



Selection of the Optimal Software for Designing Expert Systems

Mukhoid O. V.^{*}, Kostornoi O.S., Shyfrin D. M.

Research and Design Institute for Atomic and Power Pumpbuilding JSC “VNIAEN”, 2 2nd Zaliznychna St., 40003 Sumy, Ukraine

Article info:

Paper received:

June 8, 2018

The final version of the paper received:

August 21, 2018

Paper accepted online:

August 25, 2018

*Corresponding Author's Address:

sapr@vniiaen.sumy.ua

Abstract. This article deals with the problems of information processing, searching and finding the solutions in the related fields of scientific knowledge. One of the solutions to this kind of problem is the software like expert systems. The aim of the article is to test the existing software tools for constructing expert systems, to make the assessment of the existing feature set, to determine the set of criteria that influences on the selection of software, to analyze the possible ways of using it regarding to structuring, storing, searching and changing of the accumulating knowledge bases. The article provides a comprehensive analysis of existing expert system shells. As a result of the analysis, the lifecycle phase of the expert system are considered, the basic criteria are selected for each phase. Focusing on the most popular criteria, the best software for constructing the expert system is selected. A prototype of the expert system was developed in the selected shells. The complexity of the prototype development is estimated. The shortcomings of the software are identified. The analysis parameters are summarized in a comparative table. The best shell for the construction of expert systems is selected considering the selected criteria.

Keywords: expert system, production system, ES shells, analysis.

1 Introduction

The total amount of information that an average employee uses in work has significantly increased lately. The problem of processing a large array of data comes to the fore in many areas of activity

According to Sviridov S. S., a developer or researcher spends about half of working time searching for necessary information [1]. For example, in the field of customer support for software use, a consultant is guided by about 10 technical standard documents and about 40 instructions of various types (the total volume of documentation is several hundred pages). Information in the above mentioned documents is periodically updated and supplemented. And if the consultant supports several software products or several groups of users solving various tasks, then the situation becomes much more complicated

To reduce the time for finding a solution, searching for answer from several knowledge areas and choosing several alternative solutions, it is recommended to use expert systems (ES). According to Doctor of Technical Science, Professor Gavrilova TA, the expert systems shall be used “... where the main difficulty relates to the use of weakly-structured knowledge of practitioners and where the logical (semantic) processing of information prevails over the computational” [2]. In other words, the expert system

must take up the “bottlenecks” of standard search: to provide a substantive adaptation of the problem, the ability to consider the consultant skill and qualification level. In the case of a significant discrepancy in the levels of knowledge between the system and the consultant, an explanation module is mandatory. In addition, the ES should be able to learn enriching a knowledge base, check the available knowledge for correctness, consistency and completeness.

It should be noted that the task of selecting the expert system for search does not have a unique solution at present. This makes it urgent to study a variety of evaluation criteria, methods of construction and methods of using various expert systems.

The purpose of this paper is to select the best software for constructing the ES relative to CAD tasks.

2 Research Methodology

2.1 Procedure

In the process of studying the problem of selecting the best software, we will:

- determine the set of criteria that influence on the selection of software for the implementation of the ES;
- select the software that meet the selected criteria;

- prepare a “basic prototype” for testing the functionality of the selected software;
- develop the "basic prototype" in the selected software;
- identify potential difficulties in the development of the ES;
- evaluate the complexity of the ES development.

Let us dwell upon the stage of defining a set of criteria for selecting software for constructing the ES.

Factors for selecting the best ES can be conditionally attributed to three different stages of the life cycle of the ES: construction, usage and updating.

When selecting the ES, it is important to consider construction methods that influence on the speed and ease of system construction. According to P. Jackson [3], “... if the success of the project depends on the development term, then you should select a shell with embedded legend tools and advanced user interface. Interface development is one of the most time-consuming stages of system design, and the more of this work can be shifted to the shell, the faster the project will be completed”.

To construct the ES, it is necessary to pay attention to the following parameters:

- work with a convenient method of knowledge representation (logical, problem frame, rule-based etc.);
- ability to work with the knowledge base;
- general approach to all domains;
- presentation of peculiarities of the design model, the design methodology;
- consideration of the qualification of the ES user;
- modularization of the knowledge base;
- level of description details of the domain;
- task execution in the conditions of data lacking;
- availability of an inference machine;
- availability of the event log (record of system malfunction);
- ability to process the exceptional conditions;
- availability of a search mechanism of disambiguation;
- configuration of the necessary auxiliary software and equipment for the operation of the ES;
- adaptive properties (ability to customize for a specific domain);
- knowledge representation method.

