A Skeleton Rule-based Expert System Of New Generation

Submitted: 31st July 2012; accepted: 8th April 2013

Władysław Brzozowski

Abstract:
The paper presents skeleton rule-based expert system of a new generation, named EXPERT 3.0, worked out and programmed by the Author. Notion of a new generation refers here to implementation of a knowledge base of the system in a form of a computer database; previous skeleton expert systems implemented knowledge bases as text files. At first, a theory of expert systems, as one of the branches of Artificial Intelligence, is briefly presented. Then the Author's original algorithms of the system are described in the paper. Using the EXPERT 3.0 system, execution of the inference processes: forward, backwards or mixed, as well as of falsification of the main hypothesis, is possible. The EXPERT 3.0 system may be loaded with any number of parallel knowledge bases from such domains as technical, medical or financial diagnostics, as well as providing equipment, forecast and many other systems; in the paper, the inference process is illustrated by an example of the diagnostics of the damage to a MKM33 coal mill, working in a 200 MW power unit. Finally, conclusions and recommendations are formulated in the paper.

Keywords: *expert system, Artificial Intelligence, computer program, algorithm, inference process, fact, rule, technical diagnostics*

1. Introduction

Expert systems are one of the basic branches of an Artificial Intelligence (AI). Development of AI dates from the sixties of XX century. Artificial neural networks were the first developed branch of AI. Expert systems appeared in the seventies. Then, the evolution algorithms and fuzzy logic systems took place. A very promising new branch of AI, named the agents' theory, and also new branch named data mining, have appeared in the last few years.

Among the AI branches described above, expert systems were the first with important practical applications. The range of possible applications of the expert systems is very wide. Expert systems are mostly applicable to diagnostics: technical, medical, economic-financial and other, even such exotic as archaeological [1]. Besides diagnostics, expert systems may be applied to: forecasting, providing equipment, improvement, repairing, planning, interpreting, monitoring, control and instruction (teaching).

Expert system is a computer program. Since the seventies, many such computer programs had been written. However, due to the hardware and software evolution, these programs have become obsolete. It is necessary to constantly create new programs, written in new computer languages and developed in new computer environments. In this paper, EXPERT 3.0 computer program of the skeleton rule-based expert system of a new generation has been presented. This program has been worked out, written and developed personally by the Author. Notion of the new generation refers here to an implementation of the knowledge base of the system (as an element of AI), in a form of the computer database (as an element of computer science); previous skeleton expert systems, such as, for example, SOCRATES system [3], implemented the knowledge bases as text files.

Putting aside some mathematical formulae of an uncertainty management module, borrowed from solutions of the SOCRATES skeleton expert system [3], EXPERT 3.0 system is based on the Author's entirely original algorithms. These algorithms are described below in the paper. An execution of inference processes: forward, backwards or mixed, as well as of falsification of the main hypothesis, is possible using of the EXPERT 3.0 system. The system is actually principally utilized as a didactic tool in AI domain, but economic and industrial numerous applications of this system are also entirely possible. As an illustration of this thesis, a knowledge base of technical diagnostics of MKM 33 coal mill, working in a 200 MW power unit [4, 5], has been loaded to EXPERT 3.0 system, and serves as an example of inference process. After all, it is possible to load the EXPERT 3.0 system with any number of parallel expert systems (i.e. knowledge bases) from such domains as technical, medical or financial diagnostics, as well as providing equipment, forecast and many other systems.

2. The Algorithms of the EXPERT 3.0 System

The EXPERT 3.0 system is the rule-based expert system; knowledge in such a system is represented in form of facts and rules. The system comprises the following kinds of facts: introduced, intermediate and final, as well as following types of facts: enumerated, real, integer and logical. Value of introduced fact is determined by the user but not by the expert system. Value of intermediate or final fact (as opposed to introduced) is determined not by the user, but by the expert system, during the inference process, itself. Values of the final facts are an aim of the inference process. Enumerated fact is such a fact, which assumes the one value (or several values) from a set, predefined by the user (in the table of the values of enumerated facts, see below). Besides the enumerated facts, there are real and integer facts as well as logical facts, with values: TRUE, FALSE or UNKNOWN (the EXPERT 3.0 system bases on three-valued logic).

In the system, a rule has following syntax:

IF (PREMISE FIELD) THEN (CONCLUSION) (CF=?) (1)

where:
(PREMISE FIELD) – complex logical expression on a single elementary premise, or on any number of elementary premises, joined together with logical operators NOT, AND and/or OR, and also (optionally) closed in parenthesis on mutually nested hierarchical levels, in any number of these levels. The elementary premise is a comparison (of type equality or inequality) of current, actual value of single fact with its reference value. During the inference process, an elementary premise returns consequently logical value TRUE, FALSE or UNKNOWN. The reference value may be given in form of the constant. The constant may be simple (it represents then a definite invariable value) or complex, so-called computational. The computational constant is an arithmetical expression on a value of another fact; respectively a value of computational constant may vary during the inference process. It is possible to compare values of two specified facts in the elementary premise using the computational constant;
(CONCLUSION) – assignment specified value to a single, intermediate or final, fact. This assignment may be realized using a constant, simple or computational. In the EXPERT 3.0 system, a rule is so-called Horn's clause, i.e. it consists of one and only one conclusion;
(CF) – certainty factor of the rule. This is the variable from [0, 1] range, characterizing the confidence in the correctness of a rule. The CF equal to 1 means absolute correctness of a rule. The CF equal to 0 may not mean that the rule is incorrect, but means zero-confidence in the correctness of the rule. It is also possible to assign to the CF of the rule any intermediate value from [0, 1] range. The CF may not be interpreted as a probability (for example, CF equal to 0,5 does not mean that in a half of cases the rule is correct and in a second half – incorrect). The CF should be attributed not only to the rule but also to the value of each introduced fact. A special uncertainty management module, built-in into the EXPERT 3.0 system, then, from the CF of values of facts in premises, will compute the replacement CF of entire premise field, and then, taking into account also the CF of the rule, will compute the CF of conclusion. If one and the same conclusion is deduced from two or more "fired" (the notion of "fire" will be explained below) rules, an aggregated CF of conclusion is furthermore computed. The notion of logical t-norm and s-norm appears in some mathematical formulae of these calculations. The Author has borrowed these formulae from solutions of the SOCRATES skeleton expert system [3]. To make it possible to assign value to the fact in conclusion of the rule, this rule should be activated, or, according to the jargon of knowledge engineers, "fired". The rule can be "fired", if the logical value of premises field of this rule equals TRUE and the computed CF of conclusion of this rule is higher than a certain minimal threshold value. The notion of the CF is very important, taking into consideration general uncertainty with regard to different processes. The skeleton expert system, which does not consist of an uncertainty management module, should not be utilized. Niederliński [8] calls such a system (i.e. system in which all CFs are, a priori, equal 1) precise. It is of course misunderstanding; such a system should rather be called inexact.

2.1. Forward Inference Process

During forward inference process, the system deduces all possible conclusions taking into account actual values of introduced facts. The block-diagram in the Figure 1 presents algorithm of this process. In the blocks, the computations are realized according to formulae:

<u>Block 1</u>

$$\forall F \in FF \ \left(F := \text{unknown};\ CF_F := 0\right), \qquad \forall F \in FI \ \left(F := \text{unknown};\ CF_F := 0\right) \tag{2}$$

where:
F – fact;
FF – set of final facts;
FI – set of intermediate facts;
CF_F – CF of actual value of fact F.

<u>Block 3</u>
If, for example, the premise field has a form:

$$F_1 = RV_1 \text{ AND } (F_2 = RV_2 \text{ OR } (F_3 = RV_3 \text{ AND } F_4 = RV_4)) \tag{3}$$

where:
F_1, F_2, F_3, F_4 – facts (enumerated – in order to simplify the formulae);
RV_1, RV_2, RV_3, RV_4 – reference values of the facts F_1 - F_4,

then order of parameter computations will be as follows:

1) logical value of the premise $P_3 : F_3 = RV_3$, according to the truth table:

$F_3 =$	RV_3	other than RV_3	u_3
$LV_{p3} :=$	T	F	U

where:
T – TRUE; F – FALSE; U – UNKNOWN;
RV_3 – reference value of the fact F_3;
u_3 – unknown actual value of the fact F_3;
LV_{p3} – logical value of the premise P_3.

