An Expert System for the Application of Import and Export Regulations

G. Van Nevel, F. Balfroid and R.Venken

BIM, Kwikstraat 4,
B-3078 Everberg
Belgium

ABSTRACT

This paper presents a Prolog based expert system for the application of regulations which is currently developed by BIM for the Belgian Ministry of Economic Affairs. The focus is set on the system architecture and on the development methodology.

1. INTRODUCTION

The current practice of the administrative staff of the Belgian Ministry of Economic Affairs for performing tasks concerning the import and export of sugar products consists of a series of forms to be filled in and some manual manipulations in compliance with about 500 pages of European regulations and other Belgian laws.

Because regulations are extensive, complex and frequently modified, it is difficult for the executive personnel to work correctly (i.e., as mentioned in the regulations). Also the one expert responsible for the sugar sector does not have the time to check all the delivered work for correctness and completeness. Moreover, the situation is complicated by the fact that there is a large turnover of administration personnel.

Today, the information system development at the Belgian Ministry has made considerable progress in automating the more clerical, computational and archival oriented parts of the daily work. Unfortunately, due to the complexity and the evolution of the European regulations and other procedures, the current systems are unable to tackle the decision process, which constitutes the heart of the administrative tasks to be performed.

Therefore, the Belgian Ministry of Economic Affairs asked for an expert system development as a first trial to support the decision procedures about importing and exporting sugar products and to improve the quality of the delivered work in terms of correctness and completeness.

In comparison with other expert systems developments, a first additional problem for this application is that the knowledge base must be easily adaptable to take regulations modifications into account. A second additional problem is related to the fact that the interface has to operate in the Belgian national languages used by the Ministry personnel: Dutch and French.

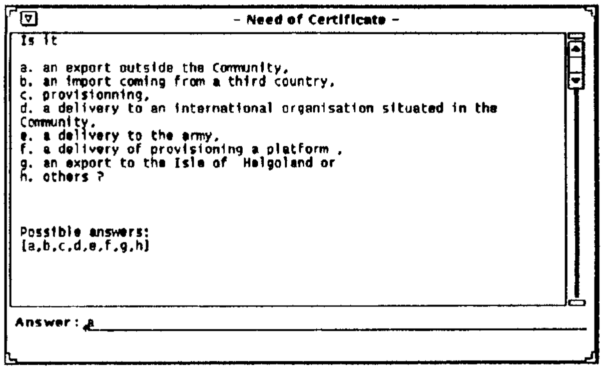
2. CURRENT STATE OF THE PROJECT

The expert system development amounts to 25 man months spread over a period of two years, starting on January 90 and ending on January 92.

During the first three months of the project, a prototype was developed in Prolog: an application of 40 pages of regulations. The corresponding knowledge base consists of 330 if-then rules, 170 Prolog clauses and 120 concepts. The main advantage of the domain dependent syntax of the if-then rules on Prolog clauses is to provide a way of representing knowledge in a more readable form for non-Prolog experts. This makes the knowledge base creation and maintenance easier. This last point is very important, because regulations are frequently modified. The concepts which appear in the conditional and conclusion part of the if-then rules (see paragraph 3.3 for more details) represent a segment or piece of regulation text.

The interface is window-oriented in compliance with the standard windowing interface, Open Look, as defined by SUN. In figure 1, we show the dialogue window which is used for the end-user interaction during a knowledge session.

FIGURE 1. The dialogue window



```
- Need of Certificate -
Is it

a. an export outside the Community,
b. an import coming from a third country,
c. provisionning,
d. a delivery to an international organisation situated in the
Community,
e. a delivery to the army,
f. a delivery of provisioning a platform ,
g. an export to the Isle of  Helgoland or
h. others ?


Possible answers:
[a,b,c,d,e,f,g,h]

Answer : a
```

