ZESZYTY NAUKOWE AKADEMII MARYNARKI WOJENNEJ
SCIENTIFIC JOURNAL OF POLISH NAVAL ACADEMY

2015 (LVI) 2 (201)

DOI: 10.5604/0860889X.1172060

Piotr Golański[1], Przemysław Mądrzycki[1]

USE OF THE EXPERT METHODS IN COMPUTER BASED MAINTENANCE SUPPORT OF THE M-28 AIRCRAFT

ABSTRACT

This article deals with the issues relating to the maintenance and diagnostics of airplanes. It presents the concept of using the expert system to support the necessary maintenance and diagnostics of the aircraft. In the beginning it describes the general principles of developing the expert systems. It briefly discusses the basic tools used and their implementation. It also presents a way of employing one of the tools, i.e. the CLIPS expert system shell for solving airplane maintenance problems. Here the technical state diagnostics simulator of the M-28 airplane is used as an example.

Key words:
aircraft maintenance, expert systems, mobile systems.

INTRODUCTION

Since their origin computer-aided maintenance systems have employed methods based on artificial intelligence, and from the start they have been known as expert systems [1, 10]. The basic difference between traditional and expert systems lies in the method of dealing with the problem. In the traditional approach the problem is solved with an algorithm which processes the data. In the expert system the problem is described and solved exclusively with appropriately prepared data structures. Therefore, tools used to build the expert systems are somewhat different to tools used to build conventional systems. In the case of conventional systems imperative languages based on the languages C and Pascal are used. To build expert

[1] Air Force Institute of Technology, Księcia Bolesława 6 Str., 01-494 Warszawa, Poland; e-mail: {piotr.golanski; przemyslaw.madrzycki}@itwl.pl

systems declarative languages are used, such as Lisp [8] or Prolog [5]. In practice, to build Expert systems, specialized tools are employed — expert system shells. These are, in fact, really expert system without knowledge.

This article presents the use of one of the expert system shells for building an airplane support system. It also discusses the principles underlying the construction and performance of such a system. In the concluding part it presents the use of the expert methods with regard to a simple diagnostic problem, employing the M-28 airplane diagnostic simulator, designed to train ground personnel. The simulator was developed in the Air Force Institute of Technology in cooperation with the Air Force Officers Higher School, within the framework of the project UDA-POIG.01.03.01-00-201/09-00 'Development of, and studies on, an airplane diagnostic simulator in virtual technology' [11].

PROBLEM ANALYSIS

The expert systems are systems employed to manage operations models of technical object operations processes. The model of such a process, which would simultaneously reflect the way of the expert system performance, can be expressed in the form of finite state machine [7].

$$M = (Q, \Sigma, \delta, q_0, F), \tag{1}$$

where:

Q — finite state set;

Σ — finite input alphabet;

q_0 — initial state ($q_0 \in Q$);

δ — transition function $\delta : Q \times \Sigma \to Q$;

F — final state set ($F \subset Q$).

The finite state set defines the knowledge states in the expert system relating to the operated object. The initial finite alphabet is a set of symbols which allow for a dialogue between the user and the system. The transition function δ , depending on the current state and the results of the user-system dialogue, changes the knowledge state in the system. The final state set determines the knowledge state which contains the solution to the problem.

The presented model (1) also describes the basic elements and performance of the expert system as well. In the literature the expert systems are described in various aspects and at various levels of detail. This article has adopted the account of the expert system structure presented in the work [12].

In this approach the expert system is composed of four parts (fig. 1):

- knowledge base — declarative representation of knowledge often in the form of *if-then*;
- working memory — data specific for the problem being solved;
- inference engine — code which is the core of the system which makes inferences based on the knowledge base and data from the cache;
- user interface — code controlling the dialogue between the user and the system.