2) CF of the premise $P_3 : F_3 = RV_3$, according to the formula:

$$CF_{p3} = \begin{cases} 0, \text{ if } CF_{f3} < CF_t; \\ CF_{f3}, \text{ if } CF_{f3} \geq CF_t \end{cases} \tag{4}$$

where:
CF_{p3} – CF of the premise $P_3 : F_3 = RV_3$;
CF_{f3} – CF of actual value of the fact F_3;
CF_t – threshold value of CF.

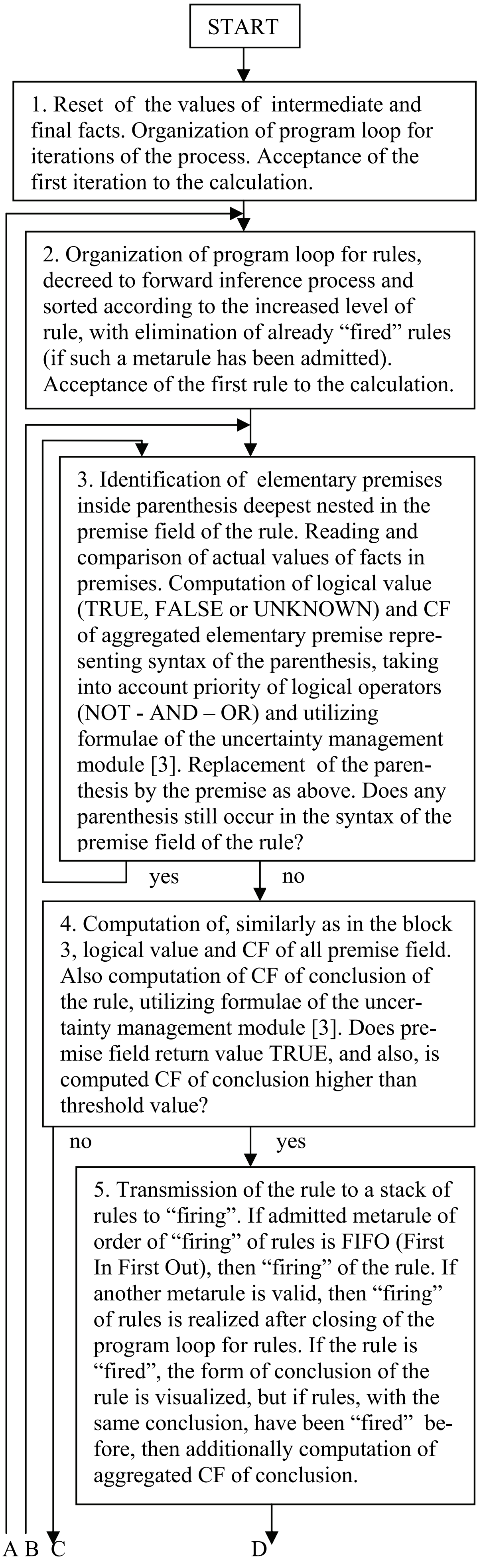
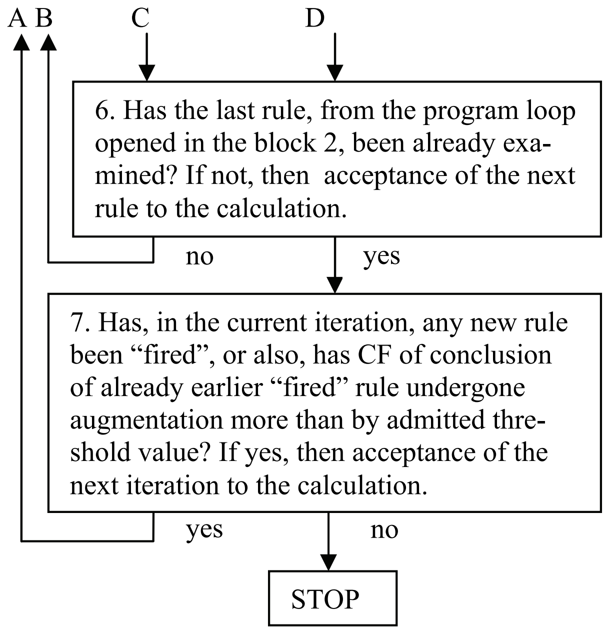



Fig. 1. The block-diagram of algorithm of forward inference process in the EXPERT 3.0 system

3) logical value and CF of the premise $P_4 : F_4 = RV_4$ (analogically as p. 1, 2 above);

4) logical value of the conjunction $C_{3\text{-}4} : F_3 = RV_3$ AND $F_4 = RV_4$, according to the truth table:

LV_{p3} =	T			F			U		
LV_{p4} =	T	F	U	T	F	U	T	F	U
$LV_{c3\text{-}4}$:=	T	F	U	F	F	F	U	F	U

where:
T – TRUE; F – FALSE; U – UNKNOWN;
LV_{p3} – logical value of the premise $P_3 : F_3 = RV_3$;
LV_{p4} – logical value of the premise $P_4 : F_4 = RV_4$;
$LV_{c3\text{-}4}$ – logical value of the conjunction $C_{3\text{-}4} : F_3 = RV_3$ AND $F_4 = RV_4$.

5) CF of the conjunction $C_{3\text{-}4} : F_3 = RV_3$ AND $F_4 = RV_4$, according to the formula (adapted from [3]):

$$CF_{c3-4} = \begin{cases} \min\left(CF_{p3}, CF_{p4}\right), & \text{for fuzzy t-l.n.,} \\ CF_{p3}CF_{p4}, & \text{for algebraic t-l.n.,} \\ \max\left(0, CF_{p3} + CF_{p4} - 1\right), & \text{for Lukaszewicz's t-l.n.} \end{cases} \tag{5}$$

where:
t-l.n. – t – logical norm, chosen by the user, accordingly to a nature of the knowledge base;
CF_{p3} – CF of the premise $P_3 : F_3 = RV_3$;
CF_{p4} – CF of the premise $P_4 : F_4 = RV_4$;
$CF_{c3\text{-}4}$ – CF of the conjunction $C_{3\text{-}4}$: $F_3 = RV_3$ AND $F_4 = RV_4$.

6) logical value and CF of the premise $P_2 : F_2 = RV_2$ (analogically as p. 1, 2 above);

7) logical value of the disjunction $D_{2\text{-}3\text{-}4}$: $F_2 = RV_2$ OR ($F_3 = RV_3$ AND $F_4 = RV_4$), according to the truth table:

LV_{p2} =	T			F			U		
$LV_{c3\text{-}4}$ =	T	F	U	T	F	U	T	F	U
$LV_{d2\text{-}3\text{-}4}$:=	T	T	T	T	F	U	T	U	U

where:
T – TRUE; F – FALSE; U – UNKNOWN;
LV_{p2} – logical value of the premise P_2 : $F_2 = RV_2$;
$LV_{c3\text{-}4}$ – logical value of the conjunction $C_{3\text{-}4}$: $F_3 = RV_3$ AND $F_4 = RV_4$.
$LV_{d2\text{-}3\text{-}4}$ – logical value of the disjunction $D_{2\text{-}3\text{-}4}$: $F_2 = RV_2$ OR ($F_3 = RV_3$ AND $F_4 = RV_4$).

8) CF of the disjunction $D_{2\text{-}3\text{-}4}$: $F_2 = RV_2$ OR ($F_3 = RV_3$ AND $F_4 = RV_4$), according to the formula (adapted from [3]):

$$CF_{d2\text{-}3\text{-}4} = \begin{cases} \max\left(CF_{p2}, CF_{c3\text{-}4}\right), & \text{for fuzzy s-l.n.,} \\ CF_{p2} + CF_{c3\text{-}4} - CF_{p2}CF_{c3\text{-}4}, & \text{for alg. s-l.n.,} \\ \min\left(1, CF_{p2} + CF_{c3\text{-}4}\right), & \text{for Lukas.'s s-l.n.} \end{cases} \quad (6)$$

where:
s-l.n. – s – logical norm, chosen by the user, accordingly to a nature of the knowledge base (alg. – algebraic, Lukas. – Lukasiewicz);
CF_{p2} – CF of the premise P_2 : $F_2 = RV_2$;
$CF_{c3\text{-}4}$ – CF of the conjunction $C_{3\text{-}4}$: $F_3 = RV_3$ AND $F_4 = RV_4$;
$CF_{d2\text{-}3\text{-}4}$ – CF of the disjunction $D_{2\text{-}3\text{-}4}$: $F_2 = RV_2$ OR ($F_3 = RV_3$ AND $F_4 = RV_4$).

