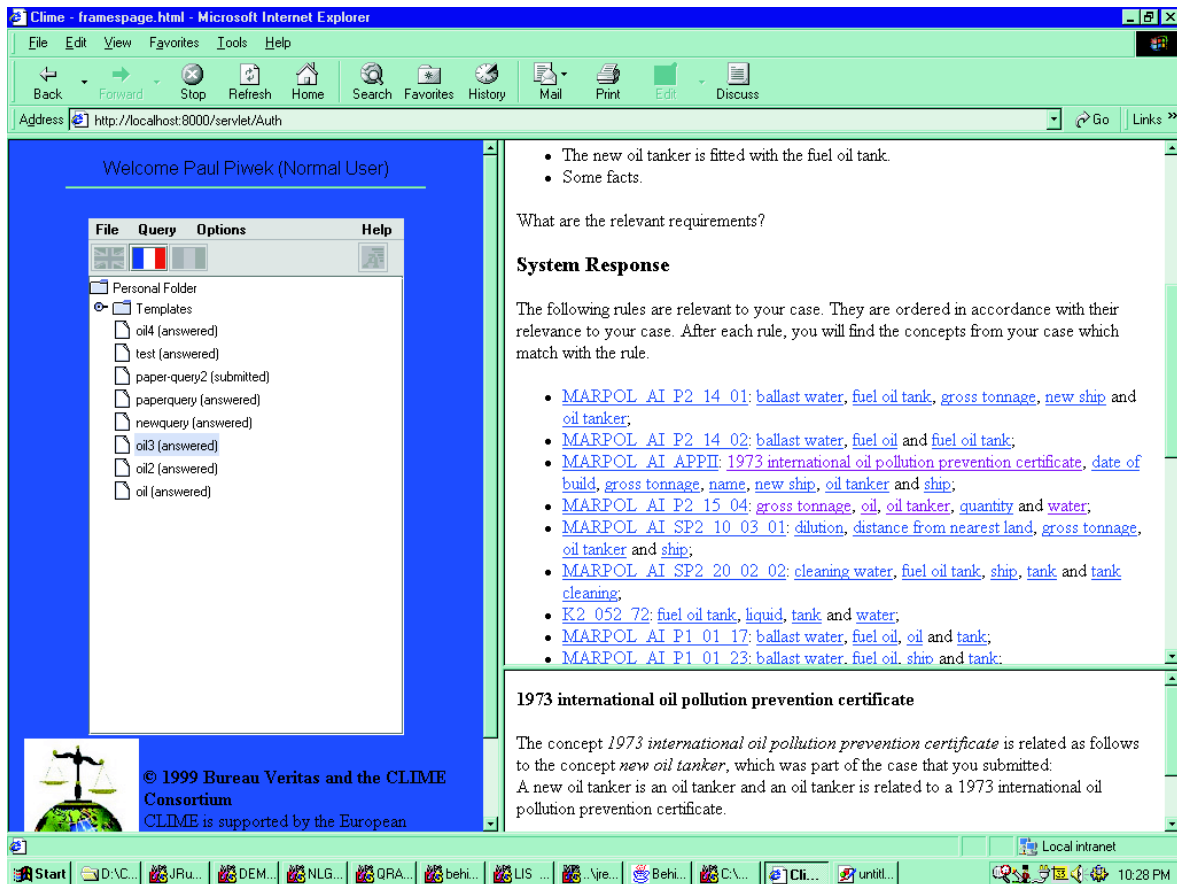
Figure 3: The CLIME user interface running in a conventional web browser

1. An analysis of the situation in which the system should operate;

2. Information on the IT skills of the end users obtained by means of questionnaires;

3. A corpus of queries posed to human experts by the potential end-users and the answers of the experts to the queries.

The situation of use

The main type of 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. Further potential users who might also benefit from the MILE system are, for example, ship owners and shipyards.

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

---

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

4

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).

Requirements for the MILE domain

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

2. Given that the application is in the legal domain, the system should operate with a high degree of accuracy.

3. The system should be accessible from anywhere in the world.

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

5. The system will be deployed in a company which employs both English and French speakers. Therefore, the system should be adaptable to both French and English speakers.

6. System responses should contain, or make available on demand, enough explanatory text to ensure the user correctly understands and has confidence in the answer given.