In terms of the ES use, it is necessary to analyze following evaluation criteria:

- functionality (finding the answer, inquiry qualification, explanation, training, etc.);
- speed of operation (speed of answer search);
- cost of acquisition, implementation, owning;
- ease of use of the ES (ease of editing tools, accessibility of the help system, user-friendly interface of the shell, ease of learning, etc.);
- availability of the Russian version;
- possibility to study the scheme of obtaining (finding) the answer;
- availability of a thesaurus (dealing with synonyms);

- possibility of searching a “negative” answer;
- logging of use cases (history of hits and search results).

When using the ES, there may be a need to improve the existing functionality or introduce new capabilities. Therefore, it is also important to consider the following parameters:

- compatibility with other computer-aided design software (integration);
- possibility of self-improvement of the ES (open source, availability of an applied programming interface);
- completeness and ease of use of technical documentation for the ES;
- reliability and perspectivity of the software company, as well as the availability of qualified technical support;
- access security, delineation of rights.

To develop the most important criteria for us to select software for constructing the ES, we will carry out a practical study using “Basic prototype”.

On a provisional basis, the “basic prototype” model contains 3 basic elements: the question, the answer, the recommendation. The ES for the “basic prototype” will solve the problem of finding recommendations for some malfunctions during user's work with the software.

At the beginning of the program, the user will be asked what the error is related to - the file, the system start or application software. Depending on the choice, a recommendation will be offered or questions will be asked to clarify the problem. The user can be offered solutions - for example, contact customer support professionals (ACS sector), send e-mail to the developer or find a defective element.

To solve such problems, an important criterion is the method of knowledge representation based on rules because they:

- allow us to create a knowledge base in “If-Then” form which is familiar to user;
- possess modularization – each unit of information can be deleted, changed or added independently of the rest;
- have self-explanatory function - it is possible easily trace the set of rules used for output.

So our intermediate conclusion is that in order to solve our task regarding the best software for constructing the ES with respect to our conditions, the following criteria are of practical interest:

- availability of the graphical interface of the program;
- ease of creating the ES;
- convenient tool for debugging the created system;
- possibility of self-improvement (open source, availability of an applied programming interface);
- necessity for installation of additional software to run the program;
- cost.

Based on the above criteria for developing the "Basic prototype", the following software products were selected to implement the required functionality of the ES.

2.2 CLIPS

The software environment for the development of expert systems is a rule-based system, the Rete algorithm is used as inference engine. The main competitive advantages of CLIPS are:

- acceptable performance;
- explicit syntax;
- availability of calling external functions written in other programming languages;
- modules written in CLIPS can be called by programs written in other languages.

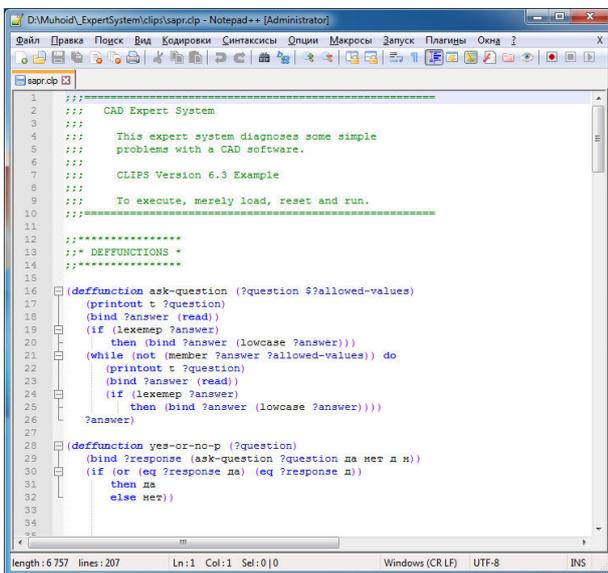


Figure 1 – Sample code written in CLIPS



Figure 2 – Graphical interface of CLIPS

2.3 Java Expert System Shell

Java Expert System Shell (JESS) is a shell for developing expert systems; the system was developed by Sandia National Laboratory.

Intelligent system Jess allows you to create a Java application providing the ability to process data based on knowledge represented as rules.

At the moment, Jess is one of the easiest and fastest shells for expert systems. Like CLIPS, the Jess kernel uses the Rete algorithm to match the facts to the rules, which is very efficient and fast in solving multiple comparison problems. It remembers the result of the last testing of knowledge and re-tests only newly appeared facts. Jess has a closed source code, unlike CLIPS.

