Anna Golik, Eng., Maciej Golik, Eng., Rafał Dreżewski, PhD, Eng.
The School of banking and Management
abaldyga@wszib.edu.pl, mgolik@wszib.edu.pl, drezew@wszib.edu.pl

DEVELOPMENT AND FUNCTIONING OF EXPERT SYSTEMS AS ILLUSTRATED BY SELECTED EXAMPLES

Introduction

In many areas of human activity a person with theoretical knowledge and practical experience (i.e. an expert) is usually an indispensable element of a decision making process. However, it often happens that an expert in a given field is not available or difficult to contact. The idea of expert systems appeared as a solution to such problems. In some cases, an expert system is able to replace a human expert in a given field or to function as a system supporting his/her decision making processes by providing alternative solutions to problems under investigation.

The practical aim of developing expert systems is to support key human decisions **(to provide expert opinion**) by drawing conclusions from the expert's knowledge that is introduced to the system. In some cases it is even aimed to replace the human thinking process by a machine's reasoning algorithm[1]. Currently, an immense amount of data is collected that comes from various sources (statistics, medical files, seismographs, Large Hadron Collider, etc.). In order to draw conclusions from data, the assistance of the computing power of computers is required and expert systems can be applied, which *provide the opportunity of reliable reasoning from historical data* [2] (i.e. from the data collected before).

The term *expert system*s may also include self-learning expert systems, i.e. the ones that are capable of acquiring new knowledge, presenting it in their structure and applying to the tasks set. A simple definition of **machine learning**, which is included in Paweł Cichosz's book is helpful in understanding that kind of expert systems. It defines self-learning machine as computer software that *is capable of (..) improving the quality of its functioning on the basis of its past experience*[3]. A further part of the definition presented in the book emphasizes **the self-acquisition of knowledge** by a self-learning expert system: the self-learning of the system is the every autonomous change in the system resulting from its experience, which leads to the improvement of the quality of its

[1]The fears against human beings being replaced by computer inspired several S-F authors (Stanisław Lem, Jacek Dukaj, William Gibson) as well as contemporary sociological and cultural schools (including Jean Baudrillard and Neil Postman)
[2] Cichosz P., *Systemy uczące się*, Wydawnictwa Naukowo-Techniczne, Warszawa 2000, p. 40.
[3]*Ibidem*, p. 18.

functioning[4]. A brief definition of machine learning is also given by Marek Jan Kasperski: *To sum up, machine learning (…) consists in increasing the amount of information in the database, the correction of the information in the database and the improvement of the database search system, together with the generation of new relationships between the possessed information*[5].

In opposition to the above definitions, there are expert systems that are not self-learning and only possess a fixed knowledge base that does not develop in the course of their work. Generally, it can be stated that an expert system is the computer software that aims at supporting or replacing a human being in making difficult decisions on the basis of various prerequisites with the use of a particular **reasoning algorithm.**

The term **Artificial Intelligence**, which is related to the idea of expert systems, refers to the ability to solve new problems. *A formation – being it either a brain or an engineering system – is intelligent if , when confronted with a completely new situation, it can solve new problems referring to its database. That ability and the readiness to solve new problems is the measure of intelligence*[6].

The science on artificial intelligence (AI) includes scientific areas dealing with expert systems. At present, AI is applied in such different areas as computer science, robotics, psychology, cognitive science and biology. It is worth mentioning that the fundamentals of AI as a field of science generated from such fields of study as mathematics (especially formal logics, theory of probability, theory of information, statistics) and psychology.

AI theory was created in 1950s[7] and the first expert systems appeared in 1960s (e.g. DENDRAL- produced at the Stanford University to identify molecular structures of chemical compounds) and in 1970s (e.g. MYCIN, also from the Stanford University, which is still popular and used to identify blood disorders). Since that time several algorithms of machine learning and expert systems have been developed. Neural networks, genetic algorithms and semantic networks are – among others – the examples of self-learning algorithms, i.e. the methods of knowledge acquisition by computer software. Some more detailed information on the subject can be found in Paweł Cichosz's *Systemy uczące się*, where the author discusses mainly inductive and probability reasoning, Prof. Ryszard Tadeusiewicz's *Sieci neuronowe*[8] and Prof. Jarosław Kasperski's *Wykłady z*

---

[4]*Ibidem*, p. 34.
[5]Kasperski M. J., *Sztuczna inteligencja*, Wydawnictwo Helion, Gliwice 2003, p. 140.
[6]Interview with Prof. Ryszard Tadeusiewicz – *„Ja" – co to takiego?*, w: Rożek T., *Nauka po prostu. Wywiady z wybitnymi*, Wyd. Demart SA, Warszawa 2011, p. 116.
[7] The term *Artificial Intelligence* was introduced by John McCarthy in 1956
[8] Tadeusiewicz R., *Sieci neuronowe*, Akademicka Oficyna Wydaw. RM, Warszawa 1993. Accessible (in large part) at: http://winntbg.bg.agh.edu.pl/skrypty/0001/.

algorytmów ewolucyjnych[9]. An impressive bibliography is to be fund in Marek Jan Kasperski's book entitled *Sztuczna inteligencja* .

