Research and Development in Expert Systems VI

Proceedings of Expert Systems 89, the Ninth Annual Technical Conference of the British Computer Society Specialist Group on Expert Systems
London, 20-22 September 1989

Edited by

Nigel Shadbolt
Artificial Intelligence Group, Department of Psychology
University of Nottingham

The right of the University of Cambridge to print and sell all manner of books was granted by Henry VIII in 1534. The University has printed and published continuously since 1584.

Published by
CAMBRIDGE UNIVERSITY PRESS
on behalf of
THE BRITISH COMPUTER SOCIETY
Cambridge
New York Port Chester Melbourne Sydney

Published by the Press Syndicate of the University of Cambridge
The Pitt Building, Trumpington Street, Cambridge CB2 1RP
40 West 20th Street, New York, NY 10011, USA
10 Stamford Road, Oakleigh, Melbourne, Australia

First published 1989

Printed in Great Britain at the University Press, Cambridge

Library of Congress cataloging in publication data available

British Library cataloguing in publication data available

ISBN 0 521 38477 X

Expert systems - a natural history

Nigel Shadbolt
Artificial Intelligence Group
Department of Psychology
University of Nottingham
Nottingham NG7 2RD

Abstract

This paper examines the origins, current state and future prospects for expert systems. The origins are traced from the schism with classic Artificial Intelligence. The characteristics of early expert systems are described and contrasted with more recent developments. A number of influential forces operating on present day systems are reviewed. The future trends in the evolution of expert systems are discussed.

1 Introduction

It has always seemed an interesting thought - can we talk about the *natural history* of a machine? In particular, can we use analogies from the natural world when thinking about the computer? There are similarities. We can see more complex computing devices developing from simpler ones. Computers occupy habitats, ecological niches to which they are more or less suited. Various forces operate to select the best of these. Some change, others find a role and remain unchanged, others perish. One can carry the analogy some way. Although it is interesting to note that there are those who suggest it is computing that offers insights into the natural world (Dawkins 1986). The genetic substrate, DNA, is after all an information encoding mechanism *par excellence*.

This paper presents a natural history for one evolving species of computing systems, namely - expert systems. What can we say of their origins, present condition and future prospects?

2 Origins

An important part of the natural historian's work is to establish the lineage or *phylogenetic* history of any species. To which *genus* does it belong? What closely related species are there? When did it become clearly distinct from these related species?

In the case of expert systems, there is agreement that they developed out of a branch of the computing genus known as artificial intelligence (A.I.). The early origins of A.I. systems can themselves be traced back to the mid '50's. Expert systems appeared as a distinct line around the late '70's. However, there is considerable argument about just when and where the divergence took place.

Many would regard MYCIN (Shortliffe et al 1973, 1976) as the first and original expert system. In part, this is because it made evident to the rest of the world the fact that something new had evolved. MYCIN came out of work conducted on the Stanford Heuristic Programming Project. The system assisted doctors in the selection of an appropriate course of treatment for patients with bacteremia, meningitis and cystitis.

It is not the only contender for the accolade *first expert system*. Some would argue in favour of the DENDRAL system (Buchanan et al 1978). It too was developed at Stanford and its function was to infer the molecular structure of unknown compounds from mass spectral and nuclear magnetic response data. Another contender is MIT's MACSYMA system (Martin et al 1971). This system, which is today in widespread commercial use by engineers and scientists, assists in a range of mathematical tasks. It uses mathematical expertise to recognise a user's problem and then selects appropriate methods and techniques of mathematical analysis. The foundation for MACSYMA was the MATHLAB 68 system (Engelman 1971) which originated in the late '60's.

Whichever one is accorded the title all these early systems share strong similarities. They were large, they incorporated substantial amounts of heuristic knowledge. They were built with applications in mind. They were American, programmed in LISP and many people saw them as the new wave in A.I.