9) logical value and CF of the premise P_1 : $F_1 = RV_1$ (analogically as p. 1, 2 above);

10) logical value of the conjunction $C_{1\text{-}2\text{-}3\text{-}4}$: $F_1 = RV_1$ AND ($F_2 = RV_2$ OR ($F_3 = RV_3$ AND $F_4 = RV_4$)) (analogically as p. 4 above);

11) CF of the conjunction $C_{1\text{-}2\text{-}3\text{-}4}$: $F_1 = RV_1$ AND ($F_2 = RV_2$ OR ($F_3 = RV_3$ AND $F_4 = RV_4$)) (analogically as p. 5 above).

Block4
The same formulae as in the block 3 and additionally (adapted from [3]):

$$CF_{conc} = CF_{pf}CF_r \quad (7)$$

where:
CF_{conc} – CF of the conclusion of the rule;
CF_{pf} – CF of the premise field of the rule;
CF_r – CF of the rule.

Block 5
If the conclusion has been deduced from two rules with the same conclusion, then (adapted from [3]):

$$CF_{aggr} = \begin{cases} \max\left(CF_{conc1}, CF_{conc2}\right), & \text{for fuzzy s-l.n.,} \\ CF_{conc1} + CF_{conc2} - CF_{conc1}CF_{conc2}, & \text{for alg.s-l.n.,} \\ \min\left(1, CF_{conc1} + CF_{conc2}\right), & \text{for Lukas.'s s-l.n.} \end{cases} \quad (8)$$

where:
s-l.n. – s – logical norm, chosen by the user, accordingly to a nature of the knowledge base (alg. – algebraic, Lukas. – Lukasiewicz);
CF_{aggr} – aggregated CF of the conclusion deduced from two rules;
CF_{conc1} – CF of the conclusion deduced from the first rule;
CF_{conc2} – CF of the conclusion deduced from the second rule.

2.2. Backwards Inference Process

Before backwards or mixed inference process, the user must put the main hypothesis. Then the system does not deduce all possible conclusions, but tries to prove only the main hypothesis. Only one of conclusions of rules, decreed to backward, or forward and backwards inference process, can be the main hypothesis. The block-diagram in the Figure 2 presents algorithm of backwards inference process. In the blocks, the computations are realized according to formulae:

Block 1
The same formulae as for the forward inference process (Fig. 1, block 1).

Block 3
The same formulae as for the forward inference process (Fig. 1, block 3–4) and additionally:

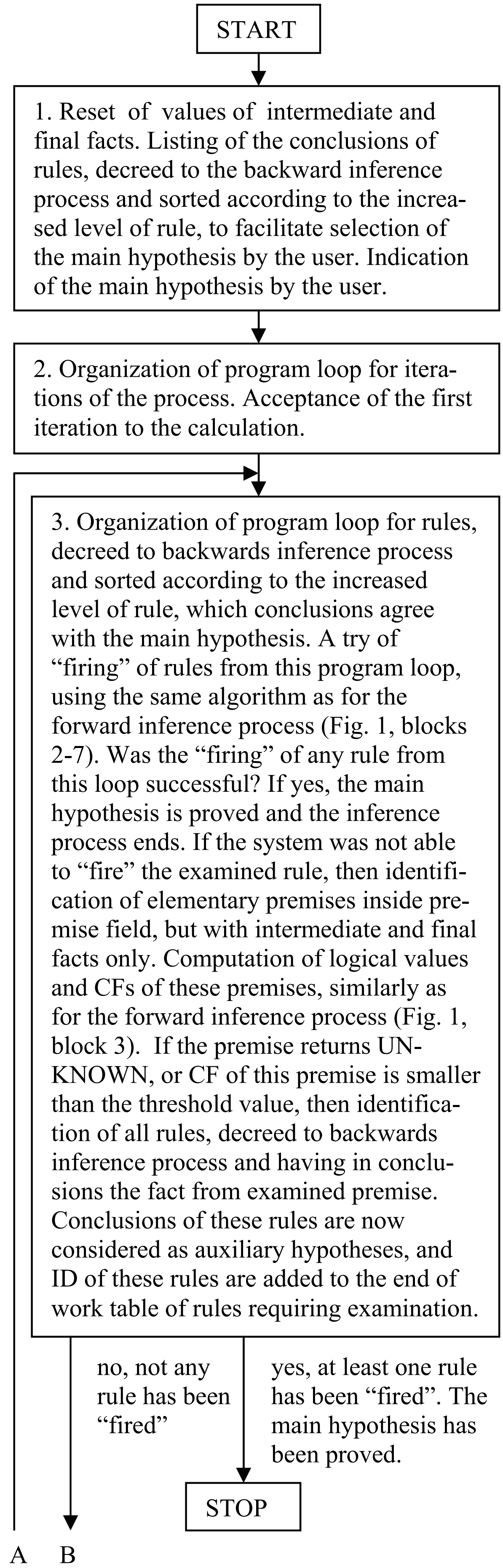
If, for example, the premise field of the examined rule has a form:

$$F_1 = RV_1 \text{ AND } F_2 = RV_2 \text{ AND } F_3 = RV_3 \quad (9)$$

where:
F_1 – fact enumerated (in order to simplify the formulae) and introduced;
F_2 – fact enumerated (as above) and intermediate;
F_3 – fact enumerated (as above) and final;
RV_1, RV_2, RV_3 – reference values of the facts $F_1 – F_3$,

and the premise P_1 : $F_1 = RV_1$ returns logical value TRUE, the premise P_2 : $F_2 = RV_2$ returns also logical value TRUE, but the premise P_3 : $F_3 = RV_3$ returns logical value UNKNOWN due to unknown actual value of the fact F_3, or returns TRUE but CF of this premise is smaller than the threshold value, then this examined rule cannot be "fired". At the same time, in the knowledge base, for example, there are following rules:

……………………..
Rule 1: IF (premise field) THEN $F_3=V_{3.1}$ (10)
……………………..
Rule 2: IF (premise field) THEN $F_3=V_{3.2}$ (11)
……………………..



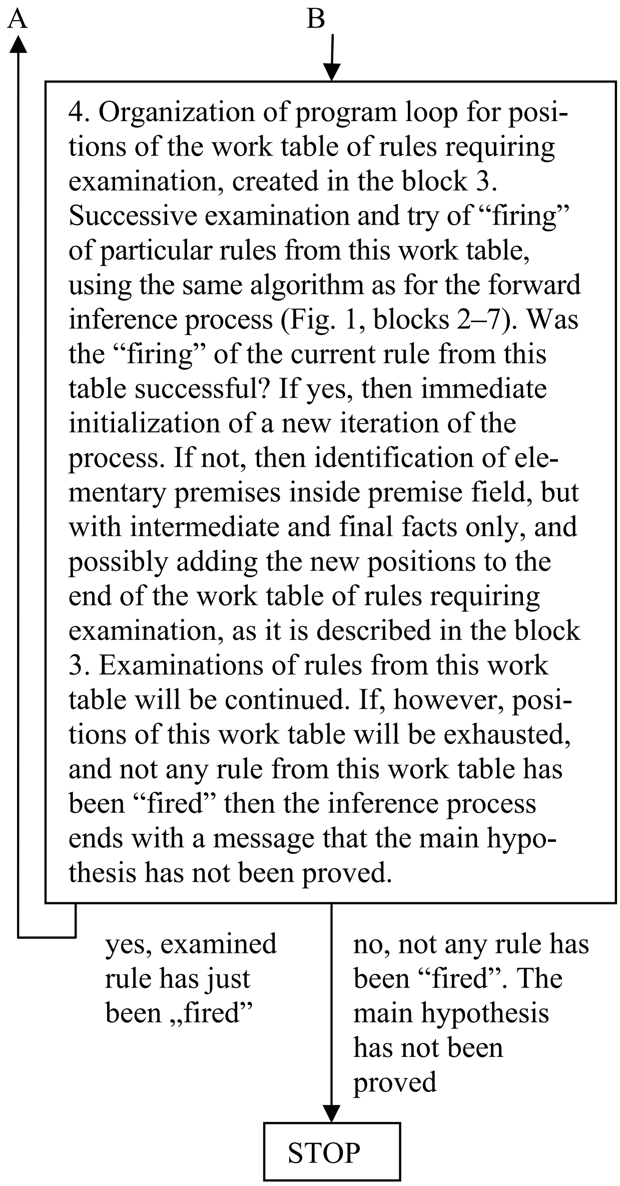


Fig. 2. The block-diagram of algorithm of backwards inference process in the EXPERT 3.0 system

where:
$V_{3.1}$, $V_{3.2}$ – values (all) of the enumerated fact F_3, predefined by the user (in the table of the values of enumerated facts, see below).