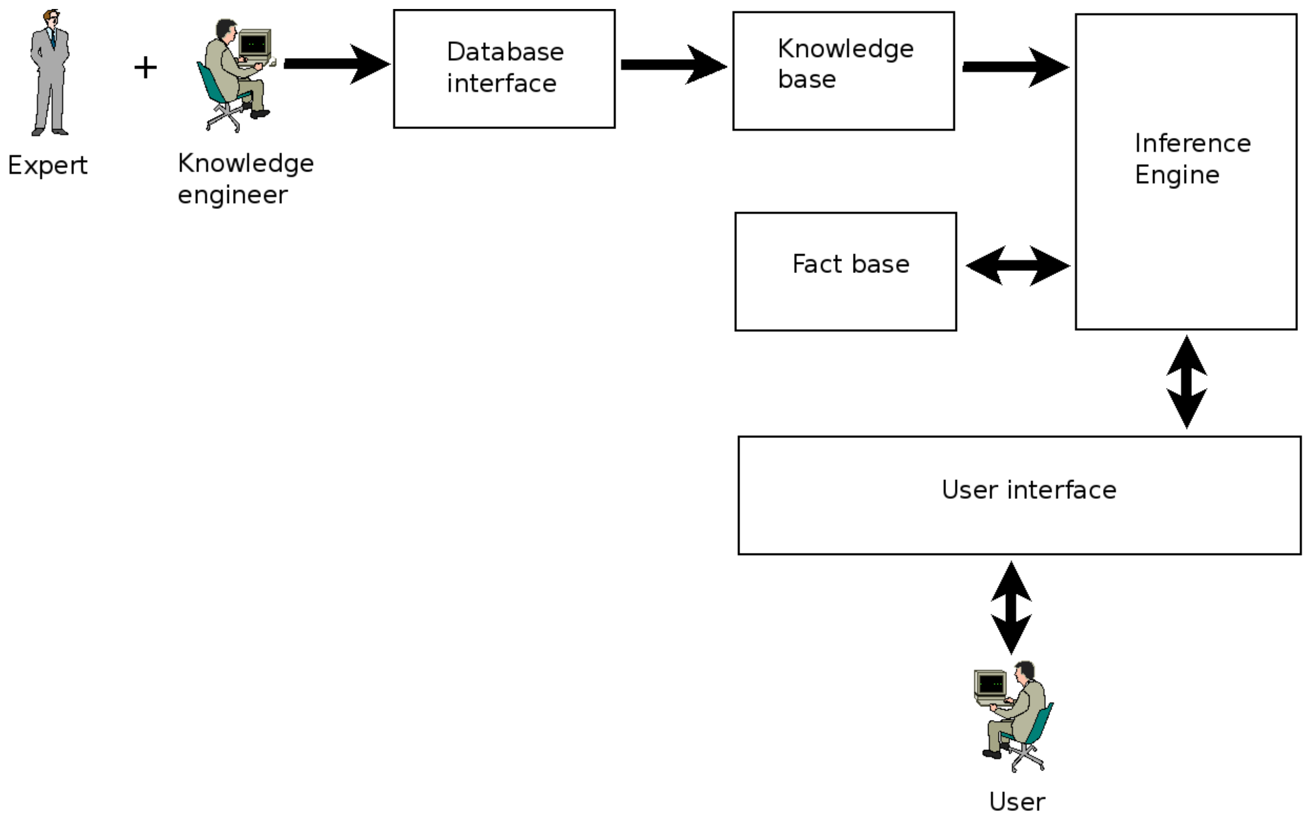
Expert systems can be classified by several factors: the structure, methods of reasoning and the type of data to be processed. Such systems can be constructed as **dedicated systems** (created from the beginning by experts and knowledge engineers or **skeleton systems**, which at the beginning have an empty knowledge base. Their inference engine creates expert opinion based on the rules of **binary logic** (that happens in the systems with rule-based knowledge representation, based on true/false values), **multi-valued logic** (as presented below – in the case of a probability-based knowledge representation in BayEx system) or **fuzzy logic** – in such systems fuzzy confidence values are assigned to the facts and reasoning rules. Expert systems are also divided into **deterministic** (the ones that process unquestionable knowledge) and **nondeterministic** or probability ones (the ones that operate on uncertain and approximated knowledge).

Let's assume that an expert system is to be applied in a hospital with its database of illnesses and symptoms that can be used by doctors. The cases of patients have been described and the sets of symptoms have been assigned to their illnesses. With the application of the information included in the database, the expert system can diagnose a distant patient who provides the doctor with the symptoms of the illness. In this case the acquisition of knowledge is conducted through the analysis of data included in the hospital database. **Knowledge acquisition** may also be carried out directly by the experts (e.g. the doctors) who provide the machine with the information in their field. The difficulty in developing such a system consists mainly in the ability to formulate the rules that constitute the basis for the experts' diagnoses – it must be possible to represent their knowledge in the system. Expert systems must also be able to acquire knowledge by learning from examples; consequently, the data are not only stored (e.g. in the form of rules given by the experts) but also added and updated on an ongoing basis by the system itself in the course of reasoning.