Research on other large expert systems continued through the late '70's across a range of application domains. These are now regarded as landmarks in the development of expert systems. CASNET (Szolovits et al 1978, Weiss et al 1978) was a large medical expert system that diagnosed and proposed treatments for disease states related to glaucoma. One of the distinctive features of CASNET was that knowledge was represented in a semantic network that attempted to provide a causal-association model of symptoms, disease processes and treatments. The PROSPECTOR system (Gaschnig 1982) helped in the interpretation of geological data and attempted to assess the likelihood of finding various types of mineral deposits. Its knowledge representation was rule and network based. It used certainty factors and probability propagation methods to encode the idea of confidence in evidence and certainty in conclusions. XCON/R1 (McDermott 1980) configures DEC VAX computer systems. It decides upon the components needed to produce an operational system given a customer's order. It is a constraint driven system - it has knowledge about what components can go together, what the constraints at the installation site are etc. It uses this knowledge, expressed in a rule-based format, to reason forward from the constraint data to a configuration that satisfies the constraints. This commercial system is still in use and is considered one of the most successful in the history of expert systems.

It is interesting to note that early versions of these systems shared many of the characteristics of the progenitor systems discussed earlier. They had large knowledge bases, ran on large computers, consumed man years of research development, were American built and they all received a lot of publicity. It is also interesting to note that they employed a wide range of techniques to represent knowledge and reason with it. In some of the earliest systems we see mixed representation methods; rules, nets and frames. However, it is also salutary to note that four out of the six named systems only reached what is termed the *prototype* stage. Clearly this was not yet a completely mature and exploitable technology.

A new direction in the evolution of expert systems was provided by the work done on EMYCIN (van Melle et al 1979, 1984). The acronym which stands for *essential MYCIN* indicates the skeletal nature of the system, which was MYCIN stripped of its domain knowledge. What one is left with is a rule-based representation language which uses a backward chaining control regime, certainty handling methods, and automatic explanation facilities. This abstraction away from the problem domain left a clean kernel system. The system was restricted to representing and reasoning with knowledge in a particular way. However, it provided an uncomplicated tool with which to build applications. In particular, it could be used to build diagnostic and classification systems. The age of the expert system *shell* had dawned.

Once the template for such shells became apparent they proliferated. Early shells tended to be primarily rule-based. They also began to migrate from costly hardware to PCs, and from LISP to a variety of other programming languages. The simplicity of a kernel rule-based shell made this process straightforward.

The history of some of these early shells illustrates some interesting differences in respect of how expert systems technology developed on either side of the atlantic during the first part of this decade. The original EMYCIN was written in INTERLISP and ran on DEC mini computers. Another early shell was KAS (Reboh 1981). The concept was very close to EMYCIN, the difference was that KAS was the PROSPECTOR system stripped of its geological domain knowledge. Again the system was written in INTERLISP and ran on mini computers.

The development of early US *purpose built* shells is exemplified by Teknowledge's products S.1 and M.1. The S.1 shell was an expert system building tool based on rule-based representations. Its basic control mechanism was backward chaining. However, it also supported alternative representational methods, including frames and a procedural language. The cost of such added functionality in the first part of this decade was that its original INTERLISP implementation ran only on specialised Xerox workstations. The more modest M.1 system was capable of running on IBM PC hardware, it was rule-based and built using a PROLOG backward chaining architecture. It had no other representational capabilities. In fact, for the US M.1 was rather an exception. A survey of US products in the early '80's shows that the main preoccupation was the provision of quite specialised knowledge engineering software which offered a wide range of capabilities. However, one paid a high price both for the software and the hardware.

The UK scene was characterised by the appearance of a number of inexpensive, PC-based shells. Many of these owed much to the effort that had gone into producing PC PROLOGS. These implementations offered a natural vehicle for shells. PROLOG requires little augmentation to function as a shell. In fact, one of the first products APES (Augmented PROLOG for expert systems, Hammond 1982) was just such a shell.

Other shells appearing at or around this time attempted to hide their PROLOG internals from the user. The recurrent feature of these systems was rule-based representations, backward chaining control of reasoning, and often an add-on which provided a means of representing and reasoning about uncertainty.