Consequently, the conclusions $F_3=V_{3.1}$ and $F_3=V_{3.2}$ will be auxiliary hypotheses. ID of the rules Rule 1 and Rule 2 will be added to the end of the work table of rules requiring examination.

<u>Block 4</u>
The same formulae as for the forward inference process (Fig. 1, block 3–4) and as in the block 3 above.

2.3. Main Hypothesis Falsification Process

If the system was not able to prove the main hypothesis (Fig. 2, block 4), it must not necessarily testify that this hypothesis is false. In this case, one should additionally try to realize falsification of the main hypothesis. To this end, one should prove another hypothesis that falsifies the main hypothesis. For example, if the main hypothesis (taken from the knowledge base of technical diagnostics of MKM 33 coal mill, working in the 200 MW power unit [4, 5]) is:

[Degree of failure of electric motor bearing No 1]=
[Initial failure], (12)

then the hypothesis, that falsifies the main hypothesis, may be:

[Degree of failure of electric motor bearing No 1]=
[Advanced failure]. (13)

This is the user's duty to indicate the hypothesis that falsifies the main hypothesis. The system assists this process, proposing to the user, as hypothesis that falsifies the main hypothesis, these conclusions of rules, in which, to the same fact, as in the main hypothesis, other values are assigned. Algorithm of falsification of the main hypothesis is identical as for the backwards inference process – its block-diagram will not be repeated. Conclusion with the same fact, as in the main hypothesis, must not be always hypothesis that falsifies the main hypothesis. For example, if the main hypothesis, in the expert system of medical diagnostics, is:

[Patient's disease]=[Meningitis], (14)

then the hypothesis, that falsifies the main hypothesis, may be:

[Patient's state of health]=[Good health]. (15)

Only the user can put such a hypothesis, and then may it prove using any kind of inference process.

2.4. Mixed Inference Process

Mixed inference process consists in realization forward and backwards inference process alternately. In the EXPERT 3.0 system, the mixed inference process begins with the backwards inference process. If the system is not able to prove the main hypothesis during this process (Fig. 2, block 4), it automatically switches to the forward inference process (Fig. 1, blocks 1–7), but without an initial reset of the values of intermediate and final facts, as well as without visualization of "fired" rules, others, than with conclusions staying in agreement with the main hypothesis. If the system is not able again to prove the main hypothesis during the forward inference process, the system automatically will switch again to the backwards inference process (Fig. 2, blocks 2–4), also without initial reset of the values of intermediate and final facts, and then will terminate inference process.

3. The computer Science Solutions of the EXPERT 3.0 System

The EXPERT 3.0 computer program has been written in Delphi 4 computer language and compiled in the RAD (Rapid Application Development) computer environment. The program is very wide; it consists of 115 forms/modules and 60.000 lines of source code. As mentioned above, the knowledge base of the system has been implemented in form of a computer database, using BDE (Borland Database Engine) database technology and Paradox local database system. A knowledge base has a form of computer database tables of: expert systems (i.e. knowledge bases), facts, values of enumerated facts, rules and constants. Two other tables, of: inference trajectories and multimedia files, are auxiliary tables. The database meets computer science requirement of so-called five normal shapes. The table of expert systems (i.e. knowledge bases) is connected with remaining tables by a one-to-many relationship. Additionally, the table of facts is also connected with the table of values of enumerated facts by the same relationship. Also the same relationship exists between the tables of facts/rules and the table of multimedia files.

Record of the table of expert systems (i.e. knowledge bases) consists of the following columns (fields): ID of the expert system (primary key); Name of the expert system; Description of the expert system; Date of creation of the record.

Record of the table of facts consists of the following columns (fields): ID of the fact (primary key); ID of the expert system (foreign key); Name of the fact; Kind of the fact (with items: introduced, intermediate, final); Type of the fact (with items: enumerated, real, integer, logical); Unit of measure (real and integer only); Number of values of the fact (20 max.; enumerated only); Is the fact multi-valued? (with items: yes, no; enumerated only); Description of the fact; Instruction how to determine value of the fact (introduced only); Value of the fact (integer, real and logical only), introduced or intermediate/final at the end of the current iteration; CF of the value of the fact as above; Value of the intermediate/final fact (integer, real and logical only) at the end of the previous iteration; CF of the value of the fact as above; Date of creation of the record.

Record of the table of values of enumerated facts consists of the following columns (fields): ID of the value (primary key); ID of the fact (foreign key); Value of the fact; Does the fact (introduced or intermediate/final at the end of the current iteration) have this value? (with items: yes, no); CF of the value of the fact as above; Did the intermediate/final fact have this value at the end of the previous iteration? (with items: yes, no); CF of the value of the fact as above; Date of creation of the record.

Record of the table of rules consists of the following columns (fields): ID of the rule (primary key); ID of the expert system (foreign key); Name of the rule; Level of the rule; Inference sessions possible (with items: forward, backwards, both); CF of the rule; Text of the premise field of the rule; Text of the conclusion