7. Given the ever changing world of maritime regulations, mechanisms should be in place for the maintenance of the system.

IT background of the end-users

In order to get a better idea of the background of the potential end-users and their current practice with respect to consulting the maritime rules we asked our industrial partner Bureau Veritas to fill in a questionnaire on these topics. The questionnaires were filled in by an engineer involved in updating the rules and an engineer from the equipment department. Both indicated that they consult the rules several times a week, both in electronic and paper form. Both extensively worked within a PC environment (MS windows). Furthermore, both were experienced in using the internet and using applications such as the MS Word word processor. This rather small sample was justified by the fact that Bureau Veritas' Business Processes are highly standardized and therefore employees in similar departments organize their work along the same lines.

The Corpus of Question/Answer Pairs

The aim of the MILE system is to partly relieve the experts on the Bureau Veritas rules (the people who author and update the rules) from their task of answering enquiries about the rules by various interested parties (Surveryors, Owners, Shipyards, etc.). For that purpose, we needed to get an understanding of the enquiries which the experts are faced with. On the one hand, our industrial partners provided us with a rough classification of different types of queries which are currently being asked (Clime, 1998):

- Closed type query: A query about an existing vessel, accompanied by a precise proposal (for changes). The query is whether the proposal is acceptable or not.

- Open-type queries: The existing vessel is fully known, the request is to know all the requirements concerning a given item, exhaustively.

- General queries: The idea is to obtain all information about a specific item.

In addition we gathered a corpus of about 120 queries. For that purpose we provided our partners at Bureau Veritas with a (MS-WORD) query template which they were asked to fill in for individual queries. An example of a filled in query template can be found in Table 1.

Number	2
Date	04/99
Question	Are the emergency towing arrangements applicable to oil tankers?
Answer	It depends. In fact, the question leads to another question: what is the deadweight of the oiltanker? In case where the deadweight equal or greater than 20000t, the answer is "yes". In case where the deadweight is less than 20000t, the answer is "no". (Reference: 8-041-16)
Follow up Question	
Answer to follow up	
Type of Client	Owner, shipyard, BV surveyor
Frequency	Low
Comments	

Table 1: Filled in Query Template

4 Validation and Verification

Verification is the task of checking whether the system conforms to its specifications. Validation involves ascertaining that the system fulfills the expectations of the software procurer.

The verification task was driven primarily by reviews which took place every six months for the duration of the project. These reviews were led by a EU project officer and involved three independent reviewers (two from industry and one academic). During the initial six months of the project the bulk of the requirements were collected. However, during the same time the first prototype of the query interface was developed and a mock-up of the full system (in JAVA Script) was developed. These were demonstrated during the first review meeting. In subsequent review meetings, successive versions of the fully integrated system were demonstrated.

During the last review meeting on the 6th of March 2001 a demonstration of the final system was given. On the basis of these meeting each of the three reviewers wrote a final review report. These reports were then summarized by the project officer as follows:

'The project concludes with the un-altered good co-operation spirit combined with very high professional skills that have been prevailing in the consortium since the start. The achieved results are in great deal in line with the expectations described in the original

project programme. The encoding effort had been under-evaluated, but this has been compensated to a certain degree by the exploration and demonstration of a new domain. The project thus terminates successfully.'

With regards to the WYSIWYM technology the following remarks were made:

'The work on the natural language interface has resulted in the embedding of the WYSIWYM technology in a complex environment. The architecture of the application supports domain and language portability for which some interest has already been raised in the railway sector.'

Validation (as opposed to verification) was effected through (1) meetings with the industrial partners in which new prototypes were demonstrated and feedback was provided and (2) through releases of the system which were distributed to all the partners in the consortium and which led to feedback which was communicated in subsequent meetings.

From the aforementioned meetings we obtained feedback pertaining to various aspects of the system which was under development such as:[3]

- Proposals for changes to the **terminology** used in the interface. For instance, a suggestion was made to replace the term 'incomplete' with 'unprocessed' for queries which the user is editing and which have not yet been submitted to the system. Various comments were also received regarding the language used in the query interface. E.g., the term 'state' (short for 'state of affairs') was replaced by the term 'fact'.

- Suggestions concerning **where to present information on the interface**. For instance, in the initial version of the query interface, non-essential information about the user and the system (name, user type) was presented on the main browser window. The end-users suggested that this information could be ommited in order to leave more room for displaying essential information concerning the query which appeared in the window. Furthermore, the end users requested that certain frequently used options be made accessible through buttons on a toolbar.