A great deal of interest was shown in this *first generation* of shells. In the UK A.I. had been progressing steadily during the late '50's and '60's. Unlike the US it never received much governmental support, but by the early '70's a number of centres of excellence and had made significant contributions to A.I. in fields as diverse as theorem proving and robotics. The UK A.I. scene suffered something of a reverse with the publication of the Lighthill report in 1973. For a while there was a real chance we would lose our research groups completely. However, the groups hung tenaciously on, and by the beginning of the '80's the environment was changing. One of the first signs of this changing environment was the founding by Professor Donald Michie in 1980 of the BCS SGES. This specialist group's goal was to bring together a community of individuals interested in the potential of expert and knowledge based systems. The 1982 Alvey Report led to the 350 million pound Alvey programme, the EEC was also getting its ESPRIT programme underway. These initiatives released large amounts of funds into IKBS and its supporting technologies. This largesse was prompted in large part by the inception of the 1981 ICOT or Japanese Fifth Generation initiative. Suddenly it seemed everyone regarded A.I. as the key to future economic success.

As awareness increased and money poured in, more and more companies and academic sites became active in the area. As momentum gathered and expectations rose perhaps it is not surprising that a sense of what the technology could realistically achieve was not always retained.

Nevertheless, this interest together with the emergence of an expert system shell technology led to real progress. Only a few years later Professor Alan Bundy (1987) in a key note lecture at ES87 was able to observe

> the UK expert system community has been very successful in the development of small scale, commercial, rule-based, expert systems. A typical example is a fault diagnosis system for a piece of specialised hardware, consisting of a set of less than 100 rules, running on a PC, in one of the many commercial shells. Part of the success consists in the unexpected (to me anyway) discovery of a large number of commercially interesting problems which yield to such a simple mechanism.

What Bundy was describing in terms of our natural history analogy is equivalent to the *speciation* of simple but effective forms - an explosion of simple systems to fill the many niches that exist.

He also referred to a preoccupation with diagnostic systems. So distinguishing one particular generic class of expert systems. There are in fact a range of types of problem solving we could imagine expert systems performing. A number of classifications have been proposed, the one below derives from Waterman[1] 1986. Each of these different types is proposed to have a different underlying problem solving structure. This is sometimes called the *inference level*. And can be regarded as a type of knowledge in its own right. It is knowledge about how components of expertise are to be organised and used in the overall problem solving system.

Type	*Description*
Diagnosis	Inferring system malfunctions from observables
Interpretation	Inferring situation descriptions from sensor data
Prediction	Inferring likely consequences of given situations
Design	Configuring objects under constraints
Planning	Designing and sequencing actions
Monitoring	Comparing observations to plan vulnerabilities
Debugging	Prescribing remedies to malfunctions
Repair	Executing a plan to administer a prescribed remedy
Instruction	Teaching of any knowledge level component
Control	Governing overall systems behaviour

Table 1: Generic Problem Solving Categories

Inference level knowledge is, of course, only one variety of knowledge one is likely to find in expertise. Using a classification due to Weilinga and Breuker (1986) we can distinguish three other general sorts of knowledge; strategic, task and domain level knowledge.

Strategic knowledge monitors and controls the overall problem solving. This can have to do with the way resources are used. What to do if the proposed solution fails or is found to be inappropriate in some way. What to do when faced with incomplete or insufficient data. Task level knowledge is sometimes called *procedural* knowledge. This is knowledge to do with how goals and sub-goals, tasks and sub-tasks should be performed. Thus in a classification task there may exist a number of tasks to perform in a particular order so as to utilise the domain level knowledge

[1] This is in fact a rather coarse characterisation and if we take just the category of diagnosis it can be further analysed into sub-types; heuristic diagnosis, systematic diagnosis through causal tracing, systematic diagnosis through localisation.

appropriately. By domain level here we mean its narrow sense - knowledge that *describes* the concepts and elements in the domain and relations between them. This sort of knowledge is sometimes called *declarative*, it describes what is known *about* things in the domain.

