INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN 2278 – 0211 (Online)

Importance of Expert System Shell in Development of Expert System

Sanjay Kumar
Research Scholar, University Deptartment of Mathematics & Information Technology
B.R.A. Bihar University, Muzaffarpur, Bihar, India
Dr. Rajkishore Prasad
Associate Professor, Department of Electronic Science, B.R.A. Bihar University, Muzaffarpur, Bihar, India

Abstract:
The expert system shell simplifies the process of creating an expert system in a faster, cheaper and more efficient way. A single expert system shell can be used to build a number of different expert systems. Information provided by the user is analyzed by the shell with the help of knowledge base before reaching to a particular conclusion. Depending upon requirement, knowledge engineer feeds domain specific knowledge in the knowledge base. For example – if the knowledge engineer feeds, expert level knowledge of 'diagnosis of Papaya plant' then the tool will behave as an expert system for diagnosis of Papaya plant. Thus an expert system shell creates an easy environment for developing expert system. This paper presents the basic concept & functions of Expert System Shells with the aim of highlighting their merits and demerits. This paper also depicts different types of shells along with their functionalities.

Keywords: *Expert system, shell, knowledge engineer, knowledgebase, inference engine*

1. Introduction

Expert system is an artificial intelligence that is used for getting domain specific information/ solution like medical diagnosis, financial advice, agricultural expert, production design etc. The most important part of the system is its shell. An expert system shell is a software development environment containing the basic components for building expert systems. It does not contain knowledge base. For every domain specific system, a knowledge engineer prepares knowledge base with the help of domain experts in a particular area. Thus, the shell-based approaches for building a system focus mainly on the system components but little on the user interface, making shell-based systems very suitable for users with programming skills.

2. Definition

An Expert System Shell (ESS) provides a framework that is used for the development of an expert system (ES) [1][2]. It does not contain knowledge base. In other words, we can say that it's a ready-made expert system without knowledge base. Depending upon requirement, knowledge engineer feeds domain specific knowledge in the knowledge base. For example – if the knowledge engineer [3] feeds, expert level knowledge of 'diagnosis of Papaya plant' then the tool will behave as an expert system for diagnosis of Papaya plant. Thus an expert system shell provides a quick way of developing expert system [4].

3. Architecture

The basic architecture of an expert system shell and its relationship to knowledge base is shown below in figure 1.