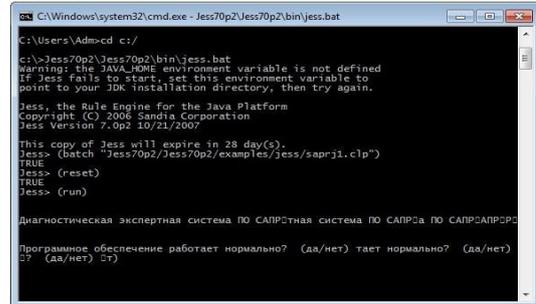


Figure 3 – Graphical interface of JESS

2.4 Prolog

Prolog is a declarative language or predicate calculus language. A predicate is a logical formula from one or more arguments. Prolog involves a set of facts and rules that ensure the finding of solutions based on these facts. Inference mechanism of Prolog is based on a comparison of facts and is the backward chaining inference. It extracts the stored (known) information by choosing answers to requests.

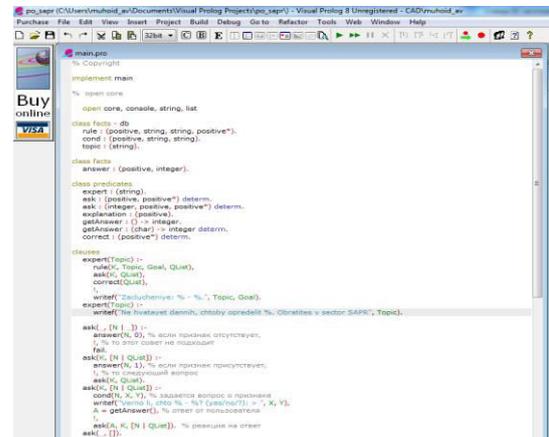


Figure 4 – Interface of IS development

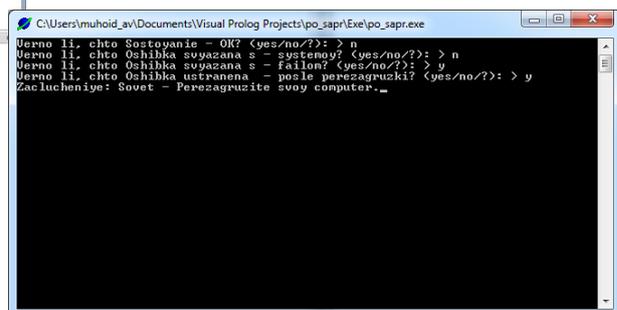


Figure 5 – Window with output of results

2.5 Expert Developer Pro

The program is designed to build a user polled system and find a solution to a particular problem. When customizer the system, one can specify only two options for answering the question – “Yes” or “No”, and then indicate the next question and possible answers. Thus, it is necessary to build a tree for finding the answer. Thus, the logic of the system is completely set by the developer; the possibility of self-learning is excluded.