Any field of expertise is likely to contain, to greater or lesser extents, elements of domain, task, strategic and inference knowledge. At any particular knowledge level the information may be explicit or implicit in an experts' behaviour. Thus in some domains the experts may have no real notion of the strategic knowledge they are following whilst in others this knowledge is very much in the forefront of their deliberations. Also, of course, the requirements on a system about how far it needs to implement these various levels will vary. But they can become evident even in *modest* first generation applications. Moreover, it is acknowledged that significant reasoning about problem domains requires more than just modelling simple relationships between concepts in the domains - it requires causal models of how objects influence and affect one another, models of the processes in which objects participate. This is a hard problem. And often the limitations of first generation expert systems means that sophisticated domain models cannot be supported. It was this point that Bundy went on to discuss in his ES87 address.

> UK knowledge engineers have also been active in building much larger expert systems, with hundreds or even thousands of rules. In addition, they have experimented with alternative knowledge representation and reasoning techniques, e.g frames, objects, semantic nets, etc. This use of large scale expert systems and of alternative and/or multiple knowledge representations has been more typical in the US market, but both are becoming more important here.

In a lecture delivered to the previous years conference, Steels (1986), had also pointed out the need for us to look beyond first generation systems. He pointed out some of the limitations of first generation systems. First generation systems tend to rely on behavioural heuristics, *if X is observed do Y*. These are *surface* models of *performance*, with no *deep* model of *competence*. A second generation system would have an additional component in the form of a deep model which gives the system an understanding of the domain over which the heuristics operate. In fact Steels argues that an important source of inspiration for this second generation component is to be found in A.I. work in areas such as qualitative reasoning (Price and Hunt 1989 this volume, Suguya 1989 this volume). This enhances the problem solving of any system by modelling the domain principles, its causal and functional properties. Associated with the need to provide deep models Steels remarked on the need for powerful methods of building KBSs. One category of tools would help in the knowledge acquisition process - a process that needs to be more sophisticated than ever if deep models are to be implemented.

We can discern then a number of recurrent themes leading up to the present; a substantial number of modest rule-based applications in place, a concern to provide a methodology for the knowledge engineer, the emergence of more powerful shells, the recognition of an impending second generation of expert systems. The questions now relate to the current well-being of the expert system species. Will the dissemination of small expert systems and shells continue? Are knowledge engineering methods available? Can the technology be scaled up to large applications? When will commercial second generation expert systems arrive?

3 Current conditions

The Alvey programme for all its faults, real and imagined, was a great catalyst. It succeeded in forging a bridge between academe and industry. And although as Bramer (1986) remarked many of the best researchers were led into endless rounds of grant writing and administration, there was real disappointment when the Bide report, which suggested a comprehensive follow on to Alvey, was largely ignored. In its place the DTI's Information Engineering Directorate (IED) has attempted to continue what Alvey began. For all its efforts it is hamstrung by the funding conditions it has to work by. Usually a number of partners, commercial and academic, will get together to form a consortium to carry out a programme of work. The consortium can apply for up to 50% of its total eligible costs, academic partners however receive 100% of their costs and this is taken from the 50% of costs the IED will pay to the consortium. The industrials then receive the remaining amount of money in proportion to their original costs. The effect of these rules is to make academic participation unattractive to industrial partners. Moreover, the absolute proportions awarded tend to mitigate against small to medium sized company participation.

The SERC is the other source of funding for academics. However, money is so tight that the SERC only manages to fund a minority of its alpha rated (technically excellent) research proposals in computer science.

Meanwhile the infrastructure that was so painfully built-up in Alvey is gradually coming apart. As yet, we have no definite commitment to funding for community Clubs, Special Interest Groups, mailshots, awareness and training.

Problems of awareness and training are still widespread in the UK scene. The awareness problem manifests itself, in part, in a perception of expert systems as a risky and esoteric technology. Too many institutions only maintain a watching brief. But watching briefs can lead to problems when, for example, management decides to sample the technology. The person maintaining the watching brief is expected to produce a compelling technology demonstrator. Often, there are simply insufficient resources in house to produce a convincing demonstration. The demonstrator fails, the technology is seen to be immature, the management remains unconvinced, the company maintains its watching brief.

At the other extreme, there are those who argue that building expert system is now all routine. They deprive the technology of its success. The achievements disappear under the moving tide of IT advances. In promoting awareness and interest in a technology, it is always important to enumerate the successes and spin offs.