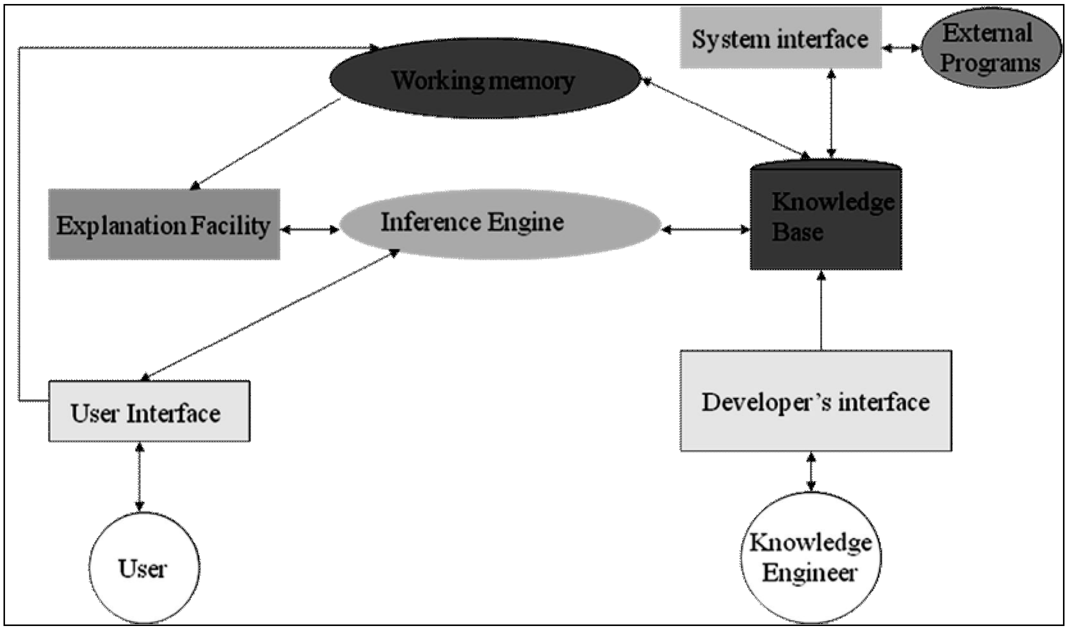

Figure1: Architecture of expert system shell

User Interface received requests from end user and forwards it to the appropriate component of ESS. For example - if the request received from end user is for getting specific information from expert system, it goes to inference engine for its execution. Similarly, if the received request from domain expert is meant for creating or updating inference rules then the interface business logic dispatch the request to knowledge base editor. The knowledge base editor is similar to text editor that provides facility to dispatch the newly created or modified rules to the knowledge base [5]. The inference engine is like a “black box” that performs reasoning over the facts and rules. It receives end user requests via user interface and using pattern matcher it selects appropriate rule from knowledge base. Now the execution engine executes rule in a certain order. There might be possibility that more than one rules in knowledge base matches with the given fact. In such situation the execution engine falls in dilemma to which it executes first [6][7]. It uses different algorithms (like Rete Algorithm used in Jess) [8] by which the execution engine decides an order in which the rule are fired. This is called "Conflict Resolution". Once rules are executed, an outcome provided by inference engine is displayed on user interface that the end user is expecting [9]. The shell component in triangular form is shown in figure 2.

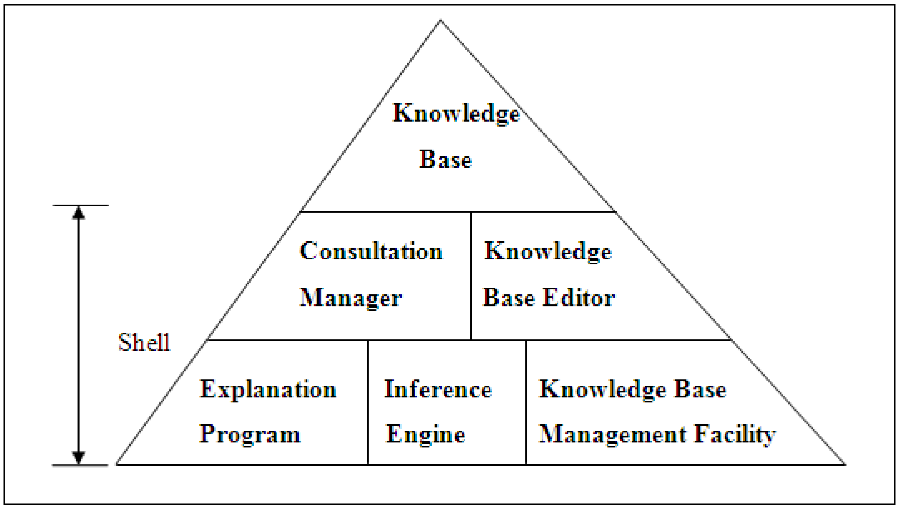

Figure2: Shell Concepts for building an expert system

4. Merits of Expert System Shell

- There is no need to build shell again and again for new expert system. Hence, the time required in building user interface and inference engine is not needed.
- Development of an expert system becomes faster, cheaper and more efficient.
- It doesn't require more experience programmer because there is no need to develop the system from scratch [10].

5. Demerits of Expert System Shell

- If the base is weak then developed product may not be reliable. It means if the shell is not matching the system requirement then the developed system will not represent expertise of a domain expert.
- Some expert systems require different look and fill, security level and user guide. This is not possible in a single expert system shell. It needs customization depending on expert system requirement.

6. Types of Expert System Sell

There are various types of Expert System Shell available till now such as Clips, Jess, Exsys, Vidwan, Knowledge Pro, K-Vision, Age, Emycin, KAS, Leonardo, Xi Plus, Savoir & XpertRule. Some of them are explained below along with their merit and demerits.