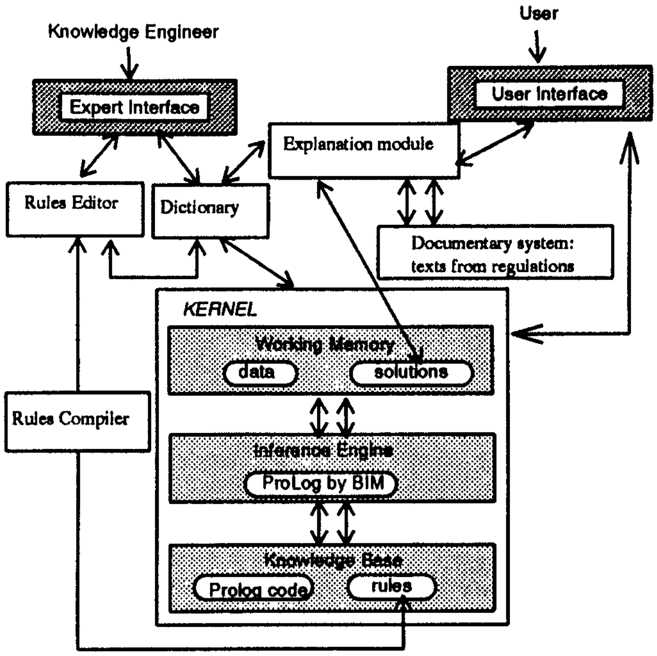
The experience acquired during the development of the prototype allowed us to work out a first specific methodology to build expert systems for regulations application. This methodology is presented below.

Thanks to demonstrations of the first prototype, elementary parts of the expert system specifically applied to known problems have been visualized such as: the knowledge base, the interface and the explanation facility. It was thus easier for the individual responsible for the project and for the target users of the system to formulate their demands and for the developers to redesign the architecture of the application accordingly.

3. EXPERT SYSTEM ARCHITECTURE

The architecture of the Import/Export expert system is presented in figure 2. We will briefly describe its main components.

FIGURE 2. Architecture of the expert system



3.1 THE KERNEL

The kernel of the expert system is divided into three main parts:

- the *knowledge base* contains the result of the compilation of the high level if-then rules (generated by the rules compiler from the external rules) and some Prolog code extending the set of built-in predicates that can be used in rules conditions (such as *append*, *member*, ...).
- the *working memory* stores data and solutions and could possibly be linked with an external database.
- the *inference engine* is an automaton that applies the regulation knowledge, which is represented in the knowledge base, to a particular case. Through application of these rules, it tries to infer a conclusion from answers supplied by the end-user, from calculations executed according to the prescribed rules and from consultation of some databases connected with the application. As such, driven by the different rules constituting the application, the inference engine establishes a link with the dictionary, controls the information flow using the user input or database fields and dynamically

builds the dialogue flow which results in the displaying of subsequent interaction screens, all contributing to the derivation of a final or intermediate conclusion for a practical case.
As the external rules have previously been compiled into pure Prolog, the inference engine is in fact ProLog by BIM itself. This approach guarantees a good performance and a response time of the application which is acceptable to the end-users.

3.2 THE EXPLANATION MODULE

This module is responsible for providing explanations about the proposed sc'utions in the form of a trace of the rules that have been used. These explanations are provided in the user language (French and Dutch). In complex rules applications, the user can navigate through the rules tree structure down to a fairly detailed level. Whenever possible, reference is made to the articles of the regulations which apply to the case. On user request, these articles are retrieved from the integrated documentary system and displayed on the screen for consultation and verification.

3.3 THE DICTIONARY

The dictionary is a repository which contains all the definitions of the regulation jargon used within the formal regulations written down in the rules. Each logical piece of regulation is defined as a concept in the dictionary so that it's possible to use it in the condition or conclusion part of the rules. For every concept used in the rules, it contains the following information items:

- typing information.
- information about the origin of the concept value: is it user input, is it stored in some database or is it derived from other concept values ?
- descriptions in the different user languages (French and Dutch). This portion is for documentation purposes only.
- scanned texts in the different user languages (French and Dutch). The canned texts are designed in such a way that they can be used at the same time for querying the user about the concept value and in the explanation process, to produce meaningful sentences, in the user's mother language.
- a possible default value.

4. DEVELOPMENT METHODOLOGY

The purpose of this section is to give a description of the methodology we are using for the development of the expert system knowledge base.

An important focal point of the methodology is the maintenance of consistency between the expert's interpretations of the regulations and the different modules of the application. First of all there needs to be a strict consistency between the different rules, between the rules and the explanation information, between the formal rules and the regulations and finally between the individual end-user tasks and the relevant regulations. This consistency checking is the responsibility of the knowledge engineer along with the expert, who is the best qualified to verify whether the formal rules are a correct interpretation of the regulations. This requires that a clear and structured representation of the rules is prepared for this verification so that interviews with the expert are structured and therefore more effective.

