http://sajie.journals.ac.za ISSN 1012-277X S.Afr.j.ind.eng.

S A Tydskrif vir Bedryfsingenieurswese, Vol 3, No 1, Junie 1989 pp 30-39

Software Tools for the Development of EXPERT SYSTEMS

P S Kruger

Department of Industrial and Systems Engineering
University of Pretoria
Pretoria
South Africa

ABSTRACT

This paper will attempt to provide information with regard to some of the expert system development tools that are available. It will also provide some guidelines which may be used in the selection of appropriate system development software.

OPSOMMING

Hierdie artikel sal poog om informasie te verskaf oor sommige van die ekspertstelselontwikkelingsprogrammatuur wat tans beskikbaar is. Riglyne word verskaf wat gebruik mag word in die keuse van toepaslike ontwikkelingsprogrammatuur.

"For every tool there is a task perfectly suited to it".

Davis' Law

1. CLASSIFICATION OF DEVELOPMENT SOFTWARE

The development of an expert system presupposes the availability of some kind of development software. A large variety of such software exists and may be classified, for convenience sake, as follows :

General purpose conventional (procedural) languages

It may be possible to develop an expert system using any general purpose procedural language for example FORTRAN, BASIC, Pascal, C or ADA. Since these languages are procedural rather than declarative, the inherent language structure is awkward and not really suitable for the development of relative complex expert systems. However, a significant number of expert systems, as well as expert system development software, have been developed using these languages especially the C language.

Low level Artificial Intelligence (declarative) languages

Languages such as Lisp, Prolog and Smalltalk were designed with artificial intelligence applications, and therefore expert systems, in mind. Although these languages are very flexible they may not be very easy to use since they require an in-depth knowledge of the principles of artificial intelligence as well as significant programming skills.

Lisp is at present the most widely used artificial intelligence language although it is in principle very similar to a procedural language. Prolog, a true declarative language with a build-in inference engine, is gaining in popularity partly due to the fact that the Japanese Fifth Generation Project is based on Prolog as well as the contribution of Borland's Turbo Prolog. Numerous implementations of both Lisp and Prolog are available. Smalltalk, and its variations, is representative of Object Orientated Programming Systems (OOPS). This particular approach, significantly different from both Lisp and Prolog, promises to become very popular and useful for the development of expert systems.

Expert System Shells

An expert system shell consists of a ready made expert system, including the inference engine, user interface, knowledge representation scheme and various built-in utilities, but with an empty knowledge base. It is therefore relatively easy to develop an expert system using a shell but with a comparative loss in flexibility. A bewildering array and variety of shells are available such as Micro-Expert, Synapse, VP Expert, 1 st Class, EXSYS, INSIGHT, Expert Edge, KES II, Savoir, dmX, Xi Plus, Twaice, Personal Consultant Plus and many more. Some shells provide an interface to a low level artificial intelligence language. For example, Personal Consultant Plus provides the capability to integrate programming code written in PC Scheme, a specific implementation of Lisp. Using a shell is probably the easiest, quickest and most cost-effective alternative for anyone considering the use of expert system technology for the first time.

Higher level development environments (toolboxes)

Hybrid expert system development environments such as Arity, Goldworks, KEE, ART and RuleMaster, was developed in an effort to maintain some of the flexibility of low level declarative languages but at the same time to provide the user with some of the power, convenience and ease of use of a shell.

Integrated Software

An expert system may include significant traditional computing components such as data base, spreadsheet and numerical operations. Most expert system software provides an interface to other existing software for this very purpose. However, a software system such as Guru provides most of these features in a single integrated package.

Deciding on an approach represented by one of these categories invariably results in a trade-off between flexibility and convenience. A shell may be easy to learn and convenient to use but may lack flexibility. On the other hand an artificial intelligence language may offer all the necessary flexibility but may be difficult to use and may require special programming skills. These concepts are illustrated in Figure 1 [3].

Ease of use and convenience

==========>>>>>

Conventional Programming Languages	Artificial Intelligence Programming Languages	Higher level Development Environments	Integrated Software	Expert System Development Shells
FORTRAN PASCAL C etc.	LISP PROLOG SMALLTALK	Arity Goldworks RuleMaster etc.	Guru	PC Plus Synapse VP Expert etc.

<<<<<==========

Development time and flexibility

Figure 1 Spectrum of expert system development tools

2. CHARACTERISTICS AND FEATURES OF DEVELOPMENT SOFTWARE

Artificial intelligence, and therefore expert system applications, are traditionally very demanding of hardware capabilities especially in terms of execution speed and memory requirements. For this reason special "Lisp machines" and artificial intelligence work stations have been developed. However, recent advances in both hardware and software have made it possible to develop and run fairly large expert systems on general purpose and less expensive computer hardware. For example, a large variety of expert system shells as well as implementations of artificial intelligence programming languages are available for microcomputers. These software packages may be used to develop and run expert systems with the size and complexity necessary for a system of practical value.

Choosing a software package for the development of an expert system is a very important decision which may be crucial to the success of the project. As already mentioned, a large variety of similar but nevertheless significantly different software packages are available. Since software selection is problem and application specific it is not possible to identify, in general terms, the "best" package. One possible approach would be to consider some of the more common characteristics and features that may be necessary for development purposes and may, or may not, be available in a specific package. In this way a checklist may be compiled and used to compare the features of different packages with the purpose of selecting the most appropriate one.

The characteristics and features of development packages may be organized in five categories as follows :

The **development environment** referring to the characteristics and capabilities which may be important during the process of developing a system,

the **run-time environment** indicating facilities which may be important from the end-user's point of view,

the **hardware/software environment** referring to the hardware requirements and the software characteristics,

the **artificial intelligence features supported** by the software, and

the **support services provided** by the software and its distributors.

These categories are not mutually exclusive since some characteristic may be of importance in more than one category.

In the following paragraphs reference will be made to several commercially available expert systems packages with the purpose of illustrating the implementation of some of the concepts. However, most of the remarks will refer to VP Expert, from Paperback Software, and Personal Consultant Plus (PC Plus), from Texas Instruments, being the two packages with which the author is most familiar. Furthermore, the remarks will be limited to some extent to development shells and particularly to those available for microcomputers.

2.1 THE DEVELOPMENT ENVIRONMENT

Knowledge base editor/creation

Some kind of text editor is usually needed for the creation of the knowledge base. The **on-line availability** of such an editor (PC Plus) may be advantageous although the possibility to use any **general purpose** editor (VP Expert) may be convenient. A dedicated editor may provide facilities such as **syntax and consistency checking** as well as features such as **automatically returning to the editor** (VP Expert) at the appropriate position whenever a compile or run-time error occurs.

Access to the rest of the software environment

It is often necessary to interface an expert system with other software. Such an interface may imply that the expert system is in control and calling the other software or the expert system may be imbedded in the other software. For these reasons characteristics such as **operating system accessibility, hooks to data base software, development software and other support software** as well as the ability to execute **user defined functions and procedures** may be important. For example, VP Expert has the capability to directly write or read to or from dBase and Lotus 1-2-3 files. Access to the development language of a shell (PC Plus/PC Scheme) increases the flexibility of the development software and may, for example, make it possible to customize an application to a larger degree than what would have been possible otherwise. Real time **interfacing** of an expert system with external sensing devices may be necessary for process control applications and PC Plus is at present one of the few products providing such a capability.

Debugging aids

Debugging any computer programme is a very important but arduous task. This is also true, even more so, in the case of expert system programmes which indicates **tracing, breakpoint controls, cross referencing, saving of cases (journaling)** and other **interactive** features may be important for efficient development.

Knowledge acquisition aids

Developing the knowledge base is the most important but also the most difficult part of any expert system project. Some software provides **knowledge extraction** utilities which may vary from simple **rule generation** from examples (VP Expert) to relative sophisticated rule induction and **run-time knowledge acquisition** (RuleMaster/RuleMaker).

Other support tools

Many expert system software products provide additional software for specific purposes. For example, PC Plus provides the add-on products PC Images and PC Online, for purposes of graphical input/output and real time interfacing respectively while PC Scheme, an implementation of Lisp, is provided as part of the PC Plus software.

Explanation/elaboration facilities

The capability to explain a line of reasoning, or how a specific goal was reached, is one of the outstanding characteristics of expert systems and important both from a development and user point of view. Some expert system shells provide build-in explanation facilities based on the syntax of the relevant rules while additional elaboration facilities may be added with the purpose of making the explanation more readable to the user (VP Expert and PC Plus).

On-line help facilities

The availability of on-line help facilities have become the de facto standard for any well designed software package. Most expert system development software provide some on-line help but the usefulness may vary significantly from product to product. VP Expert has a help facility which is to some extent intelligent since it will be invoked automatically whenever it senses that the developer is experiencing undue difficulties.

