EXPERT SYSTEM KNOWLEDGE BASE CREATION AND TUNNING SUPPORT – NPS32 GRAPHICAL ADD-ON

Petr Polách
Doctoral Degree Programme (1), FEEC BUT
E-mail: polachpetr@phd.feec.vutbr.cz

Supervised by: Václav Jirsík
E-mail: jirsik@feec.vutbr.cz

ABSTRACT

This paper describes an empty diagnostic expert system that uses graphical interface for creation and modification of its knowledge base. Graphical environment contributes to transparency of the knowledge base structure and the inference processes taking place in it – already during the initial phase of the knowledge base tuning. It also deals with the inner functionality of the expert system and its other functions – including immediate consultation work-over based on modifications implemented into the knowledge base by the user.

1 INTRODUCTION

Expert systems (ES) come under the group of interactive computer programs for decision support. They substitute human experts – making their knowledge available anywhere, anytime and to anyone who uses the expert system (the purpose is lowering the costs as compared to the assistance of a human expert). Although these systems have been known for several decades, are systematically improved and their utilization serves well with obvious advantages, we do not come across as many expert systems in practise as we could expect. That is (among others) caused by the fact that the quality of expert systems – thus their commercial applicability – is dependent on the extensiveness and quality of knowledge embedded. The requirements, both technical and human, put on the *knowledge engineer* – the programmer that (in cooperation with the human expert) *fills* the system with knowledge – are considerable. Even minor failures in meeting these requirements may lead to a flop of the whole project, its insufficiency and the project's practical inapplicability. At the same time, the way knowledge is represented in expert systems (usually a text code with a syntax dependent on mathematical apparatus used in the particular ES) may, due to its complexicity and untransparency, create barriers in the knowledge engineer's task. The goal is therefore to give the knowledge engineer a tool that draws the expert system knowledge representation nearer to human thinking.

2 EXPERT SYSTEMS – KNOWLEDGE BASE

The *knowledge base* is a key component of every expert systems. It contains the knowledge of human experts who participated on development of the expert systems. This knowledge is

then used for solving end-user's specific problems. The process of finding answers using expert systems is called the *consultation*.
In our case, the knowledge base is formes a probability net composed of nodes and bindings. Nodes stand for hypotheses and rules, bindings handle information propagation.
The consultation process is controlled by *inference engine* according to user answers (users respond to questions prepared by experts during the knowledge base development phase). The answers are marked with probability values that are incorporated into the knowledge base during consultation – thus re-forming it from a "general" knowledge base into a *current problem model*.
The relation between the knowledge base and other components of a diagnostic expert systems is shown in Figure 1.

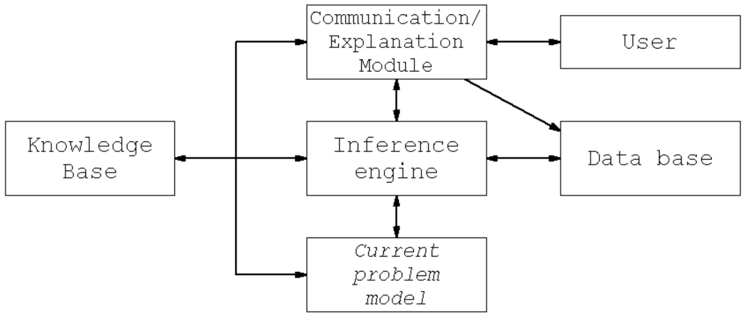

Figure 1: Diagnostic expert system structure

3 KNOWLEDGE BASE CREATION – GRAPHICAL MODE

3.1 M-CLONENPS32

M-CloneNPS32 expert system (Figure 4 shows its user interface) was created as a modification of already existing NPS32 ES. It is a schooling and experimental version based on MATLAB, focused on knowledge base operations. It includes a *fully graphical environment* for knowledge base creation and tunning as well as immediate automatic knowledge base count-over that shows the influence of modifications made by the user on knowledge nodes probablity values. Of course, adequate functions for expert consultations have been conserved.
The program was created in M-Language because of its suitability for matrix operations (as the knowledge base, graphical object listings, consultation steps history buffer, etc. are represented as matrices) therefore making the program easy to read for students.