The architecture of expert systems includes (to make it simple) an **inference engine** (which is responsible for data processing and the provision of expert opinion), **knowledge base** (database with data acquired externally by a self-learning software and applied in the reasoning process), **fact base** with the facts obtained in advance by the software or deduced independently and recorded, and the **interface** for the interactions between the system and the humans. Figure 1 presents an outline of the expert system architecture with the consideration of the above mentioned elements.

[9] Arabas J., *Wykłady z algorytmów ewolucyjnych*, Wydawnictwa Naukowo-Techniczne, Warszawa 2001.

Fig.1. General scheme of expert system architecture



The inference engine functions on the basis of an implemented reasoning algorithm which will be presented in the article by probability reasoning and deductive and inductive reasoning. The latter is particularly common and consists in *the generalization of a unit training information with the aim to acquire general knowledge*[10].

Expert systems have been widely applied in a variety of real-life domains, for example: transportation and logistics[11], managing petrographic data and knowledge[12], and finance and accounting[13].

The main goal of the paper is to present two reasoning methods applied in expert systems: **the Bayes network** and **the rule-based reasoning**. In order to comprehend the idea of the functioning of expert systems and these two reasoning methods, two different expert systems were chosen, for

[10] Cichosz P., *op. cit.*, p. 44.

[11] Abacoumkin C., Ballis A., *Development of an expert system for the evaluation of conventional and innovative technologies in the intermodal transport area,* European Journal of Operational Research, 152 (2), 2004.

[12] Abel M., Silva L., De Ros L. Mastella L., Campbell J., Novello T., *PetroGrapher: managing petrographic data and knowledge using an intelligent database application,* Expert Systems with Applications, 26 (1), 2004.

[13] Deshmukh A., Talluru L., *A rule-based fuzzy reasoning system for assessing the risk of management fraud,* Intelligent Systems in Accounting, Finance and Management, 7 (4), 1998.

which exemplary knowledge bases were constructed. The **BayEx** system, which was elaborated by a team of scientists lead by Dr Marek Valenta from the Department of Computer Science in the AGH - University of Science and Technology, applies the Bayes' theorem in its approach to problem solving. The deductive and inductive reasoning is applied by the **CLIPS**, a system that is widely used at universities, written in the C language in the NASA laboratories in 1984. In the paper also advantages and limitations of both approaches are discussed.

1. BayEx Expert System

BayEx [8] is a shell system, which means that it is provided (unlike the dedicated systems) with an empty knowledge base that can be filled in accordance with the purpose of system. The aim of the software is to provide expert opinion and, when reasoning, it applies the Bayes' theorem. Thus, it is a system with uncertain knowledge – it is based on the probability-based knowledge representation on a given subject.

The system consists of the inference engine, probability knowledge base, fact base (working memory) and the user interface. The meta-knowledge base is an additional element which stores the information regarding, for example, the order of questions asked to the user. The inference engine calculates the probability of hypotheses (the knowledge about facts) conditioned by given symptoms (facts) that are assigned an adequate occurrence probability. The occurrence probability of a given hypothesis on the condition of particular symptoms is referred to as ***a posteriori* probability** and results from the Bayes' theorem. The system is informed about the symptoms by its user in the course of preparing the expert opinion, which is exemplified below. Moreover, the hypotheses are also assigned the ***a priori* probability,** which is determined "in advance".

In short, *the basic mechanism of probability reasoning (…) is based on the Bayes' theorem, which determines the dependence of the* a posteriori *probability of hypotheses on their* a priori *probability. The dependence defines the way how the probabilities of hypotheses change after taking the data into consideration*[14]. The expert opinion provides a list of hypotheses that are sorted starting from the most probable (with the highest degree of certainty), which is the result of the reasoning about their truthfulness, with the assumption of the truthfulness of a given set of facts (symptoms). In the course of reasoning, on the basis of new facts provided by the user, current probabilities are determined.

For the BayEx to commence reasoning, knowledge base in a given field has to be entered to the system. The reasoning process is exemplified below.

[14] *Ibidem*, pp. 309-310.

Data regarding currently used web browsers have been entered to the BayEx system so that the user should be able to choose the preferable one. When investigating (preparing the expert opinion), the BayEx system dynamically (after each reply of the user) controls the order of the questions asked. Consequently, the interaction with the system, i.e. the time necessary to obtain the answer, is as short as possible, which results in the user's satisfaction.

When entering the data to the knowledge base, the hypotheses should be placed first – together with their a priori probabilities, which in the example are the same for every hypothesis. That is because it was assumed that every hypothesis has a similar occurrence probability. Moreover, the total of the a priori probabilities of all hypotheses is very close to the value of 1 (100%), which means that the system under construction meets the assumption of completeness and finiteness of the hypotheses (the assumption is discussed below). **Table 1** presents the list of hypotheses (the names of web browsers) in the knowledge base.

Tab. 1. Hypotheses and their a priori probabilities

WEB BROWSER	A PRIORI PROBABILITY
Flock	0.083333
Google Chrome	0.083333
Internet Explorer	0.083333
Konqueror	0.083333
Lunascape	0.083333
Lynx	0.083333
Maxthon	0.083333
Midori	0.083333
Mozilla Firefox	0.083333
Opera	0.083333
Safari	0.083333
SeaMonkey	0.083333

Then, their symptoms should be determined as well as their relationships with the hypotheses **(See Table 2).** The number of symptoms usually exceeds the number of hypotheses – and that is the case in the example provided.