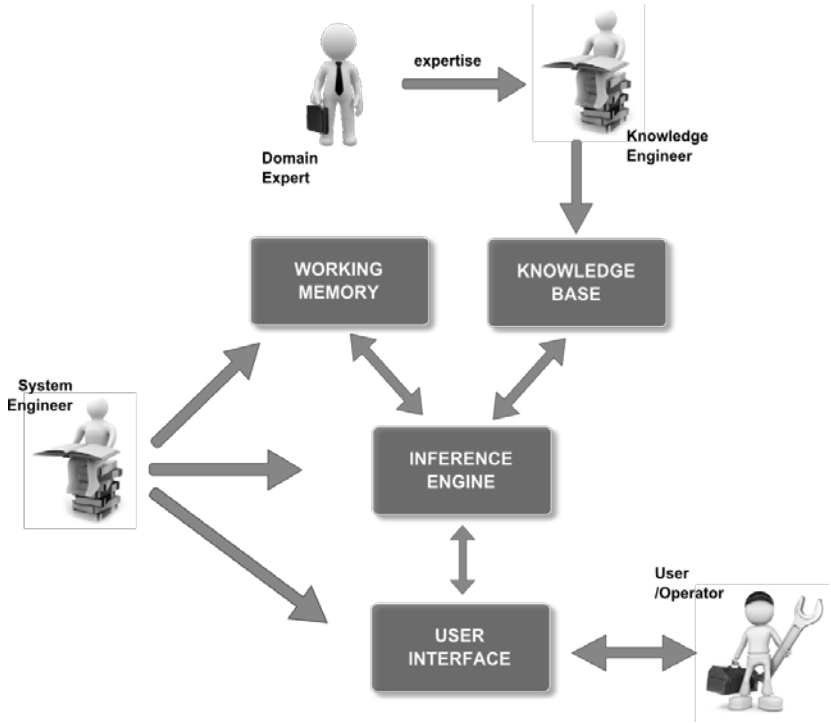

Fig. 1. A diagram of the expert system structure

The key elements in every expert system are the set of data declarations dependent on the problem, called base knowledge and the program independent of the problem (although dependent on the data structure) called inference engine. The knowledge base is the expert's domain knowledge, and it is implemented by the

knowledge engineer as the initial state of knowledge relating to the object q_0. The inference engine operates on the base knowledge and in the dialogue with the operator it generates a solution to the problem defined by the user, realizing the transition function δ. In order to interact with the system the user employs the interface, most often Graphic User Interface type, using the elements of the finite alphabet — Σ.

Inference engines work in one of two modes: forward chaining and backward chaining. Forward chaining is the top-down method which using the known facts attempts to make inferences (from the satisfied rule conditions), which are then performed as actions. This process continues until the goals are reached. Backward chaining is the bottom-up method which starts with goals, and searches the data base until it finds the rules whose actions match the expected goal. If the condition of such a rule is not satisfied, it is added to the list of goals. Backward chaining is more like a verification process than a search process and it is usually used in technical diagnostics systems [5].

IMPLEMENTATION AND PERFORMANCE OF THE SYSTEM

In order to implement the inference engine and the knowledge base, in the case of the system under consideration, CLIPS (C *Language Integrated Production System*) was used, the tool for building expert systems in National Aeronautics and Space Administration (NASA). A CLIPS was widely used in the past and is widely used at present for building expert systems relating to technical diagnostics: [2, 4, 9, 13].

The performance of the CLIPS's interpreter is based on using the inference mechanisms contained in its inference engine and the set of rules and facts written in the file m-28.clp. The rules have the form of implication:

$$\text{LHS} \Rightarrow \text{RHS}, \tag{2}$$

where:
LHS — Left Hand Side;
RHS — Right Hand Side.

As a result of the work of the inference engine the rules whose LHS match the facts placed in the cache are selected. Then, the order of triggering rules is determined and stored in the cache. In the next step the rules are triggered following their priority, and then actions contained in their RHS are performed. This way the inference engine performs forward chaining.

In the case of the discussed system, in the initial state q_0 is triggered by the no-goal rule (fig. 2), as in the cache after it is entered from the file m-28.clp there is no defined action goal, i.e. the maintenance action has not been selected. Execution of the RHS of this rule leads to waiting for the selection of action.

```
(defrule no-goal ""
  (not (goal $?))

  =>

  (format t "read new goal ")
  (assert(goal is (cel))))
```

Fig. 2. The rule of maintenance action selection

After selecting the action, i.e. determining the action goal, the inference engine looks for indirect goals which ensure execution of the selected action, triggering the propagate-goal rule (fig. 3). If, for example, the action identified with the goal pg44.diag (diagnostics of low fuel warning light) has been selected then the facts shown in figure 4 will match the propagate-goal rule. As a result of the execution of RHS of the rule two goals pg44.test and PG44 will be determined. As it can be seen, despite the fact that the inference engine CLIPS performs the forward chaining inference, appropriately formulated facts allow for executing back chaining inference and the goals which occur in the LHS of the rules are determined.

```
(defrule propagate-goal ""
   (goal is ?goal)
   (rule (if ?variable $?)
         (then ?goal ? ?value))
   =>
   (assert (goal is ?variable)))
```

Fig. 3. The rule of goal propagation

```
(rule (if pg44.test is wl and
          PG44 is wyl)
      (then pg44.diag is blown))


(rule (if pg44.test is wl and
          PG44 is wl)
      (then pg44.diag is functional))
```