- Comments to the effect that **certain functionality is not yet operational**. In earlier prototypes options such as changing the font size were not yet implemented. Although we notified our partners of many of these limitations (and our intention to implement them in a later release), we nevertheless often got comments with respect to them.

- Comments **identifying bugs in the current release** of the system. E.g., in an early release of the system the language of the system would only change after a delay and the undo button could cause the system to crash under certain conditions.

- Comments with respect to **the installation of the system**. Initially, installation of new releases of the system required manual modifications of a large number of files.

- Comments pertaining to **the speed at which the system performs tasks**. Early prototypes were rather slow, which made it difficult to test the functionality of the system in a reasonable amount of time.

---

[3]We also benefitted from comments which a group of students provided in the context of an assignment of a Usuability Evalation module taught at the University of Brighton (Masthoff, 2000).

- Comments on the **window management of the system**. In an early version of the system the query window kept being hidden from view by the main window.

- **Concept navigation** became problematic when the system covered 3000+ concepts. Initially, the only method of navigation was through menus. We were forced to find a more efficient method for supporting search in a long list of items. The use of a selection box was implemented to address this problem.

- Remarks regarding the **feedback and error reporting** supported by the system. E.g., the status of the system was not alway transparent to the user (Has the current query been processed? Has the system got stuck?).

5   Evaluation

During the last year of the project various evaluation activities took place. Again, we concentrate on the evaluation of the natural language processing and interface technology. We discuss evaluation under two headings: analytical evaluation and empirical evaluation.

The subsection on analytical evaluation describes to what extent and how the seven requirements which are listed in section 3 are supported by the technology which was developed in the CLIME project. Additionally, we analyse the MILE system in terms of the TRINDI ticklist (Bohlin et al., 1999).

The subsection on empirical evaluation discusses the results of the evaluation of the natural language processing and interface technology by the other project partners, in particular, TxT E-solutions, which performed a preliminary evaluation, and Bureau Veritas who filled in a usuability questionnaire which we supplied to them.

Analytical Evaluation

Technological solutions for the Requirements   In this paragraph, we discuss how the seven requirements which were listed in section 3 are addressed by the MILE system. We present the technological solutions to the problems which these requirements posed.

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

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 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 reliable to allow a user to enter such complex content 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).
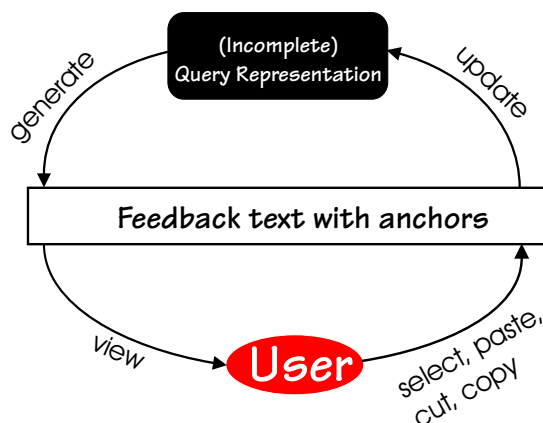

Figure 4: The editing cycle

The idea, see Figure 4, 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, 2000) 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.[4] 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

---

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

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) *Given that the application is in the legal domain, the system should operate with a high degree of accuracy.*

The WYSIWYM technology which is used to allow users to formulate queries is particularly suited for this purpose. It gives users a very high degree of control over the interpretation which the computer constructs of their query. The system does not try to parse and/or interpret the user's query (which currently still often leads to incorrect interpretations in conventional systems).[5] Rather the user is provided with a direct manipulation interface which allows her or him to directly build up her or his query.

(3) *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 LISP and partly JAVA) are running on high performance (windows NT) machines elsewhere. An overview of the different modules and their organization is given in Figure 1.

(4) *When the system is computing the answer to the user's query, the user should be able to direct his/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 5, where the arrow 6. involves the NLG technology.

