Building Knowledge Systems: Toolkits and Case Studies

John K.C. Kingston

AIAI-TR-102

21 February 1992

This paper was presented at a workshop entitled "Quantitative Methods, Supercomputers and AI in Finance", organised by Unicom Seminars Ltd. This workshop was held at the US Embassy, Grosvenor Square, London on 5-6 February 1992.

Artificial Intelligence Applications Institute
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN
United Kingdom

1 Introduction

Artificial Intelligence has been the subject of much talk – and not a little overselling – in the financial community. Many businessmen have now heard the term, and have a vague idea that AI systems are programs which aim to replace people and to create human-free offices by the year 2001. Those who have got as far as attending a presentation on the subject have heard many stories of technologically successful applications, with huge potential benefits for their business - but the perceptive have noticed that this potential is not always realised.

Most of the AI applications which are successful in the financial sector are *knowledge based systems* (KBS). Tasks which involve diagnosis, classification, assessment, identifying patterns, planning, or simply organising large amounts of information have all proved to be particularly amenable to KBS solutions. In the UK, KBS have been used to tackle a wide variety of tasks which people found difficult or time-consuming to perform; in other areas of the world, similar tasks have been tackled, but with different emphases which reflect the differing environment in which the financial communities around the world operate.

When a company decides to undertake the development of a KBS, one of the most significant – and difficult – decisions in the project is the choice of a programming tool. There are literally hundreds of programming tools to choose from, with widely varying price tags. The decision must be based on performance, facilities offered, hardware compatibility, and availability of support as well as on price.

This paper classifies the range of tools available into categories, and gives examples of the use of tools from each category in KBS applications in the financial sector

2 Software used to develop KBS applications

The four major categories of tools which are used to develop knowledge based systems are:

- Shells
- Mid-range toolkits
- Top-range toolkits
- Symbolic languages

Shells are packages which provides a method of representing knowledge and an 'engine' for drawing inferences from that knowledge. The most common representations are

- *IF-THEN rules*, with inferences being drawn by *forward chaining* or *backward chaining* through the rules
- *frames* with inferences being drawn using *object-oriented programming.*

Shells typically run on PCs with a minimum 512K of RAM. Examples of shells include Crystal, Level Five, Guru, and Xi Plus.

Mid-range toolkits have become available in the last few years to fill a gap in the market between shells and the top-range toolkits. Mid-range toolkits typically provide the ability to mix rule-based and object-based representations, and have some of the debugging facilities of toolkits, but lack the more sophisticated (and processor-hungry) features. Most of these mid-range toolkits will run on 386 PCs; some will run on Unix workstations, and some are able to run on mainframes. Mid-range toolkits include ART Enterprise, KAPPA, and Nexpert Object; the forerunner among those which run on mainframes is the Aion Development System (ADS).

Top-range toolkits provide all the facilities mentioned so far and allow them to be mingled freely. These toolkits also provide graphics which are integrated with the knowledge representation, built-in facilities for hypothetical reasoning, and a wide variety of debugging interfaces. However, toolkits typically require an 'engineering' workstation (e.g. Sun workstations) to run on. The market leaders in top-range toolkits are Inference ART and KEE.

Symbolic languages require the user to define and implement both a representation of knowledge and a technique for drawing inferences; they are thus generally slower but more flexible for developing KBS applications than any shell or toolkit. The two most popular symbolic languages are Lisp and Prolog. Both have been used for many years; Lisp has been the favourite of the AI research community in the USA, while Prolog has enjoyed a profile at least as high as Lisp's amongst AI researchers in the UK and parts of Europe.

These tools are intended for a knowledge engineer [1] to use. The idea is that these tools provide a framework, or a "shell", of a KBS. All the knowledge engineer has to do is to extract knowledge from an expert, formalise it into rules and objects, and then 'plug' these rules and objects into the tool. The tool then organises and performs reasoning based on these rules and objects.

Recently, there has been a trend towards **application-specific** tools. These are usually shells, with extra facilities designed especially for particular types of application. For example, there are tools available which are specialised for developing KBS which support help desk operators, such as Path Builder, which permits the system developer to create *situations*, which are possible problem areas to diagnose, *branches* which are questions and answers which determine which problem

[1] "Knowledge Engineer" is the title given to analyst/programmers in the KBS field

area to focus on next, and *final solutions*; and tools which are specialised for KBS tackling tasks involving design, such as IDOTS (the Intelligent System for Design of Telecommunications Systems).

It is not impossible to develop a KBS in a more conventional programming language, such as C or Cobol. However, the tools described above are so convenient that most KBS which are implemented in a conventional language were originally developed in a KBS tool, and then translated.