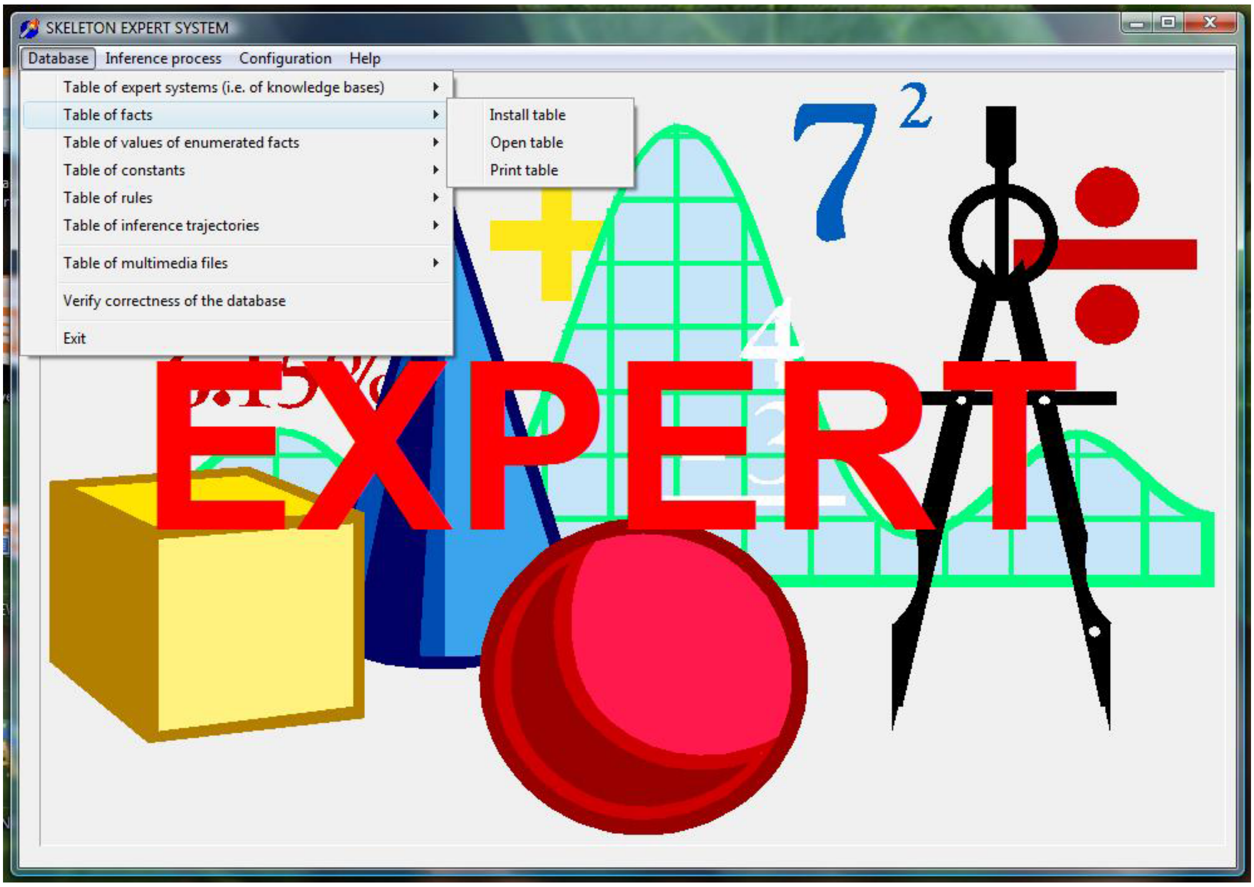


Fig. 3. The title form of the EXPERT 3.0 computer program

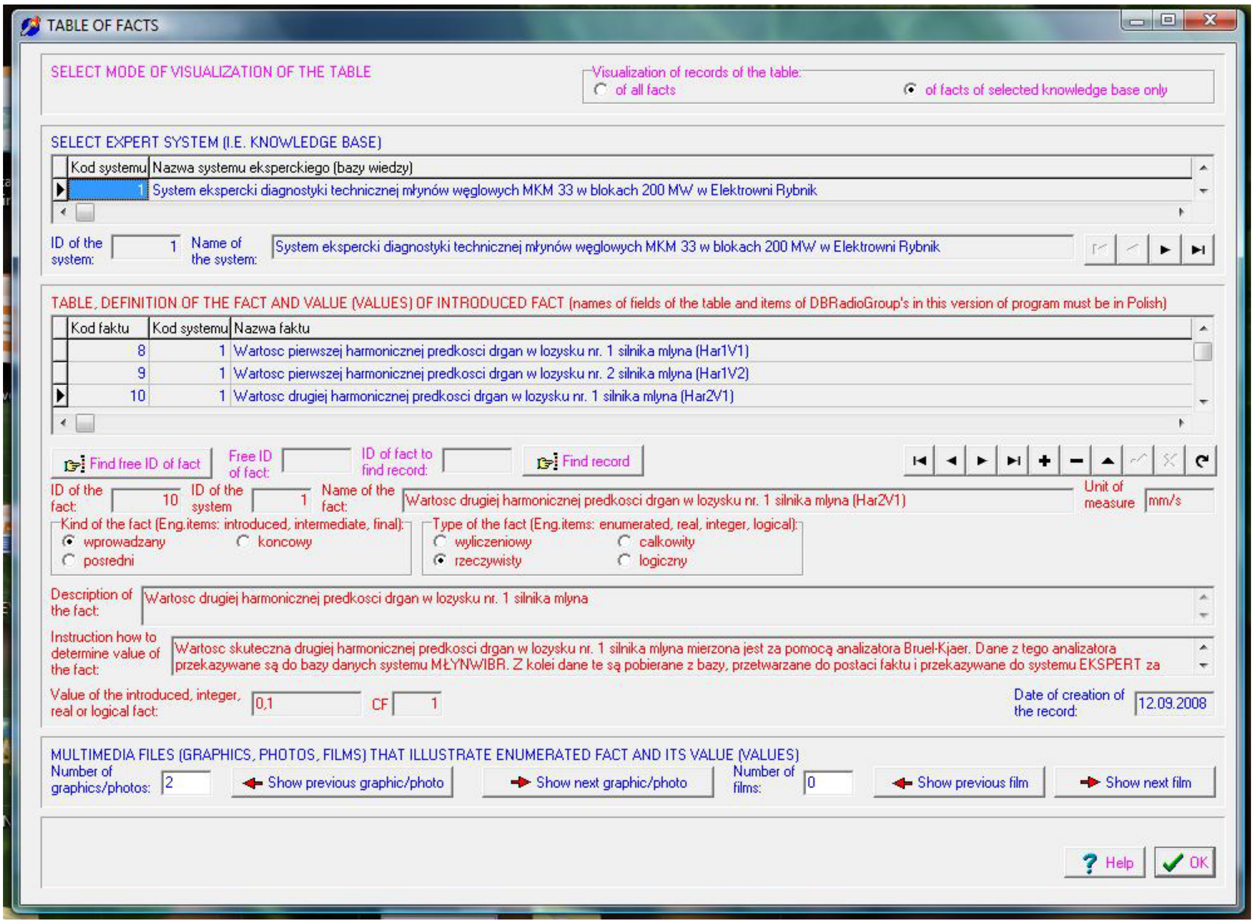


Fig. 4. The form of the table of facts of the EXPERT 3.0 computer program

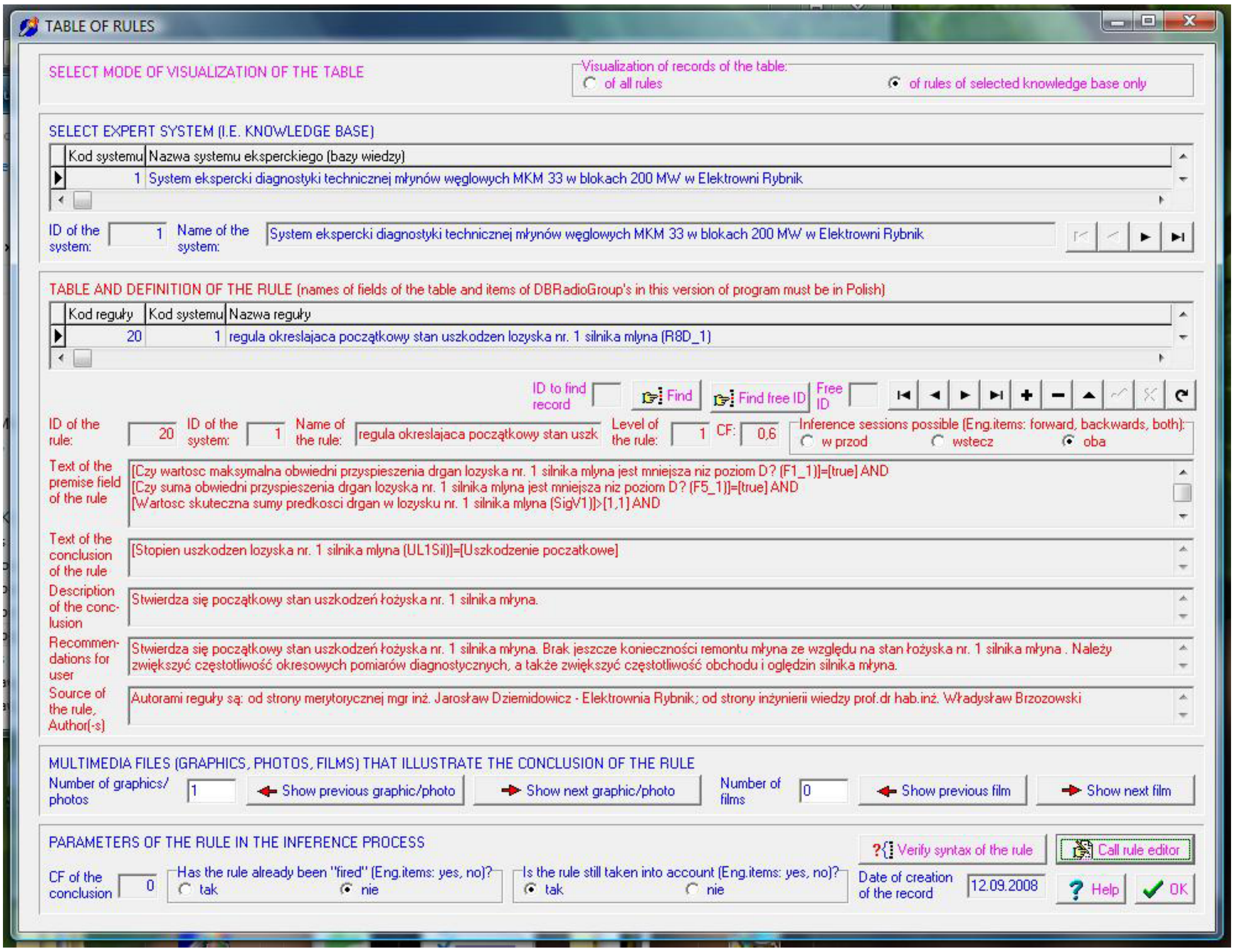


Fig. 5. The form of the table of rules of the EXPERT 3.0 computer program

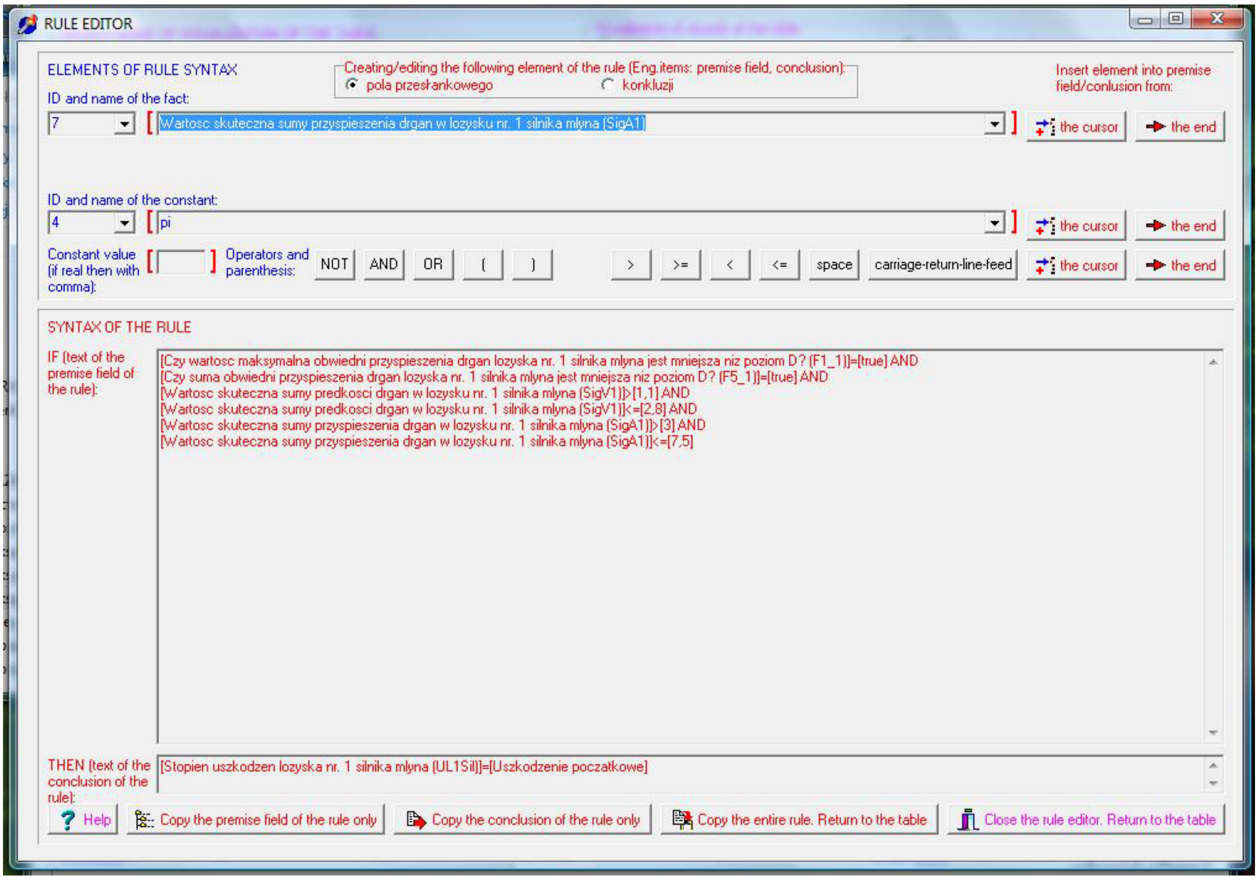


Fig. 6. The form of the rule editor of the EXPERT 3.0 computer program

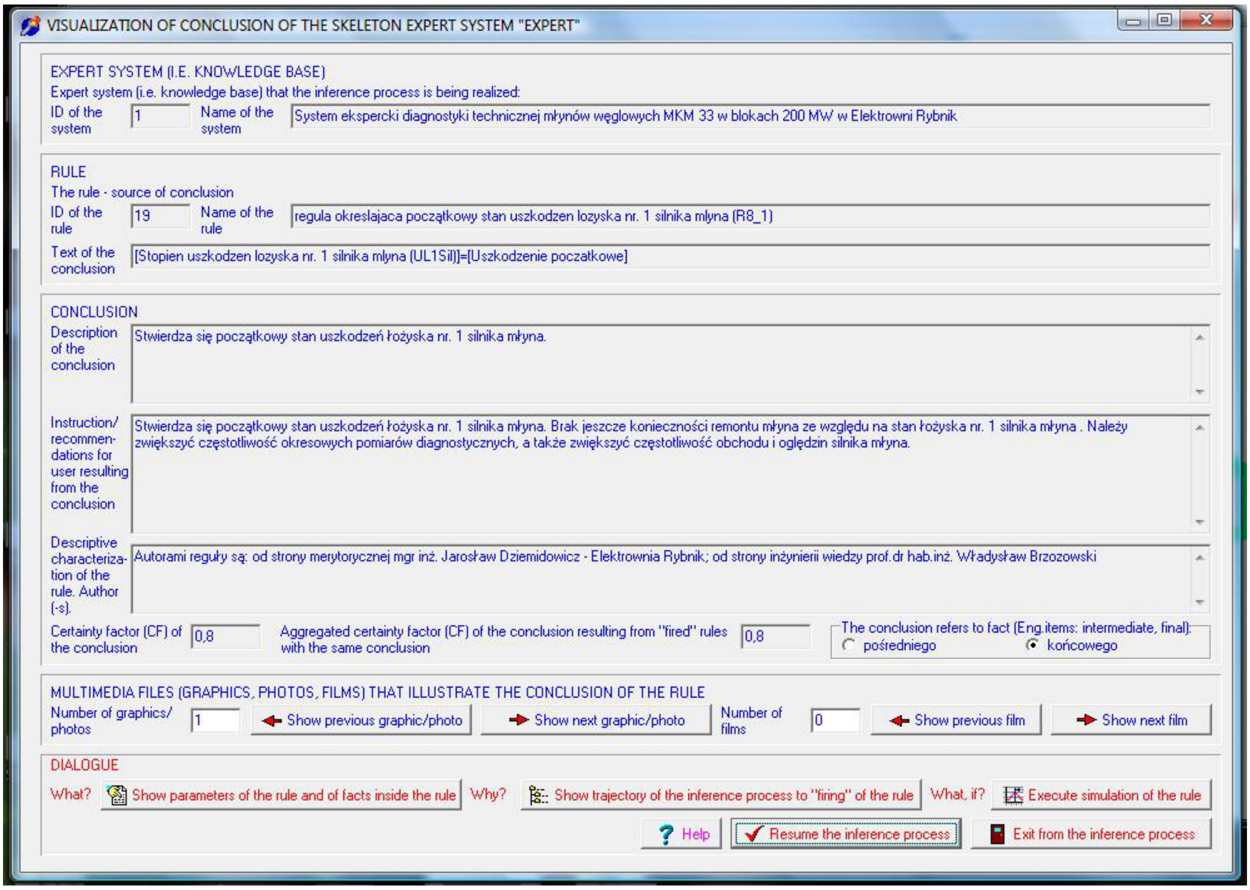


Fig. 7. The form of visualization of conclusion of the EXPERT 3.0 program

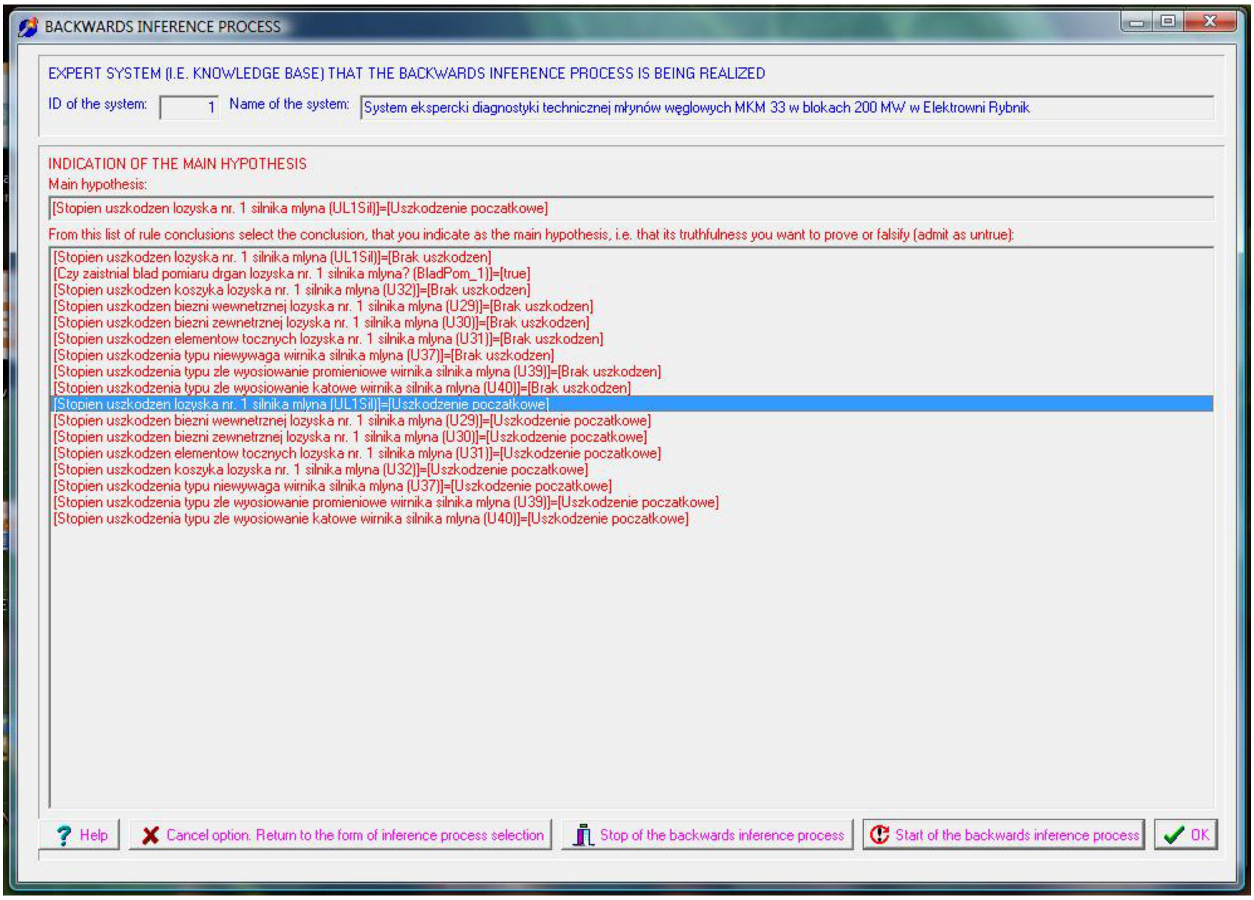


Fig. 8. The form of the backwards inference process of the EXPERT 3.0 computer program

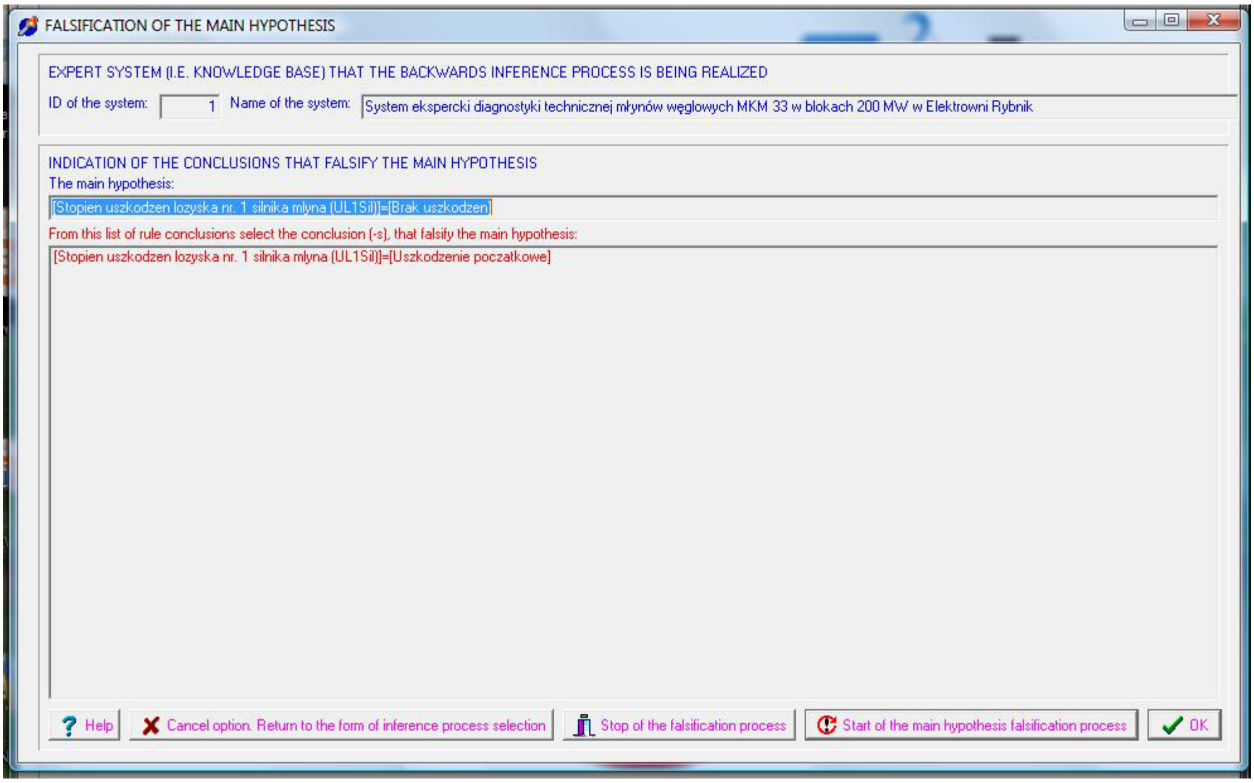


Fig. 9. The form of the main hypothesis falsification of the EXPERT 3.0 computer program

of the rule; Description of the conclusion of the rule; Instructions/recommendations for user after "firing" of the rule; Description/source of the rule. Author(-s); Has the rule already been "fired"? (with items: yes, no); CF of the conclusion of the rule at the end of the current iteration; Is the rule still taken into account? (with items: yes, no); Date of creation of the record.