Fig. 4. Facts containing propagate goal pg44.diag

Repeated triggering of the propagate-goal rule leads to determining goals and initial actions. After reaching this state, dialogue rules, checking if the goals are reached, begin to be triggered. Figure 5 presents the ask-question-leg l values-and--instruct rule generating the questions 'does the variable have a value?'. This question is transmitted to the user by means of the interface. In the case of a negative answer the function message is invoked which is also used to transmit the answer as a text message.

```
(defrule ask-question-legalvalues-and-instruct ""
   (declare (salience 15))
   (legalanswers ? $?answers)
   ?f1 <- (goal is ?variable)
   ?f2 <- (question ?variable ? ?text)
   ?f3 <- (instruction ?variable ? ?text_instr)
   =>
   (retract ?f1)
   (bind ?reply (test ?text))
   (printout t ?answers)

   (if (member (lowcase ?reply) ?answers)
     then (if (eq ?reply no)
            then (message ?text_instr)
                 (assert(goal is ?variable))
            else (assert (variable ?variable ON))
                 (retract ?f2)
                 (retract ?f3))
     else (stop ?variable) (halt)  ))
```

Fig. 5. An example of dialogue rule

CONCLUSIONS

The aim of this article is to present the concept of using expert methods for M-28 aircraft maintenance. The concept is based on using CLIPS expert system shell.

This solution has been tested on simple maintenance procedures of M-28 aircraft. The positive result of the test indicates that after appropriate extension of the knowledge base it would be possible to build a complete diagnostic expert system for this aircraft.

In addition, such a system, following necessary modifications, could be used for diagnostics of other aircrafts as well as other technical objects.

REFERENCES

[1] Barker V. E., O'Connor D. E., *'Expert' systems for configuration at Digital: XCON and beyond*, 'Communications of the ACM', 1989, Vol. 32, No. 3, pp. 298–318.

[2] CLIPS '94, *Third Conference on CLIPS Proceedings (Electronic Version)*, September 12–14, 1994, Lyndon B. Johnson Space Center, [online], http://clipsrules.sourceforge.net/documentation/other/3CCP, pdf, [access 16.03.2015].

[3] Dapeng T., Li Peiyu L., Pan Xiaohong P., *Embedded Fault Diagnosis Expert System Based on CLIPS and ANN*, 'Computational Science — ICCS 2007 Lecture Notes in Computer Science', 2007, Vol. 4490, pp. 957–960.

[4] Gartner D., Sheppard J., W., *An Experiment in Encapsulation in System Diagnosis*, Test Technology and Commercialization — Conference Record, AUTOTESTCON '96, pp. 468–472.

[5] Gatnar E., Stąpor K., *Prolog — język sztucznej inteligencji*, Wyd. PLJ, Warszawa 1991 *[Prolog — language of artificial intelligence* — available in the Polish language].

[6] Giarratano J. C., *CLIPS — User's Guide*, 2002.

[7] Homenda W., *Elementy lingwistyki matematycznej i teorii automatów*, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2005 [*Elements of mathematical linguistics and theory of automated machines* — available in the Polish language].

[8] Jurkiewicz Z., Lao M. J., *Język programowania LISP*, PWN, Warszawa 1990 [*The programming language LISP* — available in the Polish language].

[9] Long H., Wang X., *Aircraft fuel system diagnostic fault detection through expert system*, Proc. IEEE Sixth World Congress on Intelligent Control and Automation, WCICA, 2008 pp. 7104–7107.

[10] Mądrzycki P. at all, *Diagnostic Simulator of the M-28 Aircraft for the Ground Engineering Crew in Virtual Technology*, 'Polish Journal of Environmental Studies', 2011, Vol. 20, No. 5A.

[11] Mazurkiewicz D., *Computer-aided maintenance and reliability management systems for conveyor belts*, 'Eksploatacja i Niezawodność — Maintenance and Reliability', 2014, Vol. 16, No. 3, pp. 377–382.

[12] Merritt D., *Building Expert Systems in Prolog*, Amzi! inc., USA, 2000.

[13] Yan C., Ma S., Zhou G., Fang J., *Fault Diagnostic Expert System of Rolling Element Bearing Based on CLIPS*, 'Journal of Information and Computational Science', 2013, Vol. 10, No. 10, pp. 3053–3062 [available online], http://www.joics.com, pdf, [access 16.03.2015].

ZASTOSOWANIE METOD EKSPERCKICH W KOMPUTEROWYM SYSTEMIE WSPOMAGANIA OBSŁUGI SAMOLOTU M-28

STRESZCZENIE

Artykuł dotyczy problematyki wspomagania procesu obsługi i diagnostyki statków powietrznych. Przedstawiono w nim koncepcję wykorzystania systemu ekspertowego do wspomagania obsługi i diagnostyki statku powietrznego. Na wstępie przedstawiono ogólne zasady budowy systemów ekspertowych. Krótko omówiono podstawowe narzędzie służące do ich implementacji. Przedstawiono sposób wykorzystania jednego z narzędzi, jakim jest powłoka systemów ekspertowych CLIPS, do rozwiązania problemów obsługowych statku powietrznego na przykładzie symulatora diagnostycznego samolotu M-28.

Słowa kluczowe:

obsługa statku powietrznego, systemy ekspertowe, systemy mobilne.