Display and output format utilities

The development of an expert system usually has as a final goal the production of a run-time or delivery system which may be used by a non-expert. Utilities to help format output displays in an attractive, readable and user friendly way are therefore important.

Rapid prototyping

An expert system development project often suffers initially from severe uncertainty about the scope and usefulness of the proposed system. The possibility to produce rapid prototypes aids in the early detection of probable strengths and weaknesses of the proposed system and may avoid expensive mistakes in terms of general approach and the choice of software. For example, PC Plus may be used as a rule based shell for prototyping purposes. However, since it also supports frames, real time interfacing, graphics and PC Scheme the facilities exist to expand the prototype into a final delivery system.

Flexibility

As already mentioned, development software may differ significantly with respect to flexibility and ease of use. A trade off between these two characteristics is almost always inevitable.

2.2 THE RUN-TIME ENVIRONMENT

User interface and response handling

The user interface is important since it is the link between the development software/expert system and the outside world represented by either the developer or the final end-user. The development and end-user interfaces may be very similar (VP Expert) or may differ significantly (Turbo Prolog) since the needs and demands of the two types of users are not the same. An experienced developer may prefer a **command driven interface** while an inexperienced end-user may prefer a **menu driven interface**. Similar preferences may exist for facilities such as **automatic** or **programmable interface control**. Development shells often provides very specific features for response handling such as **automatic menu creation** (VP Expert), **graphic input** (PC Plus), **range/type checking** (PC Plus) and **simulated analog input** (Synapse). Other similar facilities may include **interactive query (how, why, why not etc.)**, **don't know or care responses**, **uncertain responses**, **multiple responses**, **numeric responses**, **string responses**, **yes/no responses** and **re-answering of questions**. Conversely, the creation of the interface and response handling may be the total responsibility of the developer (Turbo Prolog and Golden Common Lisp). The **user attractiveness** of a delivery system should not be under estimated and therefore capabilities such as **high resolution colour screens**, **colour graphics** and the **efficiency or speed** of the interface may be of value.

Explanation facilities

Apart from the usual explanation facilities already discussed an end-user may also have need for **retrospective** and/or **hypothetical reasoning** capabilities during a consultation.

Run time performance

Expert systems as well as the relevant development software are notoriously demanding on hardware capabilities such as **execution speed**, **memory** and **disk space requirements**. For example, PC Plus needs at least 15 Megabyte of available disk space and only performs reasonably well with at least 2 Megabyte of available memory if a microcomputer is used.

Natural language interface

Some development software include the necessary basic structures to develop a natural language interface for the specific expert system (Guru).

2.3 THE HARDWARE/SOFTWARE ENVIRONMENT

Type of software

As already mentioned, a development software package may be classified as a knowledge engineering development language, a toolkit development environment, a shell development environment and an integrated development environment. Each one of these types of software has its own advantages and disadvantages for a particular application.

Application integration

Some development software may only be used for stand alone applications but the majority has the capability to handle external routines as well as input from/output to other software. So called toolbox facilities (Arity and Goldworks) enhances the application integration possibilities while built-in integration is provided by, for example, Guru.

Cost

The cost of the necessary development hardware and software, support software, run time software and delivery hardware should be considered in the process of deciding on the most appropriate development approach. Prices of development software may vary over a very wide range and it is not always easy to identify the reasons, if any, for the wide discrepancies which may exist.

Other considerations

Other factors such as special hardware requirements, operating system, development language, software portability, limitations on maximum size of the knowledge base and source code availability should also be considered. Furthermore, the transportability of the development software as well as delivery systems may also be important. It is for instance possible to develop and deliver expert systems based on PC Plus using microcomputers, workstations (TI Explorer) or minicomputers (DEC VAX). If the distribution of more than one delivery system is contemplated, the availability and cost of a run-time version of the development software should be investigated. Factors which are difficult to quantify but nevertheless important, may include the estimated ease of learning, development and maintenance associated with a particular package as well as the present programming skills available.

2.4 THE ARTIFICIAL INTELLIGENCE FEATURES SUPPORTED

Knowledge representation schemes supported

Various knowledge representation schemes exists, each with its relative advantages and disadvantages. At present the most popular approach, implemented in most software packages, is based on the concept of rules. Although the specific implementation and syntax may differ significantly from one package to another they are all based on the well known IF...THEN...ELSE construct. A representation scheme, using frames in conjunction with rules (PC Plus) makes it easier to organize and develop large knowledge bases. Semantic networks may be implemented using an artificial intelligence language such as Prolog or Lisp.