Problems of awareness crop up in another way. The Alvey programme, although beneficial in many ways, fostered a rather introverted community. The consequence of this was that those inside the community were made very aware of the technology, and soon came to think that everyone else must be too. In fact, the technology has not really succeeded in getting outside of the Financial Times top 100 listed companies. There is still a substantial job of awareness to tackle.

There has also been a belief that awareness leads to technical competence. The fact is that acquiring an adequate knowledge engineering competence takes a lot of effort, and a current problem is the provision of trained people. The most comprehensive training is obtained by those who have taken a number of the IKBS M.Sc. conversion courses available and who have computing or a relevant cognitive science background. However many personnel are recruited from general IT conversion courses. The problem here is that these courses are often only marginally relevant to the technologies which comprise IKBS. Training problems are compounded by the fact that we lack an agreed idea of a syllabus or curriculum that might form a minimal requirement for a qualification in knowledge engineering. There are still relatively few Honours undergraduate courses in A.I., and even fewer graduates going on to complete PhDs. Those personnel in post, who are expected to acquire competence in the new technologies face the problem of finding the time and expert help to support their retraining. There is an urgent need to attend to the training requirements in IKBS and A.I.

Moving on to consider the application of expert systems technology, we noted the proliferation of small to medium sized applications. Most of these are rule-based. Most occupy modest but effective niches in a variety of areas from manufacturing to finance, power utilities to medicine. There are, however, a number of areas in which it has proved more difficult to deploy IKBSs. These include real time applications, very large databases applications, and domains where the reasoning is non-standard, where conditions are constantly changing, or where the knowledge base itself needs to be constantly updated. These application areas present technological and methodological problems that are forcing us to apply a richer range of techniques, and develop more powerful methods of IKBS specification and formulation. To some extent, they are setting the agenda for current expert system research.

If we look at current developments in methods and tools for knowledge engineering we find one particularly active area - knowledge acquisition. It is difficult to establish methods and methodologies for conducting acquisition through the life-cycle of KBS construction. The most thorough framework is provided by KADS - Knowledge Acquisition and Domain Structuring (Breuker 1987, Weilinga and Schreiber 1989 this volume). KADS embodies seven principles for the elicitation of knowledge and construction of a system. We will not detail them all here, but one is of particular relevance given the discussion earlier about knowledge levels. It recommends that the analysis should be model-driven as early as possible. This requires that one should bring to bear a model of how the knowledge is structured early on in the process, and use it to interpret subsequent data. This will involve appeal to what we have called inference level knowledge earlier in this paper. It may also include appeal to models of the domain, or devices in the domain (Chandrasekaran, 1988).

An important theme in this and other current approaches to knowledge acquisition is that the enterprise should be viewed under the metaphor of model building, rather than the mining of information. In this regard, we have moved from a transfer view of acquisition to a model view. This recognises that even within first generation expert system construction, a knowledge engineer is engaged in a subtle process. Knowledge engineering is not simply a matter of transferring knowledge from an expert into a knowledge base. The final product is a model of various aspects of an expert's knowledge.

A rather less disciplined methodology and yet one that is almost always associated with expert systems is *rapid prototyping*. The idea is that it is easier for experts to criticise a working system, than it is to specify the system in the first place. Initially, a prototype is built, without much regard to its weaknesses, and the expert makes suggestions about its performance. These suggestions are incorporated into the system by programmers, and at the next session there should be fewer errors. This cycle continues until the expert is satisfied with the behaviour of the system.

There is some debate as to whether rapid prototyping constitutes a methodology or a knowledge acquisition technique. In fact, a growing area of research is concerned to *evaluate* the various claims made about knowledge acquisition methods and techniques (Burton et al 1987, 1988, Shadbolt & Burton 1989).

Tools construction is an important area of current work. Again let us take knowledge acquisition as illustrative. Currently, there four main types of acquisition tool available or under development (Boose 1989 provides a comprehensive review).