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 texts for the answer which are produced by the NLG on

---

[5]Rayner et al. (2000) argue that for spoken language translation only systems with limited domains (e.g., travel information, hotel booking services) are feasible for the foreseeable future. Their estimate is that only for such domains it will eventually be possible (though still challenging) to construct systems with a coverage of 85 to 90% (their Spoken Language Translator has an accuracy of 75% for English to Swedish and 65.3% for English to French in the Air Travel Information Domain).
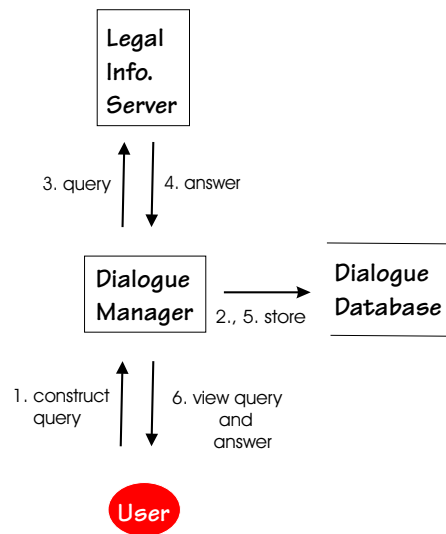
Figure 5: Simplified system architecture illustrating the life of a query

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.

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 3. 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. 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.

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 2 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.

(5) *The system will be deployed in a company which employs both English and French speakers. Therefore, the system should be adaptable to both French and English speakers.*

Currently, MILE supports English and French (the lexica cover 3000+ domain specific concepts). See Figure 6 for the QRI in French mode. 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 answer generation, a less complex generator is used. 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

---

[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.

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. Basically, 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.
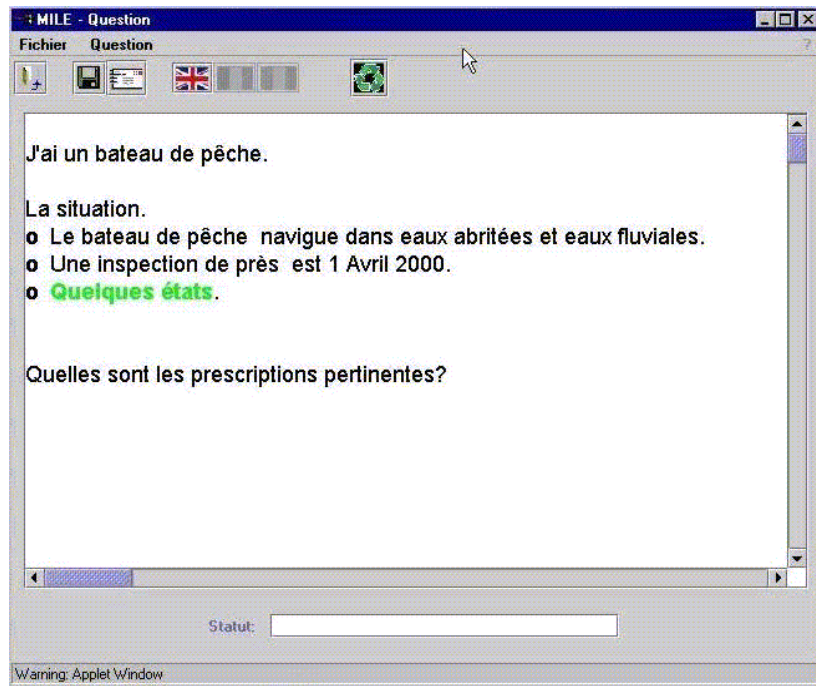

Figure 6: Query Construction with French Feedback

(6) *System responses should contain, or make available on demand, enough explanatory text to ensure the user correctly understands and has confidence in the answer given.*

The MILE system allows for two layers of interaction. Firstly, there is a layer where the user interacts with the system very much like he/she would with a conventional email system: constructing and submitting queries without waiting for an answer to be returned immediately. The second layer involves the interaction between the user and the answer to a query. This answer is presented in the form of a dynamically generated HTML document. This document has links to other dynamically generated documents containing elaborations on the answer. See Figure 3: the links in the upper left pane (e.g., "1973 international oil pollution prevention cerficate") are clickable and displayed in the lower left pane. At this level, the interaction is synchronous: the user can navigate within hypertext answer documents in real time. Figure 7 shows the text of an answer to a case assesment query, i.e., a query which consists of the description of a situation on a ship (the case) and the question "Is this situation allowed or not?". Not only the answer is displayed (one of "allowed", "disallowed" and "silent"), but also the rule instantiation on the basis of which this conclusion was drawn. Furthermore, further rule instantiations are given which apply to situations which are slightly different from the one which the user described. For each of these, it is indicated what the verdict would have been in these alternative situations.