6.1. C Language Integrated Production System (CLIPS)

C Language Integrated Production System (CLIPS) is an expert system shell for building expert systems. It is the most widely used expert system shell. It is written in C programming language. Initially, LISP was the single tool available with NASA for all expert system software [11]. This tool was having three major drawbacks like - low availability of LISP on a wide variety of conventional computers, high cost to set-up system using LISP and poor integration with other languages. During 1985s NASA - Johnson Space Center was searching for a tool that is having high portability, low cost and easy integration with external systems. Mr. Gary Riley did complete the hope of NASA and discovered the first version of CLIPS. It's a language that supports many types of programming paradigm in the form of procedural, object-oriented and rule-based programming. Its user interface closely resembles that of the programming language LISP. The inference and representation capabilities provided by the rule based programming language of CLIPS are similar to those of OPS5 of CMU but it is more powerful than it. CLIPS are independently maintained by NASA as public domain software [12][13][14].

6.1.1. Merits of CLIPS

- CLIPS provide three different tools for knowledge representation in the form of programming methodology. These are Procedural, object-oriented and rule-based programming.
- As CLIPS are written in C language, it requires ANSI compiler. All those systems that are having this compiler can have CLIPS based expert system without making any change in source code. Although, source code can be changed, but it totally depends on the business requirement.
- CLIPS can be easily embedded in other subroutine made in C like procedural language to extend the existing functionality.
- CLIPS has various extended versions like - COOL, FuzzyCLIPS, PerlCLIPS, DYNACLIPS etc. that give it an advantage with respect to support of methods like fuzzy logic and agents.

6.1.2. Demerits of CLIPS

- CLIPS supports forward chaining rules, it does not support backward chaining rules.
- Other expert system shell like JESS has also the property of using Java, which in the long run may prove to be a big advantage over CLIPS.

6.1.3. Applications of CLIPS

CLIPS are used in LIMEX, NRES, On-Line Nuclear Power Plant, FESTO, Expert Surgical Assistant etc.

6.2. Essential MYCIN (EMYCIN)

EMYCIN (Essential MYCIN) is an expert system shell derived from MYCIN - an expert system developed for diagnosis of infectious diseases. The MYCIN was one of the best known expert system developed by Dr. Edward Shortliffe in 1974 for medical diagnosis [15]. As MYCIN was domain dependent system, it could not be used for the development of other domain specific expert system. At that time the knowledge engineer were searching for a system that could be used for building expert system for multiple domains. Thanks to Dr. Crawford, who in 1987 had derived an independent system from MYCIN that is domain independent. It does not contain domain specific knowledge base. It means its knowledge base is blank [16].

EMYCIN uses backward chaining inference mechanism for rule interpreter (Reasoning). This shell is PROLOG based. Programs in PROLOG consist of rules and facts. The Knowledgebase editor, an inference engine and the explanation system are three main components of EMYCIN. This system deals with uncertainty. Instead of insisting that all predications, be true or false, EMYCIN associates "certainty factor" with each predication [17]. It provides a highly organized conceptual structure into which the domain knowledge is to be mapped.

EMYCIN = Prolog + uncertainty + caching + questions + explanations

6.2.1. Merits of EMYCIN

- Based on the rules and facts the computation that it does, it caches the results so that it cannot be duplicated.
- It provides an easy way for the system to ask the user for information.
- EMYCIN deals with uncertainty. It associates "certainty factor" with each prediction in lieu of insisting that all predictions be true or false.

6.2.2. Demerits of EMYCIN

- There are many problems in logic like improper certainty handling and false things also not stated efficiently.

6.2.3. Applications of EMYCIN

EMYCIN is mainly used in medical for the diagnosis of infectious diseases.

6.3. Java Expert System Shell (JESS)

The Java Expert System Shell (JESS) is a rule-based tool for building intelligent system called expert system. It acts as an interpreter for Jess language. It was developed in 1995 by Ernest Friedman-Hill of Sandia National Labs, USA. As the Jess engine is implemented in Java, it can also be called / executed by Java object. It contains all the features that CLIPS has. So, we can say it's a superset of CLIPS [18]. Jess can be used to build Java software that has the capacity to "reason" using knowledge you supply in the form of declarative rules [19].

Although, the latest version of Jess is 8.0 but the most stable version of Jess is 7.1p2 that was released on November 5th, 2008.

Jess rule engine uses advance version of the Rete algorithm to match rules to the facts [9]. It uses Rete algorithm for conflict resolution that provides a generalized logical description of an implementation of functionality responsible for matching facts against rules in a pattern-matching production system (a category of rule engine). A production consists of one or more conditions and a set of actions which may be undertaken for each complete set of facts that match the conditions. Conditions test fact attributes, including fact type specifiers/identifiers. The Rete algorithm exhibits the following major characteristics [20]:

- It reduces or eliminates certain types of redundancy through the use of node sharing.
- It stores partial matches when performing joins between different fact types. This, in turn, allows production systems to avoid complete re-evaluation of all facts each time changes are made to the production system's working memory. Instead, the production system needs only to evaluate the changes (deltas) to working memory.
- It allows for efficient removal of memory elements when facts are retracted from working memory. The Rete algorithm is widely used to implement matching functionality within pattern-matching engines that exploit a match-resolve-act cycle to support forward chaining and inferencing.
- It provides a means for many-many matching, an important feature when many or all possible solutions in a search network must be found.

6.3.1. Merits of JESS

- Jess has many unique features including backwards chaining and working memory queries, and of course Jess can directly manipulate and reason about Java objects.
- Jess is also a powerful Java scripting environment, from which you can create Java objects, call Java methods, and implement Java interfaces without compiling any Java code.
- Jess is available at no cost for academic use and can be licensed for commercial use.

6.3.2. Demerits of JESS

- JESS is three times slower than CLIPS because of Java but it totally depends on application type.
- It is a bit more complicated since the software has to be compiled before it can be started.

6.3.3. Applications of JESS

Diagnosis of rice plant diseases, Expert system for corporate financial rating, Expert system for student advising using JESS.

6.4. Exsys Inc - The Expert System Experts

Exsys is a rule-based expert system shell that provides the most efficient and effective way to automate the delivery of problem-solving advice. The rules are easy to learn and essentially the same as an expert would use to explain why they made a decision. Some problems are more procedural and Exsys supports these with a forward chaining option [21]. Others use the rules in a true "heuristic" way, calling and using rules dynamically as they are needed via backward chaining. The Exsys Inference Engine provides both options, and they can be mixed for maximum flexibility.

6.4.1. Merits of Exsys

- The decision-making logic and the process of the domain expert are fully captured in this shell.
- Wrapping the system in a user interface with the desired look-and-feel
- It can be easily integrated with other system i.e it is compatible with other systems [22].
- Exsys software can be easily learned and use. It can be easily deployed on the system. Its maintenance is very cheap.
- Exsys is competitively priced and many businesses see a return on investment almost immediately upon deployment [23].

6.4.2. Demerits of Exsys

- One of the demerits is the inability of the system to effectively and accurately capture subjective factors in countermeasure selection.

6.4.3. Applications of Exsys

As a knowledge automation expert system, Exsys Corvid is being used. Such as E-commerce, Web design, Human resources, Research & Development, Manufacturing, Engineering, Agriculture etc.

6.5. Vidwan

An expert system shell - Vidwan has been developed at the National Centre for Software Technology (NCST), Mumbai (India) in 1993. It's a web-based expert system shell that allows us to create rule-based expert system. As it is a rule-based expert system shell, it enables us to encode domain knowledge in the form of IF-THEN rules [24]. The inference engine of this shell applies backward-chaining inference mechanism to manipulate the piece of information. The explanation component of the shell provides facility to explain an specific query in the form of why and how. It also has an interactive text editor that enables knowledge engineer to update/create rules in the knowledgebase. This software can be embedded in programs developed in other languages [25]. The latest version of Vidwan is also compatible with UNIX operating system.

6.5.1. Merits of Vidwan

- It is a domain independent, customizable, multiuser & web-based expert system shell.
- It allows us to generate different types of report.
- It provides limited arithmetic functionalities along with interactive user interface.
- This software is easy to learn and use.
- It provides explanation facility such as why & how.
- It is compatible with DOS, Windows and UNIX operating system.

6.5.2. Demerits of Vidwan

- An expert system developed using Vidwan is quite costly comparatively.
- As it provides support of database access to user, there might be possibility of anomalies in database.

6.5.3. Applications of Vidwan

Tax planning advisor, Medical advisors like Cardiology (Hurday), Respiratory diseases, Financial & investment advisors etc [26].

7. Conclusions

The expert system shell is very helpful in building any domain specific expert system. At present few shells are web-based while few are stand alone. Some shells are license based while some are freeware. With the introduction of shell in the world of Artificial Intelligence, the development of expert system has become quite comfortable in the form of overall cost, time and skills for building the system.

8. References

1. Jackson, Peter (1998), Introduction to Expert Systems (3 ed.), Addison Wesley, p. 2- 4, ISBN 978-0-201- 87686-4.
2. Expert System & Knowledge Engineering in Wikipedia compiled by: Dr. Mohsen Kahani, pp. 2-5. 14 Sep 2012.
3. Ankita Kaur, Mansaf Alam (2013) Role of Knowledge Engineering in the Development of a Hybrid Knowledge Based Medical Information System for Atrial Fibrillation. American Journal of Industrial and Business Management, 3, 36-41.
4. Eugenia IANCU1 & Simona BUTA, Potential Developments of Expert Systems Using Infor- mation About Explicit & Tacit Knowledge http://www.arpapress.com/Volumes/Vol9Issue3/ IJRRAS_9_3_09.pdf.
5. EMYCIN: A Knowledge Engineer’s Tool for Constructing Rule-Based Expert Systems William van Melle, Edward H. Shortliffe, and Bruce G. Buchanan, pp. 307-310.
6. Griffin, N.L. "A Rule-Based Inference Engine which is Optimal and VLSI Implementable".http://www.cs.uky.edu University of Kentucky. Retrieved 6 December 2013.
7. Feigenbaum, Edward; Avron Barr (September 1, 1986). The Handbook of Artificial Intelligence, Volume I. Addison-Wesley. p. 195. ISBN 0201118114.
8. Forgy, C.L. Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem Artificial Intelligence, 19(1982) 17-37.
9. The RETE Algorithm, http://www.cis.temple.edu/giorgio/cis587/readings/rete.htm.
10. Introduction to artificial intelligence and expert systems, Copyright 1993, 1994, 1995 by Carol E.Brown and Daniel E.
11. http://www.siliconvalleyone.com/founder/clips/index.htm, NASA-CLIPS Rule-based Lang.
12. http://www.lsi.upc.edu/~bejar/ia/material/laboratorio/clips/CLIPS-snippets-eng.pdf.
13. CLIPS based decision support system for water distribution networks. Sandeep, K.; Rakesh, K. //Drinking Water Engineering & Science;2011, Vol. 4 Issue 1, p-37, ISSN 0975-5462.

14. Tilotma S, Navneet T, Khushboo S - A Rule-Based Expert System based on Certainty Factor and Backward Chaining Approach (student counseling system) vol. 2, Issue 1, January 2013, ISSN 2319 - 4847, IJAIEM.
15. The MYCIN Experiments of the Stanford Heuristic Programming Project, 754 pp., references, index, illus. electronic text Addison Wesley, Reading, MA, 1984, Edited by Bruce G. Buchanan and Edward H. Shortliffe.
16. The Structure of the MYCIN System by William van Melle.
17. Adams J.B - A probability model of medical reasoning and the MYCIN model, Mathematical Biosciences Vol.32, 1976 pp. 177-186.
18. Jess in action, Rule-based systems in Java by Ernest Friedman Hill, ISBN 1-930110 -89-8.
19. Fadhil A. Ali, "Expert system of single magnetic lens using JESS in Focused Ion Beam", Vol.3, No.6, December 2013, DOI:10.5121/ijcsa.2013.3601 1.
20. The Rete Algorithm: An introduction to the technical details of how Jess works, SAND98-8206 (revised) Unlimited Release, First Printed November 1997.
21. Building Knowledge Automation Expert Systems with Exsys Corvid by Dr. Elias Awad.
22. Integration possibilities in Exsys Corvid by Daniel Pavaloaia, Assistent at "Al. I. Cuza" University of Iaşi, Romania.
23. Knowledge Automation Expert System Software User Guide - Exsys Corvid User Manual.
24. Expert systems: Programming languages, tools and shells, July 14, 2014 by Shruthi S filed under Viral Nagori & Dr. BhushanTrivedi.
25. Computer education in India past, present and future by Dr. Utpal K. Banerjee, The Institution of Electronics & Telecommunication Engineers, pp 234-235.
26. Principles of artificial intelligence and expert system development by Rolston D. (1988), McGraw Hill book company Singapore.