Firstly, those systems which are implementations of standard knowledge elicitation techniques, such as repertory grids and concept sorting. Secondly there are those systems which use machine learning techniques to induce rules from sets of worked examples and observed data. In addition to these categories, there are also systems which use knowledge about the structure of a particular domain in order to drive the elicitation. However, these are large-scale systems dedicated to specific projects, and are not generally available. Finally, there are a number of large-scale, generic knowledge acquisition environments under construction. These typically provide a number of automated KE techniques, knowledge base editors, automated transcript analysis and various other support software for the knowledge engineer. These systems are currently at the research stage, and as yet are not generally available. Although they are not yet available they indicate the shape and form of the next generation of knowledge engineering tools.

4 The future

In the last section we mentioned some of the difficult technical problems that must be solved if expert systems are to progress. One of these is expanding the types of reasoning available to systems. Those advancing the cause of second generation systems regard the expansion of reasoning capabilities as crucial. Many applied problems require recourse to non-standard methods of reasoning such as *default* and *abductive* reasoning. Many of these non-standard reasoning methods are *non-monotonic* (Ginsberg 1987 for a review).

Work on non-monotonic reasoning is becoming an important issue in expert systems. Many current expert systems make an implicit assumption of *monotonicity*, facts true at the beginning of a reasoning session are assumed to remain true throughout. If facts subsequently become false then usually the system has to restart inference from the beginning. There is no way of determining what information generated by the system is still valid. Many deductions may have been made on the basis of a fact that is no longer true. There are systems and shells that offer mechanisms to help manage this problem, so-called *truth maintenance* or *belief revision* systems. They tend to incur high computational overheads and complicate problem solving. The provision of more elegant solutions to these problems remains an important area of work (Smith & Kelleher 1988).

Hand in hand with reasoning is the representational component of any expert system. At ES84 Professor Aaron Sloman (1984) made an appeal for the provision of more varied classes of representation to support the kind of complex modelling and reasoning that our second generation systems will require.

He argues that if we look at the notations, formalisms and representational systems used by a wide range of professions, from mathematics to music, programming to cartography, we find a huge variety of types. These have arisen to fulfill requirements imposed by the nature of the domain and the purposes for which they were to be used. Some of the forces that have shaped the development of these *representations* are perceptual and cognitive, and involve problems of parsing and interpreting certain sorts of structure. Some of the forces of development have had to do with the processes that the formalisms are involved in; calculation, planning, searching, and the detailed control of action. Sloman recommended that

> we need to explore the uses of different sorts of formalism for different purposes. We need to understand how an intelligent system can choose between different formalisms, and how it can, on occasions, create new formalisms when doing so would give new insight or heuristic power of some kind

This recommendation still stands. One very radical approach is to be found in the technology of connectionism of neural nets (McClelland & Rummelhart 1985, Rummelhart & McClelland 1985). And is is not just the problem of representation that this technology is being applied to. It is also being used to tackle problems in learning and perception, reasoning and information retrieval.

Neural nets consist of a set of processing units. In neural nets all processing is carried out by the units - there is no control or executive program. Units are connected together and each connection has a weight or strength. Unit's receive input and as a function of these inputs compute an output. The system is inherently parallel because many units can carry out their computations at the same time. In a network learning usually occurs due to the modification of the weights of existing connections.

Within such nets the representations are patterns of activation over units. Experience is recorded as changes in the weights of a net. Patterns of activation come and go, what remains are traces when they have passed. A trace is bound to be distributed over many different connections, and each connection is implicated in many different associations. The traces of different experiences are therefore superimposed in the same set of weights. In neural nets one regards the retrieval of a representation as a partial reinstatement of a network state, using a cue which might only be a fragment of the original input.

The whole connectionist research enterprise is generating a great deal of excitement. It is claimed to offer real solutions to very hard problems in representation, perception and learning. Connectionist models seem able to take in large amounts of data and self-organise so as to learn underlying regularities and patterns in the data. They are then able to recognise similar patterns in new data and reinstate previous patterns as appropriate. This makes them an exciting prospect for a whole range of expert system applications.

But such networks are not without their problems (Pinker & Prince 1988). It is often difficult to come up with the right set of inputs. One has to decide how to set the weights on the connections and how they should subsequently modify themselves. Because of the distributed nature of the knowledge in such nets it is virtually impossible to obtain explanations of the net's behaviour. Nevertheless there is no doubt that they will begin to make their presence felt within our subject.