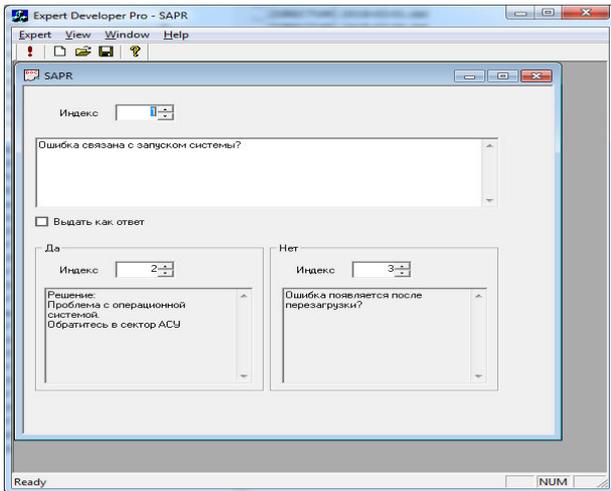


Figure 6 – Interface of ES development

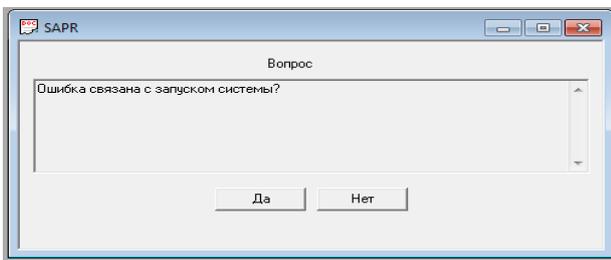


Figure 7 – Dialog window of application development

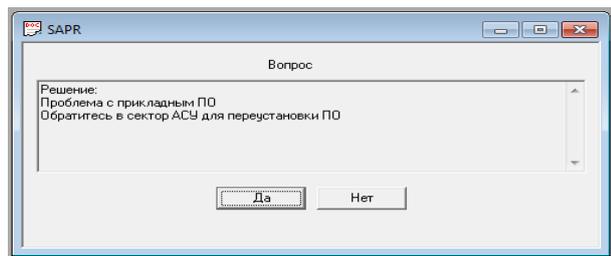


Figure 8 – Window with output of results

2.6 Exsys Corvid

Exsys Corvid is an intelligent system that can be used to develop a knowledge base in any domain. The system includes tools for program debugging and testing. Rules can exist with some probability which is expressed by the coefficient of confidence, the value is set by the expert when developing the knowledge base. The system supports the backward chaining from facts to the goal, linear

programming, fuzzy logic, neural networks and has a SQL interface.

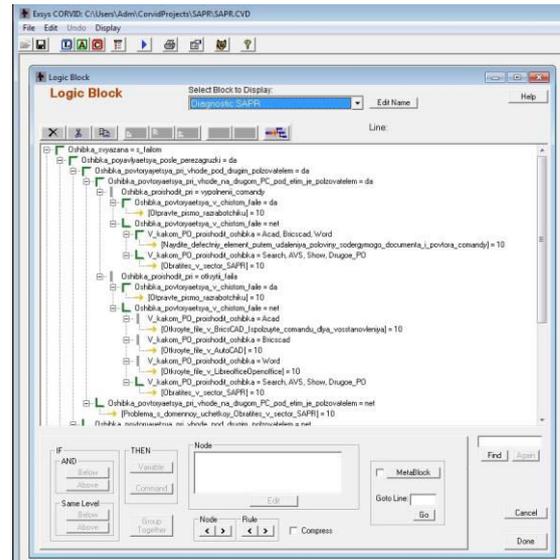


Figure 9 – Interface of ES development

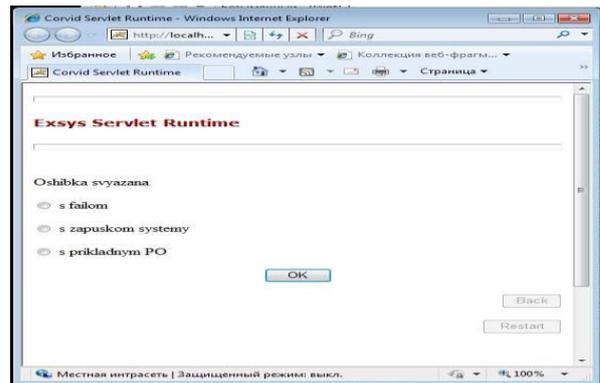


Figure 10 – Dialog window of application development

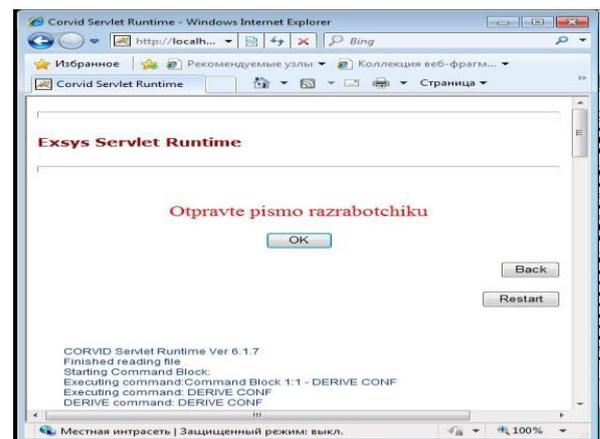


Figure 11 – Window with output of results

3 Results

Let's analyze the functionality of the selected software in relation to the declared set of criteria. The source code for CLIPS can be written in any text editor. To start the program, the shell is used. Dialogue occurs only in text mode by entering values from the keyboard. When writing a program, it is necessary to compile a list of facts and rules. At the same time when the program is working, it is possible to change the fact base which is a process of self-learning of program. It is possible to implement a graphical user interface by means of other software (for example, C++ supports CLIPS commands).

In the Visual Prolog program shell, when writing a program, it is necessary first to describe a list of conditions (facts) during execution of which specific recommendations will be displayed. At the same time, this set is permanent; there is no change in the facts when the program is working. Dialogue with the user occurs in the console desk in text mode. Also, it was not possible to implement the work of the program on the answer "NO" and it is not possible to combine the conditions for "OR".