Record of the table of constants consists of the following columns (fields): ID of the constant (primary key); ID of the expert system (foreign key); Name of the constant; Kind of the constant (with items: simple, computational); Type of the constant (with items: real, integer, logical); Unit of measure (real and integer only); Value of the constant (simple only); ID of the argument (fact) of the constant (computational only); Value of the parameter A of the constant (constant = A*argument+B; computational only); Value of the parameter B of the constant (as above); Description of the constant; Instruction how to determine value of the constant (simple only); Date of creation of the record.

Record of the table of inference trajectories consists of the following columns (fields): ID of the trajectory step (primary key); ID of the expert system (foreign key); ID of the rule examined in this trajectory step; Current level of rules in this trajectory step; Current number of iteration of the inference process; Current direction of the inference process (with items: forward, backwards); Has the examined rule been "fired" in this trajectory step? (with items: yes, no); CF of the conclusion of the "fired" rule in this trajectory step; The examined rule has not been "fired" in this trajectory step due to: (with 9 different items); Date of creation of the record.

Record of the table of multimedia files consists of the following columns (fields): ID of the file (primary key); Name of the file (filename and extension); Kind of the file (with items: graphic/photo, film); Height/width of picture/screen of the file ratio; What does the file illustrate? (with items: fact, rule); ID of the illustrated fact/rule (foreign key); Description/caption of the graphic/photo/film; Date of creation of the record.

In accordance with the notion of the skeleton expert system, all tables are entirely programmed, but they are, in the distributed version of the system, empty. This is user's duty to fill these tables with a suitable proper knowledge from certain needed domain.

The structure of database of the EXPERT 3.0 system is presented below in the title form of the computer program (Fig. 3).