(7) *Given the ever changing world of maritime regulations, mechanisms should be in place for the maintenance of the system.*

The MILE system inferred on the basis of the norms in regulations 13, 14 and 16 of MARPOL that your case is **disallowed**.

This conclusion is drawn because:

MARPOL-AI-P2-14-01: The fuel oil tank is part of the new oil tanker, the ballast water is located in the fuel oil tank and the gross tonnage of the new oil tanker is equal to or more than 150.

The following rules, if applicable, change the verdict concerning your case. For each rule, the circumstances under which that rule changes the verdict concerning your case are indicated:

MARPOL-AI-P2-16-03: If the new oil tanker proceeds on route, an oily water separating equipment is part of the new oil tanker, oily mixture is discharged, the location is not part of a special area, the distance from nearest land of the location is more than 12km and the oil content of the oily mixture is less than 100, then the case is allowed.

MARPOL-AI-P2-13-04: If the new oil tanker is a segregated ballast tanker, a segregated ballast tank is part of the new oil tanker and the ship length of the new oil tanker is less than 150, then the verdict depends on the judgement of the enforcer of the regulations.

MARPOL-AI-P2-13-04: If the new oil tanker is a segregated ballast tanker, a segregated ballast tank is part of the new oil tanker and the ship length of the new oil tanker is equal to or more than 150, then the case is allowed..

Figure 7: An automatically generated case assessment answer

A system architecture was chosen with a centralized MILE server website. This means that maintenance can also be centralized and updates to the resources are directly available to all users. A set of Legal Encoding Tools including tools for updating the natural language processing resources were developed. Basically, they allow for semi-automatic updates of the natural language processing resources whenever the legal rule base and ontology are extended.

The TRINDI tick-list   In addition to the analysis of the MILE system in terms of the domain specific requirements, in this section we investigate whether a more generic analytical evaluation is possible. The TRINDI tick-list (Bohlin et al., 1999) provides a starting point. This is a list of desired dialogue behaviours formulated in terms of yes/no-questions. Unfortunately, it is tailored specifically to spoken dialogue systems and puts a strong emphasis on situations where the systems asks questions and the user provides answers.[7] Despite all these reservations, we think that it is beneficial to examine to what extent the TRINDI tick-list does apply to the MILE system.

   In total, the TRINDI tick-list consists of 12 questions. Question 8[8] is specifically meant for spoken

---

[7]Travel information dialogues belong to that category of man-machine dialogues: the system asks the user questions about her or his destination, preferred arrival time, etc. in order to look up a suitable itinerary.

[8]Can the system deal with noisy input?

dialogue systems. The questions 2^9, 3^{10}, 4^{11}, 7^{12}, 11^{13} apply to systems which ask questions to the user and expect the user to answer.

Although the MILE systems does not ask questions in the ordinary sense, it does indirectly do so through the WYSIWYM technology. Every time the user makes a choice on a menu, this can be seen as an answer to an implicit question. For instance, suppose we are at a stage where the following feedback text is available: I have **some ship**. Here bold face indicates an anchor. If the user clicks on this anchor, a menu with types of ships appear. Implicit in this menu is the question "What type of ship do you have?". Because the user is required to formulate his or her answer as a choice on a menu, answers which give more than the required information (2), unexpected information (3), less than the required information (4) or no information (7) can not occur.

It is more difficult to interpret the question whether the system only asks appropriate follow-up questions (11). Let us assume that the equivalent of a follow-up question in the WYSIWYM setting is a menu which appears after the user has already performed a number of selections on menus. Sometimes selections which the user made earlier on will constrain which items should appear on the current menu. For instance, suppose the user has created a subset of some set of earlier mentioned objects (see Piwek 2000 for a detailed example). Now we have the following feedback text: **Some number** of the 5 Bilgepumps is driven by the main engine. The menu associated with the anchor "**Some number**" should only display the following options: 1, 2, 3, 4 and 5. If the option 10 were also avaible we would arrive in a situation where the user is allowed construct a representation of: "10 of the 5 Bilgpumps is driven by the main engine". In other words, the follow-up question presented to the user (in the form of a menu) would allow the user to formulate an inconsistency. The WYSIWYM plurals systems does in fact make sure that the situation we sketched does not arise; it will only display the numbers 1 to 5 on the menu. However, more generally speaking the problem has not yet been addressed for WYSIWYM editing. Except for the just mentioned type of check, there is no general mechanism for checking whether the items on a menu might allow the user to formulate an inconsistency.