3.2 KNOWLEDGE BASE REPRESENTATION

M-CloneNPS32 allows for import of the original NPS32 knowledge basses. That knowledge base pattern does not dispose of any capabilities for graphical representation, though. After importing the NPS32 knowledge base, it is needed to lay all the nodes and their bindings contained in the base out on the drawing desktop. Automatic oriented graph (which, in fact, is the expert system's knowledge base) lay-out is a complex mathematical task that goes beyond the scope of this paper as well as the M-CloneNPS32 project. A pre-existing solution was used. M-CloneNPS32 utilizes the lay-out algorithms of *Graphviz* package.

After import, the originally text-coded knowledge base is drawn into the knowledge base design desktop. The user can choose one of three different lay-out algorithms (as different algorithms may produce more or less suitable results for different kinds of knowledge bases). The desktop then provides the functionality for node adding or removal, position changes, bindings and rules adding or removing, node parameters changing and, of course, tracking the knowledge base information propagation by means of observing the changes of probability values during consultation.
More detailed functionality of this expert system is indicated by the diagram shown in Figure 2.

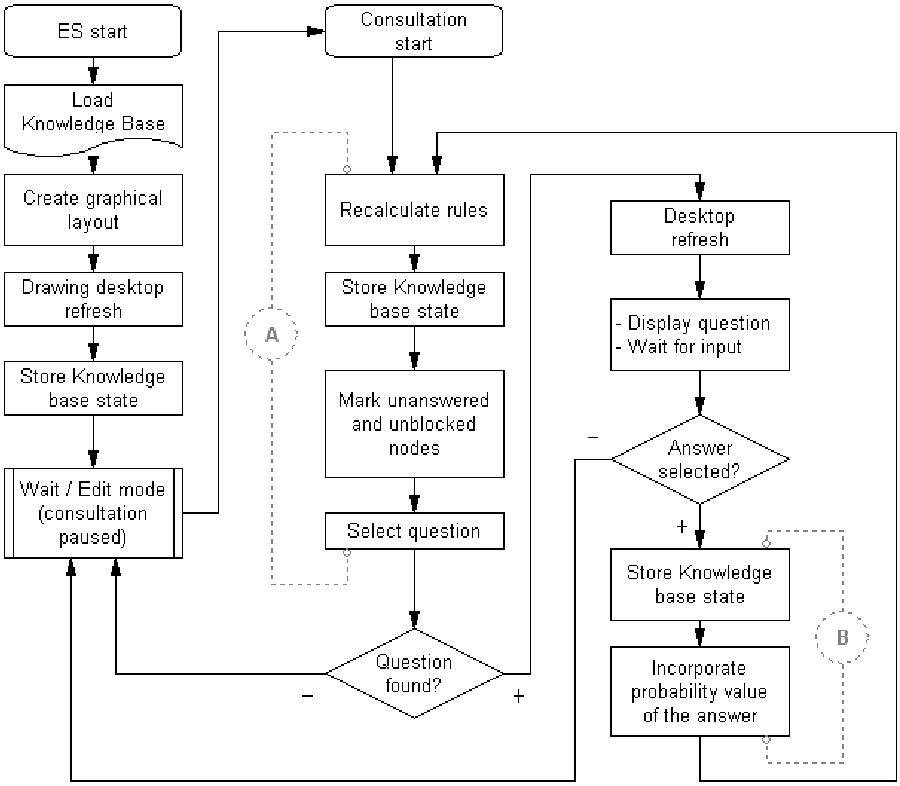

Figure 2: M-CloneNPS32 Consultation

3.3 KNOWLEDGE BASE MODIFICATIONS

From the user's point of view, there is no difference between the knowledge base creation/tuning mode and consultation mode. The program was designed so that when a significant change occurs in the knowledge base, the program counts the base immediately over – giving the knowledge engineers the option to see how the modification they made would affect any consultation done in the future. In case that a consultation had been started before the change was made, the knowledge base gets affected by the change as if the change was present from the beginning of consultation session (that is, the consultation acts the same way it would act in the end-user mode).
A situation may occur where the knowledge engineer modifies the base after *n* steps of consultation in order to find out how the intended changes would influence the probability values of the