Tab. 2. Relationships of the "Opera" hypothesis with particular symptoms - a fragment of the list

SYMPTOM	RELATIONSHIP WITH HYPOTHESIS	
	P1	P2
2.11. Linux	1.000000	0.636364
3.12. Windows	1.000000	0.909091
4.13. Mac OS X	1.000000	0.727273
5.14. BSD	1.000000	0.545455
6.21. Auto updater	0.030000	0.024545
7.22. Download Management	0.010000	0.009091
8.23. Form Managing	0.008000	0.007273
9.24. Password Managing	0.020000	0.016364
10.25. Privacy Mode	0.030000	0.016364
11.26. Spell checking	0.015000	0.010909
12.31. Ad filtering	0.035000	0.028636
14.33. Full-text search of history	0.005000	0.001818
15.34. Mouse Gestures	0.010000	0.002727
16.35. Page Zooming	0.010000	0.008182
17.36. Pop-up	0.040000	0.032727
18.37. Spatial Navigation	0.005000	0.002273
19.38. Tabbed Browsing	0.030000	0.027273
20.39. Tabbing Navigation	0.005000	0.004091
21.39. Text-to-speech	0.005000	0.000455

22.39. Voice Control	0.005000	0.001364
23.41. MathML	0.005000	0.001818
26.51. Atom Feed	0.015000	0.009545
27.52. RSS	0.020000	0.012727

It can be concluded from the list in **Tab. 2.** that *Opera* web browser is accessible to four operating systems and has all listed functions (symptoms 6.21 and downwards). Probability P1 indicates the occurrence of a particular symptom if the hypothesis is true. Probability P2 (the occurrence of a particular symptom when the hypothesis is false) is calculated by BayEx after all P1 probabilities are defined.

Tab. 3. Relationships of the "Konqueror" hypothesis with particular symptoms

SYMPTOM	RELATIONSHIP WITH HYPOTHESIS	
	P1	P2
2.11. Linux	1.000000	0.636364
4.13. Mac OS X	1.000000	0.727273
5.14. BSD	1.000000	0.545455
6.21. Auto updater	0.030000	0.024545
7.22. Download Management	0.010000	0.009091
8.23. Form Managing	0.008000	0.007273
9.24. Password Managing	0.020000	0.016364
11.26. Spell checking	0.015000	0.010909
12.31. Ad filtering	0.035000	0.028636
13.32. Caret Navigation	0.010000	0.006364
16.35. Page Zooming	0.010000	0.008182
17.36. Pop-up	0.040000	0.032727
18.37. Spatial Navigation	0.005000	0.002273

19.38. Tabbed Browsing	0.030000	0.027273
20.39. Tabbing Navigation	0.005000	0.004091
26.51. Atom Feed	0.015000	0.009545
27.52. RSS	0.020000	0.012727
28.61. BitTorrent	0.020000	0.001818
31.71. Polish	0.200000	0.127273
33.73. Open Source	0.010000	0.005455

For the sake of comparison, **Tab. 3.** presents symptoms related to other web browser. It is clear that every hypothesis is related to several symptoms and – vice versa – every symptom is related to several hypotheses. Columns P2 include similar probabilities for all hypotheses (Tab. 2 and Tab. 3). Columns P1 also have the same values of symptoms for different hypotheses. That results from the consideration of a particular function when choosing a web browser. The examples show that users, when making their choices as regards web browsers, take mainly the operating system into consideration. Such features as the use of Polish language by the software or pop-up blockers (the Pop-up symptom) are of secondary importance. The least important characteristics for the users include such symptoms as Form Managing and Voice Control. Thus, the higher probability P1, the more certain it is that the related hypothesis will rank high in expert opinions (on the condition that the user will answer affirmatively the questions on the symptoms related to this hypothesis)

Once the knowledge base includes the hypotheses and symptoms, the meta-knowledge of the system should be defined. That procedure consists in entering Yes/No questions, followed by specific questions that depend on the answer. For example, an expert system for medical diagnosis should have in its meta-knowledge a question whether the patient had a blood test. If the answer is negative, all the symptoms related to blood testing will be assigned the value: *I do not know,* and the system stops asking questions about the details of the test. If the answer is affirmative, the system asks more detailed questions concerning the blood test. It is worth mentioning that there is another case of a negative answer to the system's question about a general symptom, e.g. *Has recently a muscle pain occurred?* All related questions about symptoms (e.g. "*Has recently muscle pain occurred in the back?")* may be assigned the answer : *No*, without asking further questions.

A well constructed expert system should have a thoroughly considered meta knowledge, which will make it look "intelligent". The system should also include, apart from the questions to be asked, textual commentaries and explanations to the questions. Moreover, it should monitor what questions have been asked to avoid the repetition. It is also very important that the condition for the completion of the asking procedure is specified so that it is not too time-consuming and the results are not disproportionate to the time devoted.

Figure 2 presents the results of the work of the users (called the test objects) with the BayEx system. The expert opinion regarding the choice of the Internet browser was printed with the use of a system built-in function.