A related question is number 12^{14} of the TRINDI tick-list. Currently, the situation is such the natural language processing technology is not sensitive to whether the information which is provided to the system is consistent or not, it simply passes it on the Legal Information Server, which then has to deal with it. For information on the Legal Information Server we refer to Winkels et al. (1998).

Let us now move to question $1.^{15}$ Bohlin et al. (1999) discuss how the TRAINS (Allen, 1996), SRI AUTOROUTE (based on the Core Language Engine Technology. See Alshawi, 1992) and the PHILIPS TRAIN INFORMATION system (Aust et al., 1995) address this question. Like most of these systems the CLIME system is able to cope with the context sensitivity of user input and its potential ambiguity. However, in CLIME these issue are addressed in a different way. For instance, in the PHILIPS system ambiguity and context sensitivity of utterances are controllable in virtue of a very limited task (the system needs to find out at what time the user wants to travel from where to where). This allows the system to predict to a high degree what the user will say, and therefore allows the systems to interpret the user's utterance in the light of these predictions. It is, however, not clear how such a strategy is scaleble to more complex domains and tasks. CLIME on the other hand controls the user input

---

[9]Can the system deal with answers to questions that give more information than was requested?

[10]Can the system deal with answers to questions that give different information than was actually requested?

[11]Can the system deal with answers to questions that give less information than was actually requested?

[12]Can the system deal with no answer to a question at all?

[13]Does the system only ask appropriate follow-up questions?

[14]Can the system deal with inconsistent information?

[15]Is utterance interpretation sensitive to context?

by presenting directly to the user the operations which the system can understand. The operations allow, however, for the construction of relatively complex input involving sequences of (negated) propositions and coreference chains. In particular, when constructing a coreference chain, the user exploits contextual information. She or he can copy and paste earlier introduced objects into new propositions.

Questions 9[16] and 10[17] concern sub-dialogues initiated which are initiated by the user. The notion of a sub-dialogue has no direct equivalent in the MILE-system: the user formulates a query (using WYSIWYM), submits it, receives an answer and can if s/he wants ask follow-ups by clicking on dynamically generated hyperlinks in the answer.

Interestingly, none of the systems investigated in Bohlin et al. (1999) deals with negation (TRINDI question 6)[18] beyond utterances such 'No Bristol' in response to 'Do you want to go to Brighton?'. The MILE system, however, allows the user through WYSIWYM to formulate the negation of any proposition which can be formulated.

Empirical Evaluation

Evaluation by TxT E-Solutions TxT E-Solutions investigated version 4 of the MILE system. This system was demonstrated at the EU IST exhibition in Helsinki (22-24 November 1999). Their evaluation led to the following conclusions (taken from Bertin & Bagnato, 1999):

- The interface is particularly well-suited to users who are already familiar with windows environments (such as MS Windows), since it uses a subset of the facilities provided by common direct manipulation interfaces (Bertin & Bagnato, 1999:10).

- 'In our opinion also infrequent users can learn it [to use the interface] in an enough quick way, the WYSIWYM approach is intrinsically helpful in this regard, the user is always helped in his query creation'. (Bertin & Bagnato, 1999:10)

- Due to the use of the WYSIWYM technology, users are guided when constructing a query and cannot produce ill-formed queries. (Bertin & Bagnato, 1999:10)

- Users with no expertise in the maritime domain are able to formulate meaningful queries by means of the WYSIWYM interface. (Bertin & Bagnato, 1999:10)

- In addition to the possibility to copy and paste items in the WYSIWYM interface, it would be helpful also to carry out drag and drop operations. (Bertin & Bagnato, 1999:12)

- Selection of items in the WYSIWYM interface could be enhanced by adding the facility of a selection/type in box. (Bertin & Bagnato, 1999:11)

- The interface which was studied in Bertin & Bagnato (1999) did not support multi-threading. That is, once the user submitted a query to the system, s/he he could not proceed to carry out other tasks with the system. Rather s/he had to wait until the system returned with an answer.