We now move on to a different force which will play a part in how expert systems evolve - hardware developments. We take for granted the remarkable performance now being delivered on lost-cost machines. However, this power is changing both what we can do and how we do it.

The developments in hardware will ameliorate many problems associated, for example, with real time, on-line applications. The emergence of super-PCs will break the constraints imposed by restrictive operating systems and limited memory. A similar breakthrough is occurring in the workstation range - increasing power is offered at falling prices. This will allow quite modest organisations to run networks of powerful machines. These PC-workstations will provide 16 million instructions per second with 32 megabytes of main memory as standard. This sort of power will also support ever more extensive programming environments.

An important secondary feature of this new generation of PC-workstations will be the *routine* provision of large, high resolution displays. Such hardware devices will provide the medium for much more sophisticated Human Computer Interaction (HCI). The incorporation of graphical, video and audio displays into expert system interfaces will provide solutions to some of the problems of information presentation.

A rather different consequence of this raw power may be a move back to *brute force* methods. A major impetus behind early A.I. was the need to produce elegant axiomatisations of problems so as to circumvent hardware limitations. It was simply not possible to imagine building a natural language translation system that operated by recourse to table lookup. When planning or game-playing the search of even quite small problem spaces required intelligent heuristics to guide the search. Increasing computer power allows computationally intensive approaches to become an option. But limitless power and brute force methods can also reduce the motivation to look for principles.

In contrast to technical issues let us consider directions in applications. Whatever field is chosen it is clear that one major development will be the increasing importance of *embedded* expert systems. Such systems will sit within much larger conventional software. Such embedding requires methods and standards if a coherent and consistent design philosophy is to arise that extends from traditional pieces of software through to KBS and expert system programs.

We have already mentioned that more expert systems will be built to tackle a wider range of generic problem solving areas than has been attempted hitherto. As this happens the technology will need to be aware of developments in other branches of A.I. One example is the work that has been steadily progressing in the planning and scheduling sub-fields of A.I. Indeed there are signs that substantial collaboration is occurring in this area between expert system and A.I. researchers.

One set of questions a natural historian would ask, concerns the social life of the species being examined. Whilst it would be a little premature to enquire after the social habits of expert systems, their creators certainly have social ends and ambitions. What of them? There has been within the BCS SGES a long-standing and proper concern for the social consequences and implications of expert systems. There have been a number of conference articles, and now a journal, dedicated to these matters. We will not rework the arguments here save to remind ourselves that expert systems do not exist in a moral vacuum. They raise important issues of responsibility and accountability, matters of judgement and conscience.

5 Concluding remarks

This paper has tried to discern the origins, current state and future directions of the expert system. It has been largely preoccupied with the UK scene, but that reflects the principle interest of our parent professional body the BCS. It has also sought to draw on a somewhat strained and lighthearted analogy with the natural historian's account of the progress of natural forms.

6 References

Boose, J.H. (1989) A survey of knowledge acquisition techniques and tools. Knowledge Acquisition, 1.

Bramer, M.A. (1956). Expert systems in Britain: progress and prospects. In Research and development in Expert systems, ed. M.A. Bramer, III, Cambridge University Press.

Breuker, J. (Ed.) (1987) Model-Drive Knowledge Acquisition Interpretation Models. Deliverable task A1, Esprit Project 1098; University of Amsterdam.

Buchanan, B.G. & Feigenbaum, E.A. (1978). DENDRAL and Meta-DENDRAL: their applications dimension. Artificial Intelligence, 11.

Bundy, A. (1987). How to improve the reliability of expert systems. In Research and development in Expert systems, ed. D.S. Morales, IV, Cambridge University Press

Burton, A. M., Shadbolt, N. R., Hedgecock, A. P. & Rugg, G. (1987) A Formal Evaluation of Knowledge Elicitation Techniques for Expert Systems: Domain 1. In Moralee, D. S. (Ed.) Research and Development in Expert Systems IV. Cambridge University Pres

Burton, A. M., Shadbolt, N. R., Rugg, G. & Hedgecock, