SERIES BUI

B14207

Reprint NO.62

CI/SFB	(Agm) (D)
UDC	681.3:69

TOOLS FOR BRANZ RESEARCH & DEVELOPMENT IN KBS

A.H. Dechapunya

Reprinted from the, *"PROCEEDINGS OF THE SECOND NEW ZEALAND CONFERENCE ON EXPERT SYSTEMS, NZES'87"*, P 59. Conference Centre University of Auckland (2-4 February 1987).

BUILDING RESEARCH ASSOCIATION OF NEW ZEALAND branz

ENVIRONMENT FOR BRANZ RESEARCH AND DEVELOPMENT IN KBS

A H Dechapunya
Building Research Association of N.Z.(BRANZ)

Abstract. This report considers the implications for the computer resource of research and development at BRANZ of knowledge-based systems(KBS) for the purpose of transfer of building technology information to the building industry. The specific aim is to have KBS available on BRANZINFO(computerised information resource).

1.0 INTRODUCTION

BRANZ is a suitable place for developing knowledge bases for several reasons. The Association has close links with the building industry, and can therefore, anticipate potential needs for specific systems. Also, there are experts at BRANZ. Furthermore, the development is a logical extension of the present computerized information system for industry.

KBS development requires large resources in terms of computing, experts and knowledge engineers. BRANZ has experts, and can contract work to people working in the field of knowledge engineering. But it does not yet have the right computing tools, that is, the right combination of software and hardware. Because the fifth generation tools will not be available for at least 5 years, BRANZ must start by considering the tools currently available.

While the Association will not be actively conducting research in KBSs, it must be able to:

- Test the incremental research work.
- Build up expertise in knowledge engineering (KE).
  (The indication is that KE will be very expensive
  once the KBS market grows. Thus it is important that

BRANZ develops this expertise internally.)
- Develop the user interface (graphics etc.). (The aim is to make KBS easier to use so that users can get the right information quickly.)
- Develop KBSs.
- Evaluate AI tools.
- Take advantage of the new technology from the spin-off in AI global projects.

This paper considers the nature of a suitable BRANZ environment for research and development in KBS.

2.0 REQUIREMENT

The following are some of the requirements for BRANZ research and development in KBS:

- o The environment should be one in which knowledge engineers are developing knowledge bases and evaluating AI software tools, and experts are testing or auditing the knowledge bases.

  The teamwork approach would be essential in BRANZ KBS development because of the diverse skills required in KBS development.

- o Knowledge-based programs require adequate power (CPU and address space) to process knowledge bases and inference engines efficiently because processing involves a lot of searching and pattern matching.

- o High performance and high productivity are prerequisites of the system. The rate of development of KBS is influenced by the ease of using the system and the rate of output from it. KBS development involves:

  - Knowledge acquisition
  - Knowledge programming
  - Knowledge compiling
  - Knowledge testing.

  The tools for these tasks include an editor, a compiler, a task builder and a debugger, which must be highly interactive.

  Computing staff are in short supply in New Zealand, and knowledge engineers will be scarce and very expensive. Thus, a good environment in BRANZ for KBS development including highly interactive tools and resource is essential. The tools must not be the bottleneck.

- o The BRANZ computing system must be networked. KBS development is never finished. There is always a need to add more rules, more explanations etc to the knowledge bases. Thus, the ability to distribute the KBS quickly and conveniently within the BRANZ system is essential.

- o The hardware chosen must have been used for AI work and proven to be a good AI tool. It must be able to support commercially available expert system tools because no single software tool can solve all problems.
- o The tools will primarily be used for research and development of KBS. Thus, the structure of the system must be designed for these purposes.
- o Compatibility between development and delivery systems is essential because the end product must be able to be transferred to the delivery system without modification.
- o The long term strategy should be one of building up a team specializing in solving building-technology information problems using AI techniques. To do this, close liaison with the universities is essential.
- o The tools should have a life of at least 3-5 years
- o Price-performance, reliability, maintenance, support, training, consulting and overall BRANZ strategy are other factors which have to be considered.

3.0 TOOLS

3.1 SOFTWARE TOOLS

Expert system development tools can be grouped into:

- Programming languages
- Knowledge engineering languages

It is important to keep in perspective that BRANZ is developing knowledge bases for building technology information. To be in this business, BRANZ must:

- Constantly evaluate available technology.
- Keep developing, refining and auditing knowledge bases as new information becomes available.

3.1-1 Programming Languages

The languages used in programming expert system tasks, range from numerical processing languages(BASIC, FORTRAN, C) to symbolic processing languages(LISP and PROLOG). However, it has been accepted that numerical processing languages are difficult and impractical to use in expert system programming.

LISP and PROLOG have become standards in expert system programming language. Over the past year, much work has been done overseas on developing PROLOG and LISP. However, they are not satisfactory especially on PCs and good ones are very expensive. This has caused some researchers to go back to using conventional language such as C-compiler.

BRANZ, like many Australian AI researchers, have adopted PROLOG as the language for AI research and developments . An interpretive version, up to now, has not provided adequate response times for our needs. We are now looking at a number of PROLOG compilers.

3.1-2 Knowledge Engineering Language

A KE language, a software tool that simplifies KBS development, consists of language and facilities that assist knowledge engineers in the development of KBSs. It is normally written in LISP or PROLOG and can be thought of as a language for expresssing knowledge. It is preferable to use KE languages for KBS developments rather than using LISP or PROLOG.

There are three basic language representations:

- Rule-based (IF-THEN rules).
- Frame-based (using FRAME for reference).
- Object-oriented (using OBJECTS to communicate with one another).

Rule-based representations are the most widely used. Some KE tools have all representations which makes them powerful tools capable of handling many types of problems.

BRANZ experience to date has been with the following KE tools:

1 MARTIN
  Developed by Auckland University.

2. ESP/ADVISOR
  A commercial expert system shell.
  Designed for codes, text animation.

3. CLASS
  Developed by Auckland University for the Firecode contract.

One of the good features of the CLASS tool is its language which is still developing and thus will be further improved. At present the CLASS language does not yet have comprehensive KE development environment(knowledge compiling, knowledge debugging, knowledge editing etc.) . However, CLASS knowledge bases are relatively easy to develop, maintain and audit.

Evaluation of commercial KE languages will be a very important part of BRANZ KBS work. The following is a list of minimum criteria for evaluating KE tools for the BRANZ environment:

o Generality: The ideal situation is to have one system which can be used for all expert systems development. The significance of this is that only one type of KE language needs to be maintained.

o Evaluation: It is neccessary to test that the system can do what it is required to do. The best way is to write a small domain to test the shell. However this process could be time consuming, particularly learning the language of the shell.

o Maintainability: It is essential that the KE language can be maintained. Three reasons for this are: 1) there are always bugs in any software; 2) operating system changes may require some modifying of the KE language; 3) the KE language should be improved with new features and enhanced capabilities. The technology is such that a static KE language would be outdated very quickly, because the AI field is advancing rapidly.

o Language Representation: It has been stated before that BRANZ will not be developing KE languages, its effort is in developing knowledge bases. Thus, the language for representing knowledge is very important for BRANZ. It must be high level and as close as possible to the raw knowledge. A good language representation means that:

- knowledge can be programmed without much reorganization or constraint
- developing knowledge bases is relatively simple
- maintaining knowledge bases is easy
- auditing knowledge bases is easy
- converting knowledge bases to other languages is simple.

o Data representation: The structure of data for classifying properties is also very important. It has to maintain a balance between being too restrictive (useful only for representing simple problems) and too complex (costly in both development and resource use).

o User Interface: User interface can be divided into 2 categories- that provided by the language and that provided by the knowledge base. A good KE language must have the facilities to provide both. In the first category, the interface will be default, eg. menu/command system. The second category will be provided in the language representation where the knowledge engineer can program the interfaces into the knowledge base.

o Control Structure: This is the mechanism by which the system accesses, manipulates, and executes the rules and properties in the knowledge base. At present, most KE languages contain one or more of the following mechanisms:

- backward chaining
- forward chaining

o Development Environment: The rate of development is influenced by the ease of using the system and the rate of output from the system. KBS development involves:

- Knowledge acquisition
- Knowledge programming
- Knowledge compiling
- Knowledge debugging
- Knowledge testing.

o Local Operating System Interaction: A KE language, like any other language, has to be run under an operating system. Thus, the interaction between KE languages and operating systems is an important factor to be considered. This interaction dictates resources use, maintenance, management, etc.

o Explanation Facility: We are deliberately moving further away from the black-box approach of software. Most KE languages have a built-in facility for explaining each step of reasoning to users. Some are better developed than others. This capability helps users to draw their own conclusions about how good the information obtained is. The following is a list of minimum requirements for an explanation facility:

- the ability to communicate in natural language
- There should be two levels of explanation
  :shallow - for superficial explanation
  :deep   - for an in-depth explanation

o Inexact Information: The treatment of inexact information is still very much in the development stage.

Researchers in Australia have been evaluating commercial KBS shells for PCs and some have concluded that developing their own shells is a better strategy than using commercial ones. One of the problems of purchasing commercial shells is that most systems only offer executable images and these cannot be changed or modified to make them suit a specific environment.

3.2 HARDWARE TOOLS

The machines used for KBS research and development are broadly divided into two categories:

- Multi-user environment systems.
- Single user AI workstations.

Both types are generally 32-bit word machines with plenty of memory(8-20 Mbytes).

A multi-user environment system offers flexibility at reasonable cost per user. It allows both traditional and KBS work to be done together. This is especially valid in the BRANZ environment in which teamwork is essential. A 32-bit machine with 8-20 Mbytes of physical memory should have adequate address space for KBS processing. It is anticipated that about 4 Mbytes are required for one knowledge engineer(KE).

Single user AI workstations are designed specifically for AI work with highly interactive program development facilities (editing, compiling, debugging etc) supporting large physical memory.

The comparison between AI workstations and multi-user systems is shown in Table 1.

AI Workstation	Multi-user System
o Designed for symbolic processing	o Conventional machine
o Extremely high cost per user	o Relatively low cost per user
o High performance	o Lower performance
o Supports bit-mapped graphics	o Can support bit-mapped graphics
o For AI work only	o Very flexible
o End users require AI workstation	o Flexible
o Low capacity of on-line storage	o High capacity of on-line storage
o Limited software tools	o A large range of software tools
o Access to the system is local	o Can be accessed through PACNET.
o Team work is limited	o Team work capability

Table 1 Comparison of AI workstation and Multi-User systems

Both single and multi-options are single processing machines (one CPU per one machine for central processing). They can be considered as filling a gap for the next 5-10 years while the world is waiting for a fifth generation computer, with multiprocessing capabilities, designed specifically for symbolic processing. It is most unlikely that there would be a market for AI workstations in New Zealand because of their cost at present.

Some researchers in Australia have been using PCs for developing KBS for marketing to the large number of PC users. However they found that PCs were not powerful enough for serious KBS research and development work. Some of the reasons were:

- PCs were, and still are, 16-bit machines. (This is changing. For example, MAC+ and Amiga could become powerful workstations.)
- PCs do not have good memory management.
- PCs software support and bug fixes are inadequate.
- The backup facility on PCs is inadequate.

- PCs do not support good program development facilities.
- PCs are designed for end users.

4.0 CONCLUSION

BRANZ is in the process of implementing the environment for long term KBS research and development. This technology is advancing very rapidly and thus careful strategy is neccessary. The paper considers the nature of a suitable BRANZ environment for research and development in KBS. Highlights of the considerations are:

Flexible environment: An environment in which KBS and conventional work can coexist. An environment that fits teamwork approach. An environment in which the products can be distributed easily and conveniently throughout the BRANZ local area network. To achieve this, multi-user systems with networking capability are neccessary.

High performance system: The combination of hardware and software must be such that they will not be a bottleneck in the KBS development. This suggests, for example, that a 32-bit word machine, 8-20 Mbytes, with a compiler based KE language is necessary.

Acknowledgements:

This paper would not be possible without the teamwork structured within BRANZ. The support from BRANZ management and colleagues is gratefully acknowledged.

BIBLIOGRAPHY

Buis M, Hamer J, Hosking J.G. and Mugridge W.B. 1986.
An Expert Advisory System For A Fire Safety Code.
Report No. No 35-30-002.

Buis M, Hamer J, Hosking J.G. and Mugridge W.B. 1986.
An Expert Advisory System For A Fire Safety Code.
The Second Australian Conference on Applications of Expert Systems, Sydney

Buis M., The Construction Of Expert Systems. 1986.
(Msc. Thesis, University of Auckland).

Gero, J.S. 1985. Knowledge Engineering and Expert Systems,
National Engineering Conference, IEAust.

Gevarter, W.B. 1982. An Overview Of Expert Systems,
NBSIR 82-2505.

Hayes-Roth,F., Waterman,D. and Lenat,D.(editors) 1983.
Building Expert Systems.
Addison-Wesley Publishing Company, Canada

Marksjo, B.S. 1985.
Expert Systems in the Building and Construction Industry.
COMTEC Seminars, Morphetville.

Parkinson, R. 1986. The Rdb/OPS5 Precompiler
Proceedings of DECUS, Australia

Richer, M.H. 1986. An Evaluation of Expert System Development Tools.
Knowledge System Laboratory, Standford University,
Report No. KSL 85-19

Shannon, T.C. 1985. Artificial Intelligence
Proceedings of DECUS, Spring, USA

Waterman, D.A. 1986. A Guide to Expert Systems
Addison-Wesley, Canada

Weiss, S.M. and Kulikowski C.A. 1984.
A Practical Guide to Designing Expert Systems.
Rowman and Allanheld, USA

Whitney, R.S., and Dechapunya, A.H. 1985.
The building of a system.
Interface,(Sept. 1985),pp. 77-82

BUILDING RESEARCH ASSOCIATION OF NEW ZEALAND INC.
HEAD OFFICE AND LIBRARY, MOONSHINE ROAD, JUDGEFORD.

The Building Research Association of New Zealand is an industry-backed, independent research and testing organisation set up to acquire, apply and distribute knowledge about building which will benefit the industry and through it the community at large.

Postal Address BRANZ, Private Bag, Porirua

branz

ISSN 0111–7459