- The availability of on-line help for users who are new to the interface is desirable. (Bertin & Bagnato, 1999:12)

---

[16]Can the system deal with 'help' sub-dialogues initiated by the user?
[17]Can the system deal with 'non-help' subdialogues initiated by the user?
[18]Can the system deal with negatively specified information?

The issues which are presented in the last three items were all dealt with in subsequent releases of the system: a type in/selection box was implemented for the navigation of long menus, multi-threading was supported and an on-line help manual and tutorial were delivered as part of the system.

Evaluation results obtained through Bureau Veritas Towards the beginning of the third project year, we distributed a HCI questionnaire (closely following the recommendations in Shneiderman, 1998) to our end-user partner (Bureau Veritas). This enquiry was filled in by the tester at the user partner. This tester had just joined the project (taking over from a previous tester) around that time and was therefore a novice to the system (but an expert in the maritime domain).

The questionnaire was divided into two parts: one on the browser window (see Figure 3) and one on the query window (see Figure 2). Each question or statement in the questionnaire was followed by a box numbered from 1 to 9 and marked for their dimension. Additionally a box with the option NA (Not Applicable) was available:

```
(2) Screen layouts were helpful

never                                 always        NA
   1    2   3   4   5   6   7   8   9
  [ ]  [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]           [ ]
```

The subject was told the following:

```
Please tick the item/number which most appropriately reflects
your impressions about using the browser window.  The browser
window allows you to navigate the query database, view queries
and answers and access the text of rules.
```

The results for the browser window can be found in Table 2 on page 19. The results for the query window are presented in Table 3 on page 20. Unfortunately, although these results do provide some information about the perception of naive users of the MILE system, the findings can not be used to draw any firm conclusions. Firstly, only one subject filled in the questionnaire. Secondly, the system which this person worked with was not the final version of the CLIME system. Rather it the version of the system after year 2 of the project. Despite all these hedges, let us try to summarize which issues this questionnaire brought to light.

We start with the browser window (table 2). Overall, the values obtained are positive (e.g. high scores for most overall reactions to the browser window, the screen layout, the ease to learn to use the browser window, the order in which the users are supposed to carry out tasks, etc.). There are, however, two dimensions on which the system is judged extremely negative. Firstly, the user claims to be never sure of the result which an operation leads to. This finding might be surprising, given that the user also claims that it is easy to use the system and that the steps to complete a task follow a logical sequence. However, this discrepancy can be explained from the fact that the system which the user was faced with still contained some serious bugs. E.g., sometimes the system would stall or crash. Since these bugs were repaired in a later version of the system, overall the evaluation by this user seems positive. Secondly, according to the user the time it takes to complete a task is too long. Again, this was mainly due to the fact that the release which the user evaluated was not yet optimized with respect to its performance. This issue was addressed in later prototypes.

Let us now turn to the evaluation of the query window. Once again we have a number of dimensions on which the user's evaluation turns out be positive and a number of dimensions on which it is negative. High marks are given for the clarity of the messages which appear on the screen, the ease with which a user can learn to operate the interface (included the advanced features), the number of steps it takes to complete a query and the logical organization of how to complete a task.

However, when it comes down to formulating queries for the specific MILE application (i.e., maritime regulations), the user experiences difficulties in formulating the queries he had in mind. Furthermore, he found the interface to be difficult, rigid, without adequate power and slow. The last of these complaints was addressed by optimizing the performance of the system. The difficulties which the user experienced with formulating queries were partly due to the fact the release which the user tested worked with an incomplete underlying ontology. This meant that many concepts which the user wanted to access were simply not available. Nevertheless, we think that it is likely that even with a more complete system, the WYSIWYM technology will to some extend require the user to rethink his or her query in terms of the query structures which the system allows him or her to build. In the end, there is a trade-off between the more rigid nature of the interface on the one hand (as opposed to free text input) and its ability to deal accurately with very complex queries. For the moment, we conclude that this is an issue which needs further study.

6   Conclusions

In this paper we examined the requirements analysis, validation/verification and evaluation activities which were undertaken during the CLIME project with regards to the natural language processing and interface technology.

We indicated how the requirements were based on an analysis of the situation in which the system is to be deployed supplemented with information on the IT skills of the prospective users and a corpus of queries (and answers).