3 Case Studies

In this section, five KBS applications from the financial sector – one for each of the categories of tool listed above – will be looked at.

3.1 VATIA and VIVA: Encoding Legislation

VATIA (the VAT Intelligent Assistant) is intended to help auditors in the field remember and examine all the VAT (Value Added Tax) regulations relevant to their client. Since KBS are not well suited to the task of interpreting ambiguous regulations, VAT experts are called upon to discuss any interpretation problems. VATIA has proved very useful because the regulations concerning VAT are so complex that simply knowing which regulations are applicable is beyond most auditors. VATIA thus relieves a bottleneck by making VAT knowledge available to those who need it, when they need it.

VATIA was developed by Ernst & Whinney (now Ernst & Young) in 1988. It has been in regular use since then, and has been sufficiently well received that Ernst & Young are rewriting the system for use on Macintoshes rather than just IBM-compatible PCs, and are taking the opportunity to add further knowledge about VAT in specialised industries, such as the financial sector.

VIVA is Custom and Excise's equivalent of VATIA (Customs & Excise administrate VAT in the UK). This system is aimed at assessing the VAT dues of small traders in metropolitan areas. There are around 100,000 visits to such businesses in the UK, accounting for a significant proportion of the Excise's workload. The system, built with the assistance of Ernst & Young and their KBS methodology known as STAGES, has gone through an acceptance test where it was found to save about 40 minutes out of every 6 hour visit, and to generate considerable additional revenue.

VATIA was built using the shell Crystal, from Intelligent Environments Ltd, and VIVA was built using Xi Plus, from Inference Europe Ltd. The main advantages of shells for these projects was portability; that is, the delivered application was sufficiently compact to run on average-sized portable computers. Shells normally offer only a rule-based representation of knowledge; this proved to be adequate, because rules are a good way of representing legislative knowledge.

3.2 X-MATE: Assessing Mortgage Applications

The X-MATE system [?] was developed by Hewlett Packard's Knowledge Systems Centre, with assistance from AIAI, for a large UK building society. Its task was to assess the likelihood of mortgage applicants meeting their loan repayments. It is implemented in KAPPA-PC, from IntelliCorp, which is a mid-range toolkit.

The building society's problem was that the percentage of defaulters was too high, and it was difficult to enforce quality control on acceptance of applications because, within certain guidelines, the acceptance of applications was almost entirely at the discretion of the local branch manager.

Knowledge was obtained from experts at the building society's Head Office, who had considerable experience of mortgage application assessment; some had particular experience in analysing defaulting cases. When it came to choosing a structure for the system, the project team drew on the library of interpretation models associated with the KADS methodology. Three categories of 'typical' high risk applicants were identified. These categories were: applicants who showed evidence of having difficulty in controlling a budget; applicants who were vulnerable to reduced income if circumstances should change; and applicants who deliberately included false information in their application.

X-MATE works by using heuristic rules which examined a mortgage application, and attempted to identify features of typical high risk applicants in that application. An example would be:

```
IF the applicant has two jobs

AND the mortgage is dependent on more than 50% of the salary from
his second job

THEN there is a high risk (that changing circumstances might reduce
his income)
```

While a considerable amount of information about an applicant is available from the mortgage application form, some further information is needed to perform a thorough risk analysis of each application. X-MATE is designed to minimise the amount of time or money required to check extra information, by only performing further checking on those applicants who show a high degree of risk from the analysis of the mortgage application form.

In tests on applications from the previous few years, X-MATE was able to identify 50% of all the applications which have since been repossessed without any false positive identifications, using only the checks on the application form.

KAPPA-PC supplies both rules and objects, and both of these were used in the X-MATE system: objects to represent the available information about the applicant, and rules to perform the risk analysis. KAPPA-PC runs on a 386 PC under Windows 3, and this enabled the development team to access another package to develop the user interface for X-MATE, using DDE (Dynamic Data Exchange). KAPPA-PC has graphic facilities of its own, and with the latest release of KAPPA-PC, the external package is no longer needed.

3.3 TARA: Assistance in Currency Trading

Trading on the foreign currency markets is a clear example of a task where information overload is a problem. Large amounts of money can be made if the future prices of currencies can be predicted correctly. No method of prediction is completely accurate, but political, economic and market factors often provide an indication of possible price movements. As a result, currency traders are bombarded with so much information about world events and market activity that they cannot make use of it all.