Inference strategies supported

It is possible to develop an application dedicated inference strategy using a language such as Lisp. However, most software packages supports build-in inference strategies using **forward chaining**, **backward chaining** (VP Expert, Turbo Prolog) or **combinations** (PC Plus) and it is usually worthwhile to exploit these capabilities.

Inference control

The capability to control (fine tune) the inference strategy for a specific application may be used to significantly enhance the performance of a particular expert system. The control available in a specific software package may be **automatic** or **programmable** and may use **rule base ordering**, **significance/cost/grouping of rules** and/or **forward propagation control**. In general inference control is known as **meta-level knowledge** since it consists of knowledge about how to use the knowledge base effectively. PC Plus provides various such facilities, for example the capability to specify a group of rules to be used before the current rule is executed. Furthermore, rules may be ranked according to usefulness while priorities may be assigned to goals in a multi-goal situation. These ranking and priority parameters may be changed by meta-rules providing a kind of learning capability.

Handling of uncertainty

Expert systems, being real world applications, need to be able to handle uncertainty in one way or another. **Certainty factors** is at present probably the most popular approach but other approaches using **Bayesian logic**, **fuzzy logic** and **real value logic** is also available.

2.5 THE SUPPORT SERVICES PROVIDED

The support services provide by the software developers or distributors may be crucial during the development software selection process. Some of the factors which should be taken into consideration are :

The availability of a **demonstration version** (Synapse, EXSYS),
the quality of the **documentation**,
the **track record and application base** (PC PLUS),
the availability of a **tutor or tutorial** (Guru, EXSYS, Savoir),
whether **local training** is available (Synapse, PC Plus, dmX, EXSYS),
whether the software was **developed in South Africa** (Synapse, dmX),
whether the software is **available and supported in South Africa** (VP Expert, PC Plus, EXSYS), and
the price of the software which may vary from approximately R400 for VP Expert to R14000 for PC Plus and more.

3. CONCLUSIONS

The variety of expert system development software available, the wide spectrum of capabilities and features offered by these software and the uncertainty often surrounding a proposed expert system project all contributes to the difficulties encountered when the most appropriate software has to be selected. An approach which may prove useful for the first time user of expert system technology is to acquire a relative inexpensive package, such as VP Expert, and to use this software to become familiar with the technology and to prove, or disprove, the usefulness of the proposed application. With the knowledge and experience gained in this way it should be easier to select a software package for final development. Another approach whould be to acquire a package such as PC Plus, which is significantly more expensive, but provides the first time user with the ability to either exploit or disregard its high-end capabilities depending on whether the goal of the project is to develop a final delivery system or just a prototype.

==========

REFERENCES

1. DREWES A and van JAARSVELD J, 1988, "Criteria for the selection of an Expert System tool", Proceedings of the Expert 88 conference, 7 and 8 July 1988, CSIR, Pretoria.

2. FOSTER W, 1986,"Expert systems for industrial applications", Simulation, vol 46, no 1, January 1986, pp 27-30.

3. FRENZEL L E, 1987, "Understanding Expert Systems", Howard W Sams and Company.

4. FRENZEL L E, 1987, "Crash Course in Artificial Intelligence and Expert Systems", Howard W Sams and Company.

5. HEWETT J and SASSON R, 1986, "Expert Systems 1986", Volume 1 : USA and Canada, Ovum Ltd.

6. HEWETT J, TIMMS S and d'AUMALE G, 1986, "Commercial Expert Systems in Europe", Ovum Ltd.

7. HU D, 1987, "Programmer's Reference Guide to Expert Systems", Howard W Sams and Company.

8. KRUGER P S, DU TOIT R J and RICHTER W, 1988, "Expert Systems", S A Institute of Industrial Engineers, Annual National Conference, Sun City, 31 August 1988.

9. SHEPARD S J, 1988, "Sophisticated Expert", P C Tech Journal, Vol 6, No 7, July 1988, pp 107.

10. WATERMAN D A, 1986, "A Guide to Expert Systems", Addison-Wesley Publishing Company.

11. Software Manuals and Promotional Literature ; Guru, VP Expert, Synapse, Turbo PROLOG, Golden Common LISP, Arity, EXSYS, Savoir, dmX, XI Plus, Twaice, Personal Consultant Plus and Rulemaster