Fig.2. Example of an expert opinion in the BayEx system: questions and answers

Expert opinion of: Maciek

Knowledge base: przegladarki.bay

List of questions and answers:

1. Are you interested in selecting an operating system? Yes
2. Are you interested in selecting browser features (see commentary)? Yes
3. Are you interested in selecting accessibility features (see commentary)? Yes
4. Are you interested in selecting a new web technology support (see commentary) ? No
5. Are you interested in selecting syndicated content support (see commentary) ? Yes
6. Are you interested in selecting a built-in additional protocol support (see commentary)? Yes
7. Do you want your browser to be a text browser? No
8. Should your browser be Windows-based? Yes
9. Should your browser be Mac OS X – based? I don't know
10. Should your browser be Linux – based? Yes
11. Should your browser be BSD – based? I don't know
12. Should your browser have an in-built e-mail client? No
13. Should your browser have an in-built IRC client? No
14. Should your browser have a text to speech? No
15. Should your browser have an in-built torrent client? No
16. Should your browser have a service pack? Yes
17. Should your browser have voice control? No
18. Should your browser have mouse gestures? No
19. Should your browser have full text history search ? No
20. Should your browser have spatial navigation? No

21. Should your browser be accessible in Polish? Yes
22. Should your browser sources be open? Yes
23. Should your browser have RSS? Yes
24. Should your browser have Atom Feed? I don’t know
25. Should your browser allow caret navigation? No
26. Should your browser have spell check? Yes
27. Should your browser allow tab navigation? Yes
28. Should your browser have advert filter? Yes
29. Should your browser have Pop-up blocker? Yes
30. Should your browser have auto-update? Yes
31. Should your browser have page zooming? I don’t know
32. Should your browser have a password manager? Yes
33. Should your browser allow tabbed browsing? Yes
34. Should your browser have a download manager? Yes
35. Should your browser have a form manager? I don’t know

List of hypotheses sorted by their current probabilities:

1. 0.957663 Mozilla Firefox
2. 0.956109 SeaMonkey
3. 0.903475 Google Chrome
4. 0.851198 Opera
5. 0.762832 Flock
6. 0.741127 Konqueror
7. 0.671184 Maxthon
8. 0.565412 Internet Explorer
9. 0.479735 Safari
10. 0.414813 Midori
11. 0.210243 Lunascape
12. 0.000000 Lynx

As Fig.2. shows, the numbered questions are followed by the users' answers. First, the system asks the user questions in the area of the meta-knowledge (questions 1 – 7) in order to decide what other questions should be asked afterwards. As the user prefers a particular operating system, questions 8 – 11 aim at the determination what system he/she has in mind. Having asked 35 questions (including the Yes/No questions) the system generates a list of hypotheses that constitute a list of browsers to be suggested to the user.

Probability-based reasoning has several advantages**: simplicity of knowledge representation** and the **possibility to model uncertain knowledge** (i.e. the real one that we encounter on a daily basis). However, the drawbacks of the probability-based expert systems are as follows: the reliance on the users' True/False answers (without the possibility to represent user's ignorance[15]), frequently wrong order of the questions asked and the following three limitations of the system as regards the knowledge base under construction. The first limitation concerns the set of hypotheses in the system knowledge base – such a system has to be **finite and complete**, which means that it should include all possible solutions to a problem. Another condition for a correct expert opinion is the mutual independence of hypotheses in the knowledge base (as a result, when building the system knowledge base, the expert has to construct a set of hypotheses in such a way that it does not include statements dependent on each other, e.g. B-type hepatitis and hepatitis as separate hypotheses). The last limitation of the construction of that type of expert systems is the necessity to include by a given hypothesis a set of symptoms that are related to it, which results from the fact that the **symptoms within a hypothesis have to be conditionally independent from one another**. It is quite clear that the above mentioned limitations are often difficult or impossible to overcome. Consequently, expert system reasoning that is based on probability knowledge representation is not free of errors. However, expert systems should be used as the algorithm applied *reaches sometimes the error level approximate to the one that is offered by far more advanced and costly algorithms*[16].

2. CLIPS Expert System

CLIPS [3] includes both an inference engine and a rule editor. Thus, it constitutes a complete environment for developing, testing and running expert systems. It is commonly used by universities, it is free of charge and has an extensive documentation. Moreover, it is an extended

[15] BayEx system makes it possible for its user to give the following answers to questions asked: *Yes; No; I don't know; Yes, to some extent; No, to some extent.* The answer *I don't know* does not provide the expert opinion with any information, while the last two give a 50% certainty. Thus, BayEx aims at giving the user a chance to give the answer which is as close to truth as possible.

[16] *Ibidem*, p. 310.

tool set – allows for rule-based, object-oriented and procedural representation of knowledge. What is more, it can be installed in various operating systems.

The architecture of the system includes: an inference engine, knowledge base, fact base and the user interface. Into the knowledge base only true rules of **IF [premise] –Then [conclusion]** scheme can be entered (the truthfulness of the conclusion results from the truthfulness of the premise). The inference engine, when it knows the rules in the knowledge base and knows that the condition is true, enters into the fact base a new fact that informs about the truthfulness of the conclusion. Thus, CLIPS applies forward chaining (***Modus Ponendo Ponens***) – on the basis of presented facts it constructs a new logical statement [17].