Currecy traders have different strategies for predicting prices. Two of the most popular are *technical analysis*, which uses statistical analysis of past patterns of price movements to try to identify similar trends in current price movements, and *fundamental analysis*, which tries to predict the supply and demand of a commodity by studying economic and political factors as well as market psychology. Both of these strategies required considerable expertise on the part of a trader to interpret the information provided in order to make a "buy/don't buy" decision.

TARA was implemented by Manufacturers Hanover Trust (MHT) to assist traders in the analysis and interpretation of information. It primarily uses the technical analysis approach. It identifies significant rises and falls in currency prices based on a live data feed, and uses these data to maintain a number of statistical models. It also contains a knowledge base for identifying patterns and spotting trends in these models. When the price crosses a trend line, rules fire and a "buy" or "sell" recommendation is given.

TARA also makes some use of fundamental analysis, in that it knows that certain technical analysis models are only good predictors in certain economic conditions. It does not take account of business or political factors, however.

The knowledge base of TARA was implemented using KEE from IntelliCorp Inc. KEE is a powerful (and relatively expensive) top-range toolkit which requires powerful and expensive hardware. KEE cam handle the large amount of processing which TARA has to perform (30 models may be active, and the knowledge base has to monitor each of them). KEE was also a good choice for TARA because, at the time when TARA was built, there were few mid-range toolkits available. Today, however, less and less applications are using top-range toolkits such as KEE because of the availability of mid-range toolkits. Those applications which do use top-range toolkits are those which need a lot of knowledge processing power, or those which require the specialised features of the toolkits, such as those which require investigation of multiple alternative scenarios.

Another difficulty with top-range toolkits is integrating the hardware on which they run with other hardware. This is particularly true for some of the earlier users of top-range toolkits (such as TARA), which run on highly specialised Lisp-based workstations. In fact, TARA is no longer used by MHT, because it cannot integrate with the company's VAX/VMS systems.

3.4 The Intelligent Banking System: Processing of telexes

Getting computers to understand 'ordinary', or natural, language has been a research topic in AI for many years now There are many problems because much of our comprehension of language is based on common sense knowledge However, in limited domains of knowledge, such as the area of processing international banking telexes (which use a standard format), there has been some success.

The International Banking System (IBS) is an AI system which was developed for Citibank, New York [**?**]. It can process messages relating to

- Funds transfer
- Letter of credit issuance
- Letter of credit reimbursement
- Funds transfer problem enquiry

The funds transfer message processing module was the first to be developed. Funds transfer messages normally use only a subset of English. However, because of time pressure on the sender, common problems include:

- Abbreviations
- Spelling mistakes
- Necessary information is omitted
- Unnecessary information is supplied

In addition, the information given by a particular word is highly context dependent - that is, it depends on the other words in the message. The result is that there are thousands of possible meanings for a telex, based on different combinations of words. This kind of highly combinatorial problem is often better suited to an AI solution than a conventional solution. Also, the system must reach a decision in under 60 seconds.

The IBS system combined a number of AI approaches to language parsing (flexible parsing [?], case-frame grammars [?] and semantic grammars [?]) to arrive at a final system which was able to identify abbreviations and spelling mistakes, ignore unnecessary information, and identify incomplete sentence fragments.

The IBS system also incorporated a rule-based knowledge based system which inferred values and constraints on values using combinations of phrases, and made decisions based on the overall message content.

The IBS system was written in Lisp. There are at least two reasons why Lisp was useful for this project. The first is that shells and toolkits have few facilities to help with the construction of natural language processing systems, while there are recognised techniques for doing so using Lisp or Prolog. The second reason was for speed: the rule-based system which is part of IBS was originally implemented in a KBS tool, but was rewritten in Lisp, which produced a tenfold increase in speed.

3.5 INCA: Intelligent Networking Control Assistant

SWIFT - the Society for Worldwide International Financial Telecommunications - has seen the use of its networking services grow tremendously in the last decade. Transactions on the system have risen from 50 thousand per day in 1977 to 1 million per day in 1991. As a result, it is absolutely critical to SWIFT's business that their telecommunications networks are kept up and running at all times.

The network sends information about network events to a control centre in Brussels, where operators must decide on the meaning of the events, and what action to take, if any. The task of interpreting network events is not an easy one, and so SWIFT decided to build a knowledge based system to help with the interpretation. The INCA system was the result. It took a year to build, and now filters out 97% of network events before the operator sees them, thus considerably reducing the load on the operators.

INCA was implemented using DANTES [?], on Texas Instruments Explorer workstations. DANTES is a KBS tool targetted at real-time network monitoring applications. The real-time constraints mean that DANTES must run fairly fast. DANTES achieves this by making optimisations based on its limited application domain: it has a specialised conflict resolution technique, it filters the data which is used by the rule system, and it does some of its own memory management.