JESS uses the CLIPS syntax, but it runs on the Java platform, so you need to install additional software of Sun. Unlike CLIPS, the source code is closed. The use of

the program is free for non-commercial use. The program runs in the console text mode.

Expert Developer Pro is a program with a graphical interface. To develop the program, it is necessary to indicate a sequence of questions, while arranging a scheme for finding the answer. The built-in system accepts only 2 variants of answer – "Yes" or "No" thus it is impossible to combine conditions automatically. The source code is closed, so no further development is possible. The advantages include the fact that you do not need to know programming languages to construct the ES, the system is constructed in the user-friendly graphic mode.

Exsys Corvid is commercial software. To work, you need a Java platform, as well install Apache server. It is necessary to build a complete tree for searching the answer, creating variables necessary for the program operation by setting up the logical and command blocks. Self-learning of program is not provided. The user interface is very diverse - you can select the answer option with the help of radio buttons, drop-down list, and active graphic zones. The advantages include the lack of the necessity to learn the programming language, but on the other hand the process of developing and configuring the ES is quite complex and time-consuming.

Table 1 – Comparison of parameters of ES shells

Software	Program operating mode	Ease for development of ES (ranking)	Convenient debugging tool	Possible improvement	Add	Free of charge
CLIPS	Text mode*	2	+	+	–	+
JESS	Text mode	3	–	-	+	+
Visual Prolog	Text mode	4	+	-	–	–
Expert Developer Pro	Graphic mode	1	Not required	-	–	+
Exsys Corvid	Graphic mode	5	+	-	+	–

4 Conclusions

As a result of the research, the following software products used for constructing the ES were tested: CLIPS, JESS, Prolog, Expert Developer Pro and Exsys Corvid. In terms of development ease, Expert Developer Pro and CLIPS proved to be the best. Expert Developer Pro and Exsys Corvid are characterized by graphic mode, and CLIPS can be also integrated with a graphical interface written in other software. Debugging tools are not required for Expert Developer Pro, and only JESS has no convenient debugging tool.

Only CLIPS of the entire tested software has open source code and can be implemented. The disadvantages include the need to install additional software to run the program, which is typical for JESS and Exsys Corvid.

Considering the ease of development, convenient debugging, ability to be improved, self-learning, as well as the absence of the need to install additional software and free distribution, the best choice for constructing the ES can be CLIPS.

References

1. Sviridenko, S. S. (1989). *Modern Information Technologies*, pp. 305.
2. Gavrilova, T. A., & Khoroshevskiy, V. F. (2001). *Intelligent Systems Knowledge Base*, p. 34.
3. Jackson, P. (2001). *Introduction to Expert Systems*, p. 283.
4. Gary, R. *CLIPS online documentation*, Digital resource: <http://clipsrules.sourceforge.net/OnlineDocs.html>.

Вибір оптимального програмного забезпечення для побудови експертних систем

Мухойд О. В., Косторной О. С., Шифрін Д. М.

Науково-дослідний і проектно-конструкторський інститут атомного та енергетичного насособудування
АТ «ВНДІАЕН», вул. 2-га Залізнична, 2, 40003, м. Суми, Україна

Анотація. Стаття присвячена проблемі обробки інформації, пошуку і знаходженню рішень у суміжних областях наукових знань. Одним із рішень такого роду завдань є клас програмного забезпечення – експертні системи. У статті поставлено завдання розглянути представлений на світовому ринку програмний інструментарій для побудови експертних систем; оцінити наявний функціонал, визначити набір критеріїв, які впливають на вибір програмного забезпечення; провести аналіз імовірних засобів використання стосовно до областей структурування, зберігання, пошуку, зміни баз знань, що накопичуються. У статті проведено всебічний аналіз оболонок для експертних систем, що існують. У результаті аналізу розглянуті етапи життєвого циклу експертної систем; для кожного етапу обрано основні критерії. Орієнтуючись на критерії, що користуються найбільшим попитом, виконано підбір оптимального програмного забезпечення для побудови експертної системи. В обраних оболонках розроблено прототип експертної системи. Проведено оцінку трудомісткості та складності виконання прототипу. Виявлено недоліки розглянутих програмних продуктів. Параметри, за якими проводився аналіз, зведені до порівняльної таблиці. Обрана оптимальна оболонка для побудови експертних систем з урахуванням наведених критеріїв.

Ключові слова: експертна система, продукційна система, оболонки для експертних систем, аналіз.