Parameters of facts are presented in the form of table of facts (Fig. 4). If determined predefined graphic image (in the table of multimedia files) is assigned to the value of the determined enumerated fact, it is possible to create a rule with graphic premises. Consequently, the EXPERT 3.0 system may perform similar functions as the SCANKEE skeleton expert system [6]. In the EXPERT 3.0 system, multimedia files (graphics, photos and films) may be also used to instruct the user, regarding assignment values to introduced facts (for example, by measurement), and, particularly, regarding interpretation of conclusions and/or undertaking necessary activities (for example repairs, overhauls) resulting from the conclusions of the system (Fig. 7).

Parameters of rules are presented in the form of table of rules (Fig. 5) and in the form of rule editor

(Fig. 6). The following rule, taken from the knowledge base of technical diagnostics of MKM 33 coal mill, working in the 200 MW power unit [4, 5], shown in these forms as an active record, may serve as an example of syntax of simple rule in the system EXPERT 3.0 (after translation into English):

IF
[Is maximum envelope of vibration acceleration of electric motor bearing No 1 of mill less than level D?]=[true] AND [Is sum of envelope of vibration acceleration of electric motor bearing No 1 of mill less than level D?]=[true] AND [Root-mean-square value of sum of vibration speed of electric motor bearing No 1 of mill]>[1.1] AND [Root-mean-square value of sum of vibration speed of electric motor bearing No 1 of mill]<=[2,8] AND [Root-mean- square value of sum of vibration acceleration of electric motor bearing No 1 of mill]>[3] AND [Root-mean-square value of sum of vibration acceleration of electric motor bearing No 1 of mill]<=[7,5]
THEN
[Degree of failure of electric motor bearing No 1 of mill]=[Initial failure] (16)

Premise field of this rule consists of 6 elementary premises, joined together with conjunction. In the syntax of each elementary premise, the name of fact, the comparison operator and the reference value are present. Differently from the other skeleton expert systems, in which only illegible identifiers of facts occur in the rules, here full name of fact, with spaces, Polish diacritical letters and other characters, all enclosed in square brackets, are introduced into the rule. The rule becomes very readable. The same manner refers to the values of the facts, especially of enumerated facts. It may seem that generation of such a rule, considering necessity of introduction of full names of facts, is very arduous. On the contrary, this process is very simple, thanks to the rule editor built-in into the system. The name of the fact is only once introduced into the system (in the table of facts). Then, the rule editor copies names of the facts, clicked by the user from a list, to the different created rules.

It is also worth noticing that the logical facts in two first elementary premises in the shown above exemplary rule are so-called coverings, i.e. certain aggregated parameters computed by foreign computer programs, that usually mediate in preparation and transmission values of introduced facts into the expert system. Such an organization considerably hastens and facilitates realization of the inference process.

During the inference process, after each "firing" of rules, the form of visualization of rule conclusion is shown (Fig. 7). Before the backwards or mixed inference process, the user selects the main hypothesis in the form, shown in the Fig. 8. Before the main hypothesis falsification process, the user selects hypothesis (hypotheses) that falsify the main hypothesis in the form, shown in the Fig. 9.

4. The Applications of the EXPERT 3.0 Program

As mentioned above, the program is actually principally utilized as a didactic tool in AI domain. Using this program, students created already hundreds of knowledge bases from different domains.

As an industrial application of this program, the knowledge base of technical diagnostics of MKM 33 coal mill, working in a 200 MW power unit, has been transferred from old SOCRATES skeleton expert system [4, 5] and loaded to the EXPERT 3.0 system.

Technical diagnostics is a branch of science which evolved from the theory of exploitation, cybernetics and reliability. Among the many methods of technical diagnostics, the methods of vibration/acoustic diagnostics have found special applications in the power plants. These methods are applicable anywhere in a technological process where vibrations and noise occur and where device failures may be the cause of these vibrations and noise.

The systems of vibration/acoustic diagnostics are particularly applicable to machines and rotational devices, such as turbo sets, feed water pumps, cooling water pumps, condensation pumps, coal mills, flue gas fans, air fans, mill fans and others. The computer in such systems processes signals of vibration displacement, collected from many sensors, differentiates these signals twice, takes them into the Fourier series and calculates amplitudes and RMS (root-mean-square) values of individual harmonics of vibration speed and acceleration (see the exemplary rule (16) above).

On the basis of these parameters, specialists in vibration/acoustic diagnostics decide about the technical state of a device, and especially about the presence and degree of progress of typical device failures. The purposes of this diagnostic analysis are: a) to lengthen the durability and life of the material in the machines; b) to determine principles for rational exploitation of the machines; c) to determine the scope of necessary replacement of the machines and devices; d) to ensure that damage does not reach a point which threatens power unit break-downs and the destruction of the machine or device; e) to determine the scope of maintenance work and its timing and f) to optimize maximum elongation of overhaul life. The range of knowledge and experience of vibration/acoustic diagnostics is already enormous. In order to make this knowledge accessible and to utilize it, one must use methods of knowledge engineering. Expert systems are especially applicable here.

The knowledge base of technical diagnostics of MKM 33 coal mill consists of above 500 rules and 300 facts (in this number: 200 introduced facts, principally vibration/acoustic parameters). The system diagnoses above 50 elementary failures of coal mill. The system was utilized in one of the big Polish power plant [4, 5].

5. Conclusions. Recommendations

Expert systems are important tools in many branches of world/national economy and industry, especially as the systems of technical, medical and financial diagnostics.

It is recommended to utilize modern computer programs of expert systems, written and developed in contemporary computer languages and RAD computer environments, as user friendly programs. Such programs should utilize computer database, as modern computer science tool, to load parameters of knowledge base.

Expert system should consist of the uncertainty management module.

Expert system should make it possible to realize the forward, backwards and mixed inference process, as well as the main hypothesis falsification process.

AUTHOR

Władysław Brzozowski – Retired associate professor (in a domain of electric power engineering and also of computer science –Artificial Intelligence) of Częstochowa University of Technology, Institute of Electric Power Engineering in Częstochowa (Poland); Retired associate professor (in a domain of computer science – computer languages and programming) of The Gen. Jerzy Ziętek Silesian (Higher) School of Management in Katowice (Poland);
E-mail address for correspondence:
wlad_brzoz@poczta.onet.pl.

REFERENCES

[1] Gondran M., *Introduction aux systèmes experts*, Eyrolles, Paris 1984.
[2] Cholewa W., Pedrycz W., *Systemy doradcze*, Skrypt nr. 1447, Politechnika Śląska, Gliwice 1987. (in Polish)
[3] Iwański C., *Instrukcja obsługi systemu szkieletowego SOCRATES*, IBS PAN, Warszawa, 1990. (in Polish)
[4] Brzozowski W., Dziemidowicz J., „Ekspercki system diagnostyki i remontów młynów węglowych". In: *Materiały VI Międzynarodowej Konferencji Naukowej „Aktualne Problemy w Elektroenergetyce"*, KE PAN O/Katowice – IEiSU Politechnika Śląska, Gliwice – Kozubnik, 1993. (in Polish)
[5] Brzozowski W., „Struktury i metody pozyskiwania wiedzy w systemie diagnostyki technicznej na przykładzie systemu eksperckiego EKSPEM". In: ed. Kulikowski R., Bogdan L., *Wspomaganie decyzji. Systemy Eksperckie, Materiały Konferencji*, IBS PAN, Warszawa, 1995. (in Polish)
[6] Guzowska-Świder B., „Zastosowanie systemu ekspertowego SCANKEE do symulowania procesu analizy chemicznej", red.: Kulikowski R., Bogdan L., *Wspomaganie decyzji. Systemy Eksperckie*, Materiały Konferencji, IBS PAN, Warszawa, 1995. (in Polish)
[7] Mulawka J.J., *Systemy ekspertowe*, WNT, Warszawa, 1996. (in Polish)
[8] Niederliński A., *Regułowe systemy ekspertowe*, Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Gliwice, 2000. (in Polish)
[9] Brzozowski W., "Expert Systems of the Technical Diagnostics in Power Plants", *Przegląd Elektrotechniczny*, no. 3, 2009, pp. 124–129.
[10] Barzilay R. and others, *A new approach to expert system explanations*, 2012 (http://www.cogentex.com/papers/explanation-iwnlg98.pdf).