4 A Future Look

Over the last five to ten years, the attitude of practical solution builders to KBS has changed markedly. They no longer want to know what KBS are, but how they can be applied—developed and deployed—in a controlled manner. As a result, there has been a great deal of activity directed towards formulating explicit methods (or "methodologies") for developing KBS.

Conventional software tools can facilitate KBS development. Standard project management systems, and indeed drawing packages, have a role. Moreover, several methods, particularly those based on established conventional software development methods, advocate the use of conventional CASE tools, where possible, to support the use of data flow and entity-relationship diagrams, and the documentation of interviews.

In addition to simply using such tools, there is scope for enhancing them for KBS purposes, and several CASE tool vendors are exploring this possibility. For example, Aion (who market ADS, a mid-range KBS tool) have a long term strategy to link CASE tools to ADS. Last summer, they demonstrated a link to KnowledgeWare's Analysis and Design Workbench (ADW), with data and process descriptions from ADW being downloaded into ADS to produce a running system. They hope that such an approach supported by generic methods can be used for the rapid development of conventional, KBS and mixed systems.

There are also a number of tools intended to support particular KBS development methods. Examples include:

- Alpha/PRO from Information Systems Associates, supporting their own Alpha/DS method.
- K-STATION, developed and marketed by ILOG, which has been available since June 1990. This tool supports a specific knowledge engineering analysis and design method, known as Knowledge Oriented Design (KOD), which was developed by Claude Vogel from CISI Ingenierie.
- Shelley, a non-commercial tool intended to support the analysis of expertise and the design stages of the KADS methodology.
- A PC-based support tool for KADS, which is currently under development.

Finally, there are tools which are intended to help with with knowledge acquisition; notable examples are

- ProtoKEW, which allows a number of structured knowledge acquisition techniques to be applied without a knowledge engineer being present
- Nextra, which translates acquired knowledge into rules suitable for the Nexpert Object toolkit.

K-STATION and Shelley are also primarily concerned with knowledge acquisition, and could be classified as knowledge acquisition tools.

While few of these tools are commercially available as yet, and those which are are not yet widely used, it is likely that these tools, and others like them, will have a significant impact on the development of KBS applications over the next five years.

For more information on methods for KBS, see [?].

5 Conclusion

We have seen some of the areas in which AI has been applied in the financial sector, and we have seen that different tools have different advantages in terms of flexibility and efficiency. Applications for portable computers should probably be written in a shell; applications involving natural language processing should look towards a symbolic language; and applications in network diagnosis or help desk support should consider some of the application-speciific toolkits now available.

Top-range toolkits used to be "the" tools for writing KBS. These days, they are worth considering if the application will have to deal with large amounts of data, or if the application involves predicting possible future scenarios; however, their place is increasingly being taken over by the mid-range and mainframe toolkits. This is particularly true in the UK financial sector, whose investment in AI has been conservative in relation to other sectors, and in relation to the financial sector in the USA and Japan.

The future is likely to see methods for KBS development becoming increasingly prominent, particularly as supporting computer tools are developed.

Addresses

```
ADS:
Trinzic Ltd
 Portmill House
 Portmill Lane
 Hitchin
 Hertfordshire SG5 1DJ
 0462 422525

Alpha/PRO:
 Information Systems Associates
 The Estate Office
 154 Saxon Drive
 London W3 0NT
 081 993 8080
```

Crystal:
Intelligent Environments
 Northumberland House
 15-19 Petersham Road
 Richmond
 Surrey TW10 6TP
 081 940 6333

Level Five:
Information Builders Inc.
 Station House
 Harrow Road
 Wembley
 Middlesex HA9 6DE
 081 903 6111

Inference ART, ART Enterprise, Xi Plus:
Inference Europe
 31-37 Windsor Road
 Slough
 Berkshire SL1 2ED
 0753 811855

K-STATION:
ILOG S.A.
2, av. Gallieni, BP 85.
94253 Gentilly Cedex
France
Tel: +33 (1) 46.63.66.66
Fax: +33 (1) 46.63.15.82
KEE, KAPPA, KAPPA-PC:
IntelliCorp Ltd UK Office
 10 Jewry Street
 Winchester
 Hampshire SO23 8RZ
 0962 735348

Nexpert Object, Nextra:
Neuron Data
 34 S. Molton Street
 London W1Y 2BP
 071 408 2333

ProtoKEW:

Dr N. Shadbolt
Department of Psychology
University of Nottingham
University Park
Nottingham
NG7 2PD
0602 484848