Verification of the compliance of the system with the specifications was addressed through six-monthly review meetings led by the project officer from the EU. The system was judged positivily during these meetings and at the end of the project the officer concluded that the project was completed successfully. Validation that the system under construction lived up to the expectations of the end-users was ensured by regular feedback meetings between the system builders and the end-users and releases of the system to the end-users for testing purposes. Out of these interactions, many comments on the system were gathered which were used to improve it.

Finally, various evaluation activities were undertaken. Although it came to light that there is still scope for improving the use of the WYSIWYM technology used for query formulation, in general, the results were positive. In particular, the learnability, ease of use and transparency of the WYSIWYM technology were evaluated favourably.

7 References

Allen, J., B. Miller, E. Ringger & T. Sikorski (1996), 'Robust Understanding in a Dialogue System', *Proceedings of the 34th ACL*, University of California, Santa Cruz, 62–70.

Aust, H., M. Oerder, F. Seide and V. Steinbiss (1995), 'The Philips automatics train timetable information system', Speech Communication, 17, 249–262.

Bertin, A. & A. Bagnato (1999), 'Evaluation of the Natural Language Interface'. CLIME (EP 25414) Document. Task 3.3., 6 March 2000 (Circulation restricted to: CLIME Consortium).

Bohlin, P., J. Bos, S. Larsson, I. Lewin, C. Mathesin & D. Milward (1999), *Survey of existing interactive systems*, TRINDI deliverable D1.3, available at:
`http://www.ling.gu.se/research/project/trindi`.

Clime (1998), CLIME. *Description of the RTD Project*, 8 January, 1998.

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

Masthoff, J. (ed.) (2000), *Analytical Usability Evaluations of WYSIWYM*. Unpublished MS.

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.

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

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

Piwek, P., R. Evans, L. Cahill & N. Tipper (2000), 'Natural Language Generation in the MILE System', In: *Proceedings of the IMPACTS in NLG Workshop*, Schloss Dagstuhl, Germany, 33–42.

Rayner, M., D. Carter. P. Bouillon, V. Digalakis & M. Wirén (2000), *The Spoken Language Translator*, Cambridge, Cambridge University Press.

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.

Shneiderman, B. (1998), *Designing the User Interface*, Addison-Wesley, Reading, Massachusetts.

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.

Overall reactions to the browser window	NA	terrible - wonderful
	7	frustrating - satisfying
	7	difficult - easy
	7	inadequate power - adequate power
	5	rigid - flexible
Screen layouts of the browser window were helpful	8	never - always
Arrangement of information on query files and folders (as displayed in the left-hand side frame of the browser window) is	8	illogical - logical
Arrangement of information in answer texts to your queries is	[no answer]	illogical - logical
Answers which the system returns make sense given the questions you put to the system	4	no sense at all - ok
Answers which the system returns provide new information to you	5	new - well-known information
Answers which the system returns provide useful information to you	5	useful - no use
Terminology used for manipulating queries and folders is inconsistent	[no answer]	inconsistent - consistent
Terminology used in the text of answers to your queries is	[no answer]	inconsistent - consistent
Performing an operation leads to predictable results	1	never - always
Messages which appear on screen	8	confusing - clear
Learning to operate the browser window	8	difficult - easy
Learning advanced features	6	difficult - easy
Number of steps per task	8	too many - just right
Time it takes to complete a task	1	too long - just right
Steps to complete a task follow a logical sequence	8	never - always

Table 2: Answers of Evaluator concerning the Browser Window

Overall reactions to the query window	NA	terrible - wonderful
	5	frustrating - satisfying
	2	difficult - easy
	2	inadequate power - adequate power
	3	rigid - flexible
Screen layouts were helpful	7	never - always
Arrangement of queries on the screen	6	illogical - logical
Terminology used in the query window	5	inconsistent - consistent
Formulation of the queries you want to is	1	impossible - easy
Performing an operation leads to predictable results	5	never - always
Messages which appear on screen	8	confusing - clear
Learning to operate the query window	8	difficult - easy
Learning advanced features	8	difficult - easy
Number of steps for completing a query	8	too many - just right
Time it takes to complete a query	1	too long - just right
Steps to complete a task follow a logical sequence	7	never - always

Table 3: Answers of Evaluator concerning the Query Window