goal-nodes (he needs to see the results exactly the same way the end-user would see them after taking the same *n* steps of consultation). Then also such modification could completely change the way the consultation would go. In such case the expert system lacks information needed for a complete knowledge base count-over (some answers that need to be answered were not answered during the previous *n* steps) preventing the automatic count-over from continuing. It stops at the point where the consultation first differs from the "historical" consultation scenario. The knowledge engineer is then given the choice to either continue and finish the consultation *manually* (i.e. the same way the end-user would), stop the count-over and go on modifying or discard the change and return to the previous state.
The expert system behaviour during base modification is shown in Figure 3.

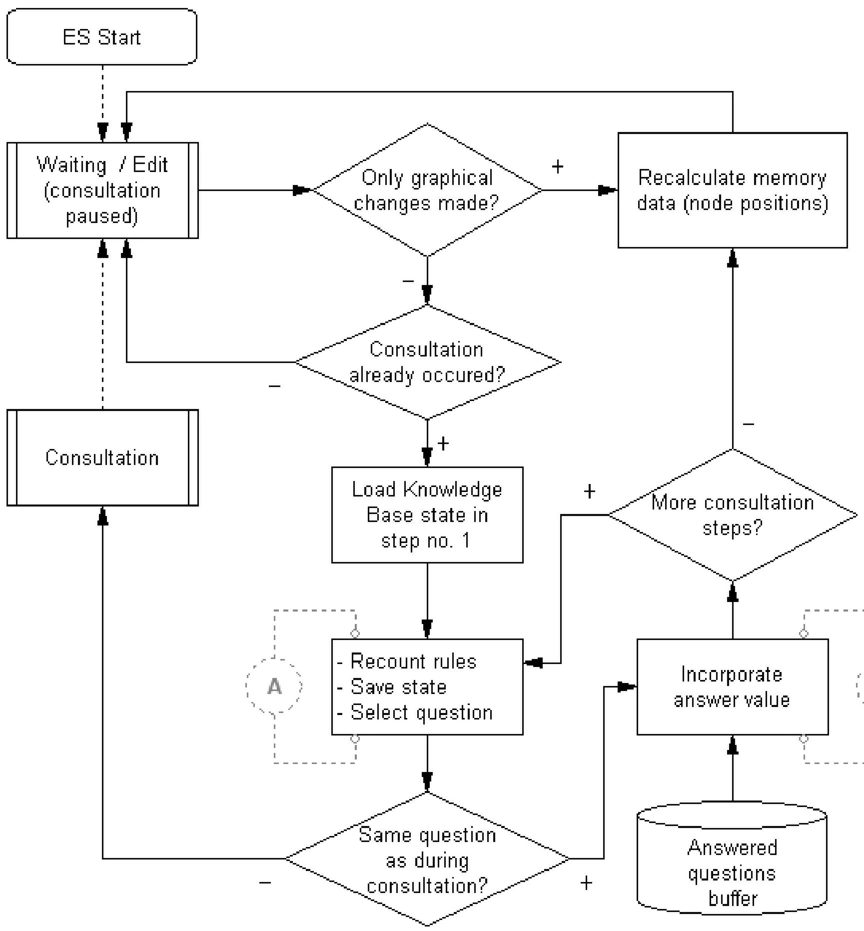

Figure 3: Knowledge Base recount

4 CONCLUSION

A modification of a well-proved expert system develop at the FEEC BUT was created. It significantly extends the possibilities for knowledge base creation and tunning. The implementation of graphical environment for knowledge base representation, modification and inference process tracking, as well as the implementation of functions for immediate verification of the

modifications influence on the consultation can be considered its substantial asset.
The program was written in M-Language for teaching purposes (better transparency of inner processes).

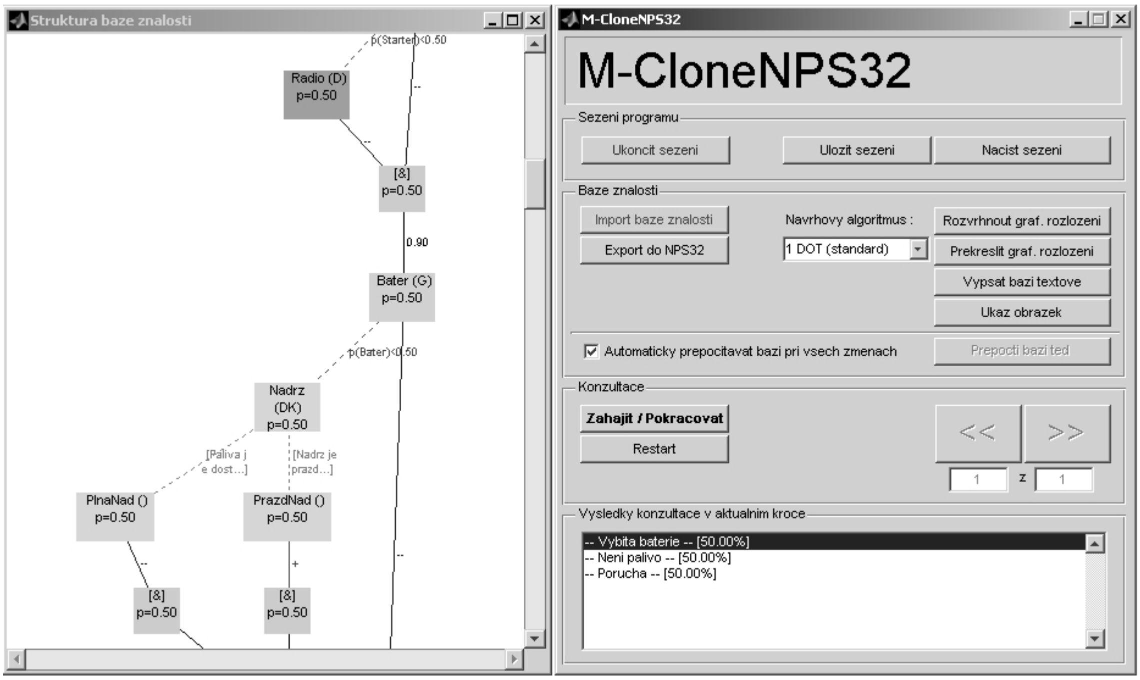

Figure 4: M-CloneNPS32 user interface

ACKNOWLEDGEMENTS

This paper has been prepared as a part of the solution of MŠMT research plan no. MSM0021630529 – "Inteligentní systémy v automatizaci".

REFERENCES

[1] MAŘÍK V., ŠTĚPÁNKOVÁ O., LAŽANSKÝ J. a kol.: *Umělá inteligence 2*. Academia, Praha, 1997, ISBN 80-200-0504-8

[2] JIRSÍK V., VALENTA J.: Expertní systém pro výběr automobilů. *Automatizace*, Praha, 2006, no. 5, p. 319–321, ISSN 0005-125X

[3] NORTH C.S.: *Neato User's Guide*. Technical Report 59113-921014-14TM, AT&T Bell Laboratories, Murray Hill, NJ, 1992.

[4] SUGIYAMA K., TAGAWA S., TODA M.: *Methods for Visual Understanding of Hierarchical System Structures*. IEEE Transactions on Systems, Man and Cybernetics, p. 109-125, 1981.