Forward chaining (it is data-driven method as opposed to goal-driven backward chaining) consists in checking by the system - on the basis of the knowledge included in the knowledge base - whether the facts added by the user to the fact base are true. CLIPS that provides expert opinions with the use of the rule-based knowledge representation should possess in its fact base the facts that exist in the rules of the knowledge base since a given rule cannot be used in reasoning if a given fact does not exist in the fact base.

Before moving on to an example of an expert opinion, let's look how forward chaining is conducted in the system. CLIPS constructs a set of conflicts that includes rules whose conditions are met (i.e. the ones whose facts are known). Then, the solution of the set of conflicts is conducted by selecting a given rule from the set of conflicts and then firing it. The firing of the rule consists in substituting the facts from the fact base to the rule and checking whether the premise of the rule (which depends on the fact or the facts) is true. After that a new fact is entered to the system fact base and, as a result, a new set of conflicts is constructed. If - after firing the rule - it turns out that the premise is false, another rule is chosen from the constant set of conflicts (thus, the rule that has been checked is removed from the set of conflicts). As it can be expected, the reasoning is finished when there are no more rules in the set of conflicts and the premise of the last rule that has been fired is false.

It should be added that in the systems with rule-based knowledge representation there are different strategies of designating the rules from the set or solving the set of conflicts. Subsequent rules from the set are selected by means of strategies that are domain-dependent or independent. **Domain-dependent strategies** have additional elements in their knowledge base that control the data that have been assigned to the rules by the knowledge expert (e.g. arbitrarily assigned weights to the rules). **Domain-independent strategies** are not influenced by the expert and may accept the following forms: *Apply any rule from the set, First apply long rules* (the ones with many facts), *First*

[17] Backward chaining (goal-driven) is also possible. However, it is not the subject of the article.

use short rules (the ones with few facts), *Always use the eldest rule in the set* or *Use the rule whose fact was deduced last.*

Figure 3 presents an example of a code that is recognized by CLIPS.

Fig. 3. CLIPS code fragment

```
(defrule q0
=>
(printout t "Answer t (yes) or n (no). Is it clear? >")
(assert (q0 (read))))

(defrule q0_odp
(and (q0 ~t))
=>
(printout t "User answered: no. Work finished”  crlf))

(defrule q9
(and (q8 t))
=>
(printout t "Is the Internet cable plugged-in tightly? >")
(assert (q9 (read))))

(defrule q9_odp
(and (q9 n))
=>
(printout t "Plug in the cable tightly crlf))

(defrule q10
(and (q9 t))
=>
(printout t "Is the Internet cable broken or damaged along its length? >")
(assert (q10 (read))))

(defrule q10_odp
(and (q10 n))
=>
(printout t "Contact ISP, exchange the cable." crlf))
```

As the example in Figure 3 shows, question q10 should appear if the user answers affirmatively to q9 (“Yes, the Internet cable is plugged in tightly”). If the user answers *No* to q9, the reply is “Plug in the cable tightly”. As it can be seen in Figure 4, the questions follow one another, depending on the answers given.

