Poonam Shodh Rachna (ISSN 2456-5563)
(A multidisciplinary Research Journal)
Vol. 1, Issue.V, July 2022, PC : PSR-2207012

A REVIEW ON OPEN SOURCE AND FREE EXPERT SYSTEM SHELLS

G.N.R. PRASAD
Sr. Asst. Professor, Dept. of MCA, CBIT, Gandipet, Hyderabad
E-mail : gnrp@cbit.ac.in

Abstract

A whole development environment for creating and sustaining knowledge-based applications is called a shell. It gives a knowledge engineer a step-by-step process so they may directly include the subject matter experts in organising and encoding the knowledge. Shell use cuts down on development time by up to 50%. Open source is a set of values and procedures that encourages public access to the creation of commodities and knowledge. It is important for people to have the freedom to utilise software in all manners that are beneficial to society. Software is different from tangible items like chairs, meals, and fuel in that it is much easier to copy and modify. Software is only as valuable as its potential uses allow it to be. The numerous Open source and free software related to Expert System Shells is covered in this article. In this study, many expert system shells and their comparisons are examined.

Keywords : Expert System Shell, Open Source, Free Software

1.0 INTRODUCTION

Expert systems that are knowledge-based, or simply "expert systems," leverage human knowledge to address issues that ordinarily call for human intellect. These expert systems use computer data or rules to represent the expertise knowledge. When an issue has to be solved, this data and rules may be used. Although books and manuals contain a vast amount of knowledge, a person must read and interpret the information in order for it to be put to use. Traditional computer programmes carry out operations using conventional decision-making logic; they only have a rudimentary understanding of the method to solve that particular problem and the requisite boundary conditions.

The majority of expert systems are created using specialised software programmes known as shells. These shells need information to be input in a certain format and include an inference mechanism (backward chaining, forward chaining, or both) (all of which might lead some to categorise OPS5 as a shell). In addition to these characteristics, they frequently include tools for creating user-friendly user interfaces, manipulating lists, strings, and objects, and interacting with databases and other applications. Even though they have a far smaller application space than the majority of computer languages, these shells are nevertheless considered to be languages.

2.0 EXPERT SYSTEM SHELLS

Although a knowledge base may be built using any ordinary programming language, the expert system shell makes the process easier. The information input by a user is really processed by the shell, which then compares it to the knowledge base's ideas and offers an evaluation or fix for a specific issue. In order to control the entry and output of data, an expert system shell offers a layer between the user interface and computer operating system. In order to reach a certain conclusion, it additionally manipulates the data supplied by the user in conjunction with the knowledge base. The shell's structure is quite reminiscent of an interpreter or the front end of a database software. The shell also controls the user interface, handling tasks including validating numerical numbers placed on the screen, controlling the mouse, and displaying graphical objects.

The shell is frequently offered as a finished good that enables the buyer to create a knowledge base from start in the same manner that they would buy a database management system. On the other hand, knowledge bases can be marketed as goods, much like data, where a shell or interpreter may be included incidentally in the package.

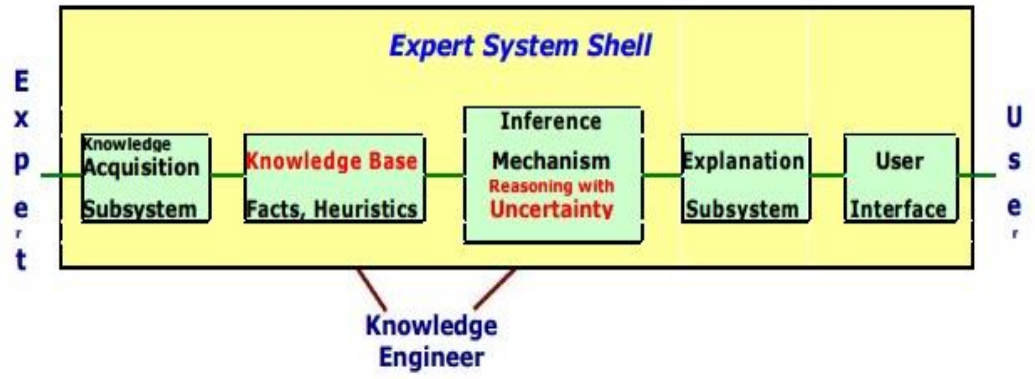

Fig : Expert System Shell

2.1 Java Expert System Shell (JESS)

The Java Expert System Shell is a quick, scalable, and portable reasoning engine created in the Java programming language by Sun Microsystems. It was created at Sandia National Laboratories in the USA and is offered without charge for educational purposes. Jess has a big user base that spans the globe. Because it offers an easy approach to include extensive reasoning skills into Java-based applications, Jess is frequently utilised in agent research. Numerous software extensions for fuzzy logic, databases, blackboards, and language comprehension have been created by Jess users. Many practical systems, including those built on a multiagent paradigm, have been deployed using Jess.

At Sandia National Laboratories in Livermore, California, Ernest Friedman-Hill created the rule engine and scripting environment known as Jess. You may create Java software with the ability to "reason" using the knowledge you provide in the form of declarative rules by utilising Jess. One of the quickest rule engines out there is Jess, which is compact and lightweight. You have access to all of Java's APIs thanks to its robust scripting language. Jess comes with a fully functional development environment built on the acclaimed Eclipse platform.

2.2 Jboss.drools

Business rule management system (BRMS) Drools is based on Charles Forgy's Rete algorithm and an upgraded Rules Engine implementation, ReteOO, that is specially designed for Java. More crucially, Drools offers for Declarative Programming and is flexible enough to fit the semantics of your issue domain with Domain Specific Languages, graphical editing tools, web based tools and developer productivity tools.

2.3 Euler Proof Mechanism

Euler is an inference engine supporting logic based proofs. It is a backward-chaining reasoner enhanced with Euler path detection. It has implementations in Java, C#, Python, Javascript and Prolog.

2.4 InfoSapient

It's a rules engine that enables users from the business world to enter and update business rules using standard language. The automatic decision-making that results is comparable to how knowledgeable humans make decisions. To choose the optimum course of action, a number of factors may be weighed concurrently along with risks and possibilities. InfoSapient will always choose the "best" course of action in the presence of confusing circumstances, which are frequently present in

business operations. This eliminates the need to rely on the knowledge and expertise of a widely dispersed staff to make complicated decisions when they are required.

2.5 Jena

A Java framework is used to create Semantic Web apps. It offers a programming environment for RDF, RDFS, and OWL as well as SPARQL and has an inference engine that uses rules. As a result of work done with the HP Labs Semantic Web Program, Jena is open source.

2.6 JLog

It is a Java-based Prolog interpreter implementation. Its main advantage is that it can be used practically anywhere that supports Java (including a web browser), making it ideal for educational settings. It functions as both an applet and an application.

A GUI debugger, inquiry panels, online help, animation primitives, and a built-in source editor are all included. When used as a programme, it can open and save files but not when used as an applet (because of the security model for web browsers; however it is possible to copy and paste Prolog source between JLog and a local text editor).

Developers that require a Java-based embedded Prolog engine can use JLog. JLog makes it simple to consult, create queries, and assess query outcomes. Additionally, it has tools for mapping between Prolog words and typical Java objects.

2.7 JEOPS

A project called the Java Embedded Object Production System aims to offer Java production system power. Through a collection of classes intended to give Java some form of declarative programming, JEOPS enhances this language with forward chaining, first-order production rules. That makes it easier to create intelligent software, such software agents or expert systems. JEOPS is a forward chaining RULE ENGINE built on Java. Through the usage of rules, Java Application Servers, client applications, and Servlets are able to enhance business processes.

2.8 FOCL

It is a Common Lisp-based expert system shell and machine learning software. Quinlan's FOIL programme is expanded by the machine learning programme by including a suitable explanation-based learning component. Horn Clause programmes are taught to FOCL by examples and (optionally) prior knowledge. The expert system has a graphical interface to the rule and fact base as well as a backward-chaining rule interpreter.

2.9 BABYLON

It is an environment for creating expert systems. It comes with frames, restrictions, a logic formalism akin to Prolog, and a description language for medical applications. It has been ported to a variety of hardware systems and is written in Common Lisp.

2.10 MIKE

A fully functional, portable, and free software environment called the Micro Interpreter for Knowledge Engineering (MIKE) was created with teaching at the UK's Open University in mind. It has frame representation language with inheritance and "demons" (code activated by frame access or modification), as well as user-settable inheritance techniques. It also has forward and backward chaining rules with conflict resolution strategies that the user defines. Both user-specified and automatic "why" explanations (evidence histories) are offered for rule execution. There includes features for both coarse- and fine-grained rule tracing, as well as a new "rule graph" presentation that clearly displays the history of rule execution. A conservative and portable subset of Edinburgh-syntax Prolog was used to write MIKE, which serves as the foundation of an

Open University course on knowledge engineering. It is given as unprotected source code. In the BYTE issue from October/November 1990, MIKE version 1 was discussed. Two more recent versions have replaced MIKE v1.50, which was previously accessible via a number of ftp servers. These newer versions are MIKEv2.50, a turnkey DOS version with a menu-driven interface and frame- and rule-browsing tools, and MIKEv2.03, a full Prolog source code version with a RETE algorithm for fast forward chaining, a truth maintenance system, uncertainty handling, and hypothetical worlds.

2.11 WindExS

The Windows Expert System is a forward chaining expert system that runs on Windows and is completely functional. Because of its modular nature, the user may add new modules as needed to increase the system's capabilities. Inference Engine, File Manager, User Interface, Message Manager, and Knowledge Base modules are available in WindExS. It offers graphical knowledge base representation and forward chaining.

2.12 RT-Expert

C programmers can include expert systems rules into their C or C++ projects using this freeware expert system. A rule-compiler that turns rules into C code and a library that houses the rule execution engine make up RT-Expert. Borland Turbo C, Borland C++, and Microsoft C/C++ compilers are all compatible with RT-Expert for DOS. The personal edition is permitted for usage in academia, research, and leisure activities. The RT-Expert personal edition is not licenced for use in commercial applications. Commercial software running in the DOS, Windows, and Unix environments can be found in professional versions.

2.13 CLIPS 6.0

An OPS-like forward chaining production system, the C Language Integrated Production System was created by NASA in ANSI C. The CLIPS inference engine has customised conflict resolution methods, dynamic rule insertion, and truth preservation. It is simple to integrate CLIPS—including the runtime version—into other programmes. The inference engine is directly connected with the object-oriented language that is part of CLIPS, known as COOL (CLIPS Object-Oriented Language). The platforms on which CLIPS may be used include Macintosh, VAX 11/780, Sun 3/260, and HP9000/500. IBM PC compatibles, including Windows 3.1 and MS-DOS 386 versions, are also supported.

2.14 FuzzyCLIPS

It is an expansion of the rule-based expert system shell from CLIPS that can represent and work with fuzzy rules and facts. FuzzyCLIPS can handle precise, fuzzy (or inexact), and combination reasoning in addition to the CLIPS capabilities, enabling the free mixing of fuzzy and conventional terminology in expert system facts and rules. Fuzziness and uncertainty are two fundamentally imperfect notions used by the system. Versions are accessible for PC, Macintosh, and UNIX platforms. Please read the terms of use in the FuzzyCLIPS documentation before using the programme, which is free to use. The WWW provides access to FuzzyCLIPS (World Wide Web).

3.0 CONCLUSION

The word free means not free, it is freedom. You are freedom to use the software and to modify the code. The open and free software provide the similar functions of commercial or proprietary GIS software. The lack of awareness among on us is moving towards proprietary software. The GNU licensed software supports free software. The JESS is top and most used free open source used by many developers.

REFERENCES

[1] 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.

[2] 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.
[3] 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.
[4] Griffin, N.L. "A Rule-Based Inference Engine which is Optimal and VLSI Implementable".http://www.cs.uky.edu University of Kentucky.
[5] 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.
[6] Ruyle, Kim E. (1988, August). Artificial Intelligence in Education. 5pp. ED 306 945.
[7] Shaoul, Jean, & Smith, P.J. (1987). Expert Systems--A Competent Tool for Training? 6pp. ED 317 173.
[8] Wilson, Harold O., & Burford, Anna Marie. (1990, March). Artificial intelligence and expert systems. Journal of Education for Business, 65(6), 275-79. EJ 408 060.
[9] Bruce G. Buchanan and Edward H. Shortliffe, "Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project", Addison-Wesley, Reading, MA, 1985.
[10] Frederick Hayes-Roth, "The knowledge based ex pert system: A tutorial", IEEE Computer 17(9):11-28, 1984.
[11] Bruce G. Buchanan and R.O. Duda, "Principles of Rule-Based Systems", Tech Report HPP-82-14, 1982.
[12] Bruce Buchanan and Reid Smith, "Fundamentals of Expert Systems", Annual Review of Computer Science 3, 23-58, 1988.
[13] Joseph Giarratano and Gary Riley, "Expert Systems Principles and Practice", PWS Publishing (20 Park Plaza, Boston, MA
[14] James P. Ignizio, "Introduction to Expert Systems: The Development
and Implementation of Rule-Based Expert Systems", McGraw-Hill, 1991.
[15] Samuel J. Biondo, "Fundamentals of Expert Systems Technology: Principles and Concepts", Ablex, Norwood, NJ, 1990. 160 pages
[16] Dennis Merritt, "Building Expert Systems in Prolog", Springer-Verlag, 1989.
[17] David Hu, "C/C++ for Expert Systems", Management Information
Source, Portland, OR
[18] Firebaugh, Morris W., "Artificial Intelligence: A Knowledge-Based
Approach", PWS-Kent, Massachusetts
[19] Charles L. Forgy, "OPS5 User's Manual", Technical Report,Carnegie Mellon University, School of Computer Science, Pittsburgh
[20] Thomas Cooper and Nancy Wogrin, "Rule-based Programming with OPS5", Morgan Kaufmann Publishers, 1988
[21] Charles L. Forgy, "RETE: A fast algorithm for the many pattern/many object pattern match problem", Artificial Intelligence 19(1):17-37, September 1982.
[22] Daniel P. Miranker, "TREAT: A better match algorithm for AI production systems". In Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87), pages 42-47, August 1987.
[23]Daniel P. Miranker, "TREAT: A New and Efficient Match Algorithm for AI Production Systems", Morgan Kaufmann Publishers, 1990,143 pages.
[24] Mark Perlin, "The match box algorithm for parallel production system match", Technical Report CMU-CS-89-163, Carnegie Mellon University, School of Computer Science, Pittsburgh, Pennsylvania, May 1989.
[25] Michael A. Kelly and Rudolph E. Seviora, "An evaluation of DRETE on CUPID for OPS5 matching", in Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89), pages 84-90, Detroit MI, August 1989, Morgan Kaufmann Publishers.
[26] Dr. G. N. R. Prasad, "Intelligent online job portal system using java expert system shell (jess)", International Research Journal of Modernization in Engineering Technology and Science, Volume:04/Issue:05/May-2022.