As an answer to this consistency problem we based our methodology on the law segmentation process described in [Schauss 90]. Thanks to the experience we gained during the development of the prototype (and the first part of the expert system) and thanks to discussions with representatives and experts of the Ministry mainly concerning organizational problems, we extended this methodology according to our needs. Our methodological approach is based on the following phases, which we describe below:

- isolation of end-user tasks;
- isolation and sorting of relevant articles;
- representation of articles in a roster form;
- identification of condition and conclusion concepts;
- conversion of roster representations into dendrographs;
- implementation and testing.

4.1 ISOLATION OF END-USER TASKS

For an expert system to be operational it is crucial that it will perform the tasks that end-users expect it to do, and that the normal procedures will not change drastically.

On the basis of interviews with end-users and of existing studies for previous automation projects, information about the focussed tasks is gathered. It is important for each task to be divided into sub-tasks if possible. Elementary sub-tasks are easier to map to the different articles of the regulations than a general task. It is also preferable to determine the sequence in which the tasks have to take place.

After their identification, tasks and sub-tasks are treated one by one. This is set out in the following paragraph.

4.2 ISOLATION AND SORTING OF RELEVANT ARTICLES

Per task, the relevant portions of the regulations are isolated to reduce the amount of information that needs to be handled by the knowledge engineer. For example, the knowledge engineer worked with the expert to identify the different articles that are related to the specific task of verifying whether a certificate is required for a particular case. Without going into details, it turned out that no less than 12 articles are related to this issue. Subsequently the expert added some crucial information by sorting them according to their importance and the sequence in which they need to be applied.

4.3 REPRESENTATION OF ARTICLES IN A ROSTER FORM

An article useful in a task is segmented into fragments which are horizontally (rows) and vertically (column) ordered in a roster. The conclusions of an article are put in the right most columns. Their conditions are put as follows:

- the conjunctions of conditions leading to a given conclusion (AND-conditions) are placed in a row;
- the alternative conditions (OR-conditions) are placed in columns.

Figure 3 presents a roster example taken from our application.

FIGURE 3. Roster representation

CONDITIONS			CONCLUSION
not in free practice			no certificate is required
exporting products	customs regulation where import is allowed with suspension of customs rights of levying of equal action working		
		particular regulation where export is allowed without the collection of export duties as meant in article 28	

The roster representation worked out to be helpful for readability and for easy validation by experts who are not necessarily familiar with a formal specification language or a rule language. With these representations, a first step towards rules, it is still possible for an expert to give advice about the interpretation of the articles such as explanations, interpretation and context of specific terms, missing information, etc.

For the example given in figure 3, the expert identified that the condition "free practice" has to be considered in the context of an import case and that the expression "active refinement" is always abbreviated to "A.R." at the Ministry. The roster therefore needed to be changed as in figure 4, where a header has been added for each column. This header is a collective term for the alternative conditions in a column.

FIGURE 4. Roster after expert's validation

CONDITIONS				CONCLUSION
type_ operation	free practise	customs _regulation	special_ regulation	required_ certificate
import	not in free practise			no
export		customs regulation where import is allowed with suspension of customs rights of levying of equal action working		
			A.R.	

4.4 IDENTIFICATION OF CONDITION AND CONCLUSION CONCEPTS

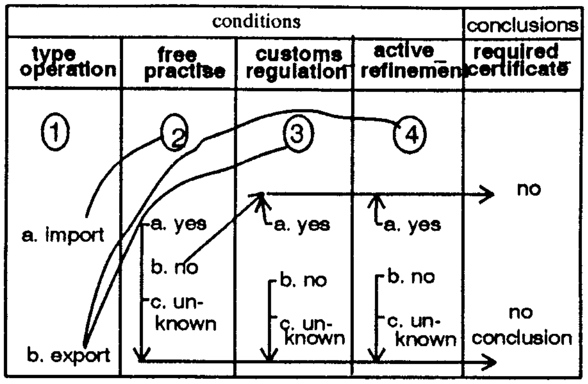
The following step consists of creating names for each roster column header and for each segment text in a column. The former are in fact generic concepts names while the latter are their possible instances.

The new created concepts are put in the dictionary of the application ([de Vey Mestdagh & Bos 89]). Such as described in paragraph 3.3. The new data in the dictionary must be validated by the expert.

4.5 CONVERSION OF ROSTER REPRESENTATION INTO DENDROGRAPHS

In the previous phase, the textual fragments in the roster representation were replaced by concepts and instances of these concepts. In the dendrograph representation, concepts instances coming from AND-conditions in the roster representation are linked via arches. Each linked group is then connected by an arrow to its corresponding conclusion concept. An example of such a dendrograph is given in figure 5.

FIGURE 5. Dendrograph



Finally, the dendrograph representation has to be validated by the expert.

4.6 IMPLEMENTATION AND TESTING

Once the final dendrograph built, it is very easy to derive formal rules. For example, the dendrograph in figure 5 has produced the following rules (figure 6):

FIGURE 6. Rules

REG 3719/88 ART 2 SEC 1 SUBP a SAYS

IF *TYPE_OPERATION IS IMPORT AND*
FREE_PRACTISE IS NO

THEN *REQUIRED_CERTIFICATE IS NO*

REG 3719/88 ART 2 SEC 1 SUBP b SAYS

IF *TYPE_OPERATION IS EXPORT AND*
CUSTOMS_REGULATION IS YES OR
ACTIVE_REFINEMENT IS YES

THEN *REQUIRED_CERTIFICATE IS NO*

Those formal rules are given as entry to the rules compiler which produces in turn their corresponding pure Prolog form. Nevertheless, even if the formalized knowledge is validated by the expert and is supposed to be consistent, it is still needed to test the implemented knowledge to check if the code is correctly entered in the knowledge base.

5. CONCLUSION

We presented the solid basis that we are using in the development of an operational expert system for the application of regulations. A clean architecture is used and the multi-languages problem is solved by means of its dictionary module. A structured methodology is followed for the knowledge base development; it aims at providing a formalization which is very readable, can be easily critiqued by the expert and is near to the form used in the implementation. This application has been proved successful on a first prototype of the system dealing with 40 pages of regulations.

ACKNOWLEDGMENTS

Particular thanks are due to the members of the Belgian Ministry of Economic Affairs and mainly to the project coordinators: Mr. Van Den Abeele, Mr. Debremaecker and Mr. Liekens. Also acknowledgements are due to the experts Mr. Boret and Mr. Nerinckx. The authors are also grateful to Mr Schauss for the development of the formalisation method from which theirs has been derived. At BIM, Michel Moreau, Paul Meurisse, Jean-Louis Binot, Eric Feron and Jan De Rijke are contributing to the successful development of the project. Final thanks to Brad Scurlock who read and commented this paper.

REFERENCES

[Bratko 86]
Prolog Programming for AI.
Ivan Bratko,
Addison Wesley, 1986.

[Brouwers & al 90]
Système expert d'aide à la rédaction de cahiers des charges dans les marchés publics d'acquisition de logiciels.
D. Brouwers, J. Gérard, E. Montéro,
CRID-Facultés Universitaires Notre Dame de la Paix,
Namur, Belgium.

[de Vey Mestdagh & Bos]
Juridisch systeem voor verwerking van tegenstrijdige aannames en afleidingen.
C.N.J. de Vey Mestdagh, G. Bos,
Kennissystemen, April 1989, 3.

[Nitta & al 88]
A Knowledge Representation and Inference System for Procedural Law.
Katsumi Nitta, Juntaro Nagao, Tetsuya Mizutori,
New Generation Computing, 5 (1988).

[Schauss 90]
Les Systèmes Informatisés d'Aide à la Décision Judiciaire.
Marc Schauss,
Congrès de l'Académie Internationale de Droit
Comparé, Aug. 90, Univ. McGill, Montréal.

[Sergot & al 86]
The British Nationality Act as a Logic Program.
M.J. Sergot, F. Sadri, R.A. Kowalski, F.Kriwaczek,
P. Hammond, H.T. Cory,
Communications of the ACM, May 1986 Vol. 29, 5.

[Venken & al 91]
Expert Systems Embedded in Administrative Applications: Approach and Example.
Raf Venken, Fabienne Balfroid, Geert Van Nevel,
to appear in "Operational Expert System in Europe"
G.P. Zarri, Ed., Pergamon Press (1991).