Fig. 4. Content of the CLIPS dialog window

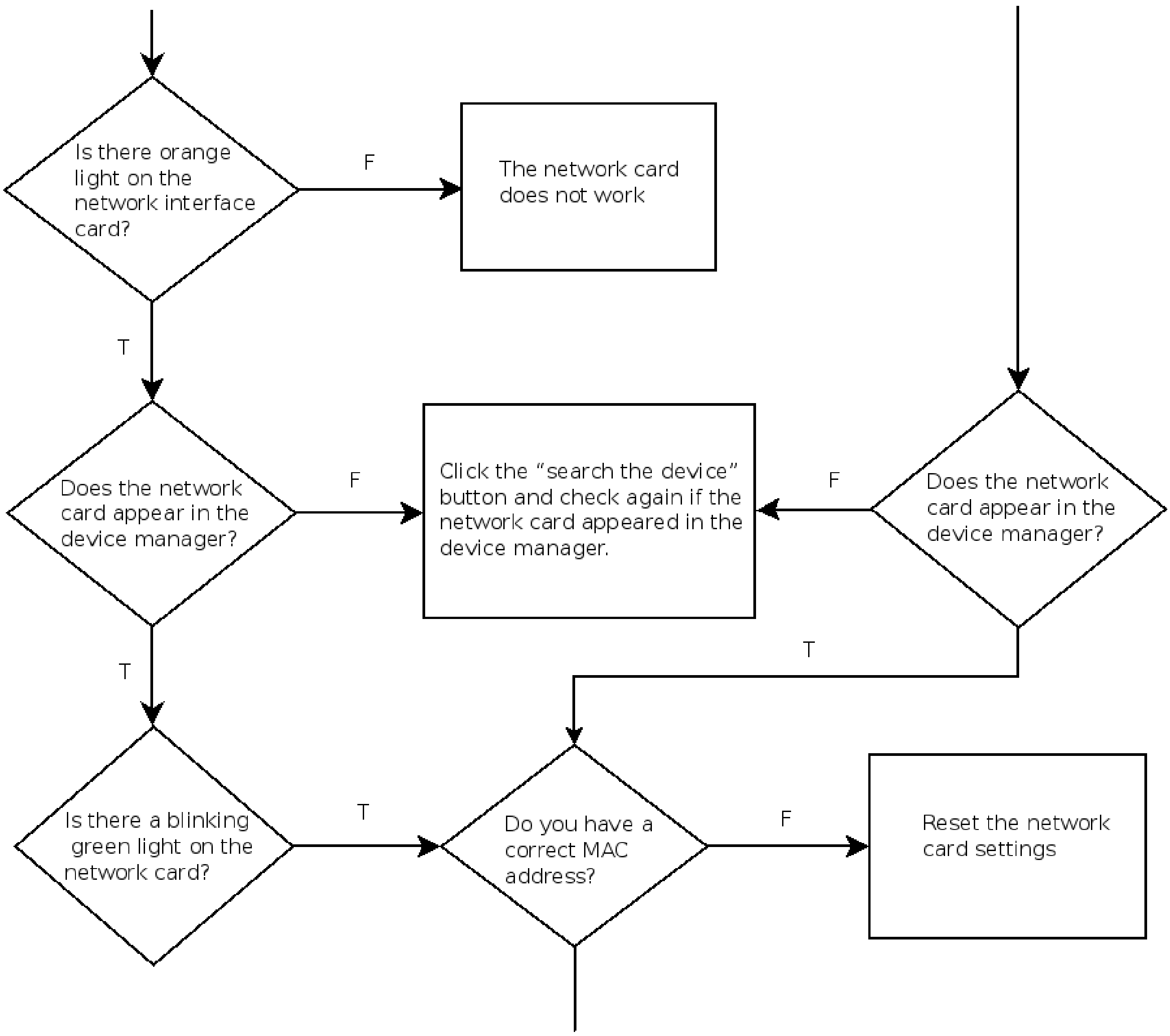
```
==> f-2     (q1 t)
FIRE     3   q2: f-2
Have you subscribed to the Internet? >t
```

```
==> f-3     (q2 t)
FIRE     4   q3: f-3
Is the computer on? >t
==> f-4     (q3 t)
FIRE     5   q4: f-4
Is the computer monitor on? >t
==> f-5     (q4 t)
FIRE     6   q5: f-5
Have you paid for the Internet? >t
==> f-6     (q5 t)
FIRE     7   q6: f-6
Have you reset the computer? >t
==> f-7     (q6 t)
FIRE     8   q7: f-7
Has the Internet functioned until today? >t
==> f-8     (q7 t)
FIRE     9   q8: f-8
Has the Internet cable been plugged in to the right computer port? >t
==> f-9     (q8 t)
FIRE     10   q9: f-9
Has the Internet cable been plugged in tightly? >n
==> f-10     (q9 t)
FIRE     11   q9_odp: f-10
Plug in the cable tightly.
```

Figure 4 presents the content of the CLIPS user interface window. The user answers the questions with either *t* (yes) or *n* (no) – the only two options. As it can be seen above, every answer is followed by a further question or the final result of the expert opinion shows on the screen.

A fragment of a decision tree of an expert system that analyzes the problems with the Internet connection is given in Figure 5.

Fig. 5. Fragment of a decision tree in CLIPS



It has to be pointed out that the user is limited as regards his/her answer options. The main problem in using a rule-based system is the possibility to represent the reality only with the use of two-value logic – only certain knowledge can be represented. The system can only conclude about the truthfulness of particular facts or express its lack of knowledge about them. Thus, linguistic possibilities to represent the knowledge are poor and do not model the reality in a complete way, which may seem to the users of such expert systems both useless and unfriendly in solving their problems.

The modifications of the rule-based knowledge representation in expert systems that aim at making the representation closer to reality may consist in the improvements that express the uncertainty of knowledge. In the case of the probability-based knowledge representation (as it was presented in the BayEx system) the uncertainty is expressed by the occurrence probabilities of a symptom together with a given hypothesis. Here, the uncertainty of knowledge can be propagated by adding to the rules the certainty degree as regards their truthfulness. The methods of Dempster-Shafer[18] and Zadeh - the application of **fuzzy set**[19] based knowledge representation - embed adequate coefficients to the rules.

Despite obvious drawbacks, the rule-based knowledge representation has its advantages – the intuitiveness and clarity of rules. According to Cichosz: *Among all methods of knowledge representation that are used in machine learning, there is no other method as close to the methods of recording knowledge by humans as the rule-based method. A rule, consisting of the if- part and the then- part, provides a decision adequate to the situation where conditions are met; it can be noted as: IF conditions, THEN decision. That makes the rule-based method so popular; it is considered the clearest to humans and is applied in inductive learning as willingly as the decision trees*[20].

Summary and Conclusions

In this paper we presented and compared two types of reasoning methods applied in expert systems: the Bayes network and the rule-based reasoning. The comparison was illustrated by two exemplary expert systems realized with the use of BayEx and CLIPS.

In conclusion, one should point out to such significant advantages of experts systems as the possibility to document reasoning and apply various fact bases in a simple way as well as the comparatively low costs of presenting expert opinions. However, one should keep in mind the disadvantages (in comparison to human reasoning) that have not been eliminated yet: a narrow range of expert opinions, the necessity to operate on data that come from devices and not from senses, the lack of common sense and, infrequently, the triviality of conclusions.

An expert system can learn to play chess well (when given a particular set of the game rules) and even win with a human, as the IBM's *Deep Blue*, or be a winner in TV quiz with a particular way of answering questions as *Watson* (IBM) in *Jeopardy.* Moreover, such a system may rescue one's life by making a right medical diagnosis that is based on correct reasoning or by predicting an earthquake.

Nowadays, expert systems steer unmanned vehicles and bomb-dismantling robots, search complex data bases of scientific investigations and various surveys with the aim to find repetitive

[18]The explanation of the Dempster-Shafer method is included in the following papers: Shenoy Prakash P., *Using Dempster-Shafer's Belief-Function Theory in Expert Systems*, in volume edited by Yager R. R., Federizzi M., and Kacprzyk J., *Advances in the Dempster-Shafer Theory of Evidence*, John Wiley & Sons, New York 1994, pp. 395–414 and Sentz K., Ferson S., *Combination of Evidence in Dempster-Shafer Theory*, in Technical Report No. SAND 2002-0835, 2002.

[19] The theory of fuzzy sets is described in the book by L. Rutkowski, *Metody i techniki sztucznej inteligencji*, PWN, Warszawa 2005, p. 121 sqq.

[20] P. Cichosz, *op. cit.*, p. 195.

patterns (the area that deals with - among others - the application of AI to look for knowledge is called *Data Mining*). They are also used in constructing computer games and when granting bank credits. AI is indispensable in space probes and the recognition of human speech. Thus, the range of the applications of expert systems in present-day world is vast. Undoubtedly, the development of technologies will be accompanied by further progress in expert systems. Continual development in this field is the promise of the access to the most precious thing – the information and the resulting knowledge.

Bibliography

[1] Abacoumkin C., Ballis A., *Development of an expert system for the evaluation of conventional and innovative technologies in the intermodal transport area,* European Journal of Operational Research, 152 (2), 2004.

[2] Abel M., Silva L., De Ros L. Mastella L., Campbell J., Novello T., *PetroGrapher: managing petrographic data and knowledge using an intelligent database application,* Expert Systems with Applications, 26 (1), 2004.

[3] Arabas J., *Wykłady z algorytmów ewolucyjnych*, Wydawnictwa Naukowo-Techniczne, Warszawa 2001.

[4] Cichosz P., *Systemy uczące się*, Wydawnictwa Naukowo-Techniczne, Warszawa 2000.

[5] Deshmukh A., Talluru L., *A rule-based fuzzy reasoning system for assessing the risk of management fraud,* Intelligent Systems in Accounting, Finance and Management, 7 (4), 1998.

[6] CLIPS Online Documentation, http://clipsrules.sourceforge.net/OnlineDocs.html.

[7] Kasperski M. J., *Sztuczna inteligencja*, Wydawnictwo Helion, Gliwice 2003.

[8] Rożek T., *Nauka po prostu. Wywiady z wybitnymi*, Wyd. Demart SA, Warszawa 2011.

[9] Rutkowski L., *Metody i techniki sztucznej inteligencji*, PWN, Warszawa 2005.

[10] Sentz K., Ferson S., *Combination of Evidence in Dempster-Shafer Theory*, in: Technical Report No. SAND 2002-0835, 2002.

[11] Shenoy Prakash P., *Using Dempster-Shafer's Belief-Function Theory in Expert Systems*, in: Yager R. R., Federizzi M., and Kacprzyk J., *Advances in the Dempster-Shafer Theory of Evidence*, John Wiley & Sons, New York 1994.

[12] Tadeusiewicz R., *Sieci neuronowe*, Akademicka Oficyna Wydaw. RM, Warszawa 1993, in: http://winntbg.bg.agh.edu.pl/skrypty/0001/.

[13] Tadeusiewicz R., *Tryumf czy kapitulacja rozumu?*, in: „Znak" 9 (484)/1995.

[14] Valenta, M., et al., *System ekspertowy BayEx ver. 2.4. Instrukcja użytkownika*, Department of Computer Science, AGH, 2010.

Abstract

The article presents two reasoning methods applied in the subarea of Artificial Intelligence – a field of science on expert systems: the Bayes network and the rule-based reasoning. The principles of developing expert systems in the two approaches are given in examples based on BayEx and CLIPS.

The article includes brief definitions of Artificial Intelligence and expert systems, which should encourage the reader to explore this field of science. It discusses in a clear way the construction and the aims of expert systems and the functioning principles of particular algorithms that constitute the fundamentals for reasoning, i.e. an independent method of finding solutions to problems on the basis of data received. The examples present several applications of expert systems in the present-day science and practical life. The authors agree with Prof. Ryszard Tadeusiewicz's words: *One's own intelligence is better than the artificial one, yet artificial one is better than none*[21].

[21] Tadeusiewicz R., *Tryumf czy kapitulacja rozumu?*, in: „Znak" 9 (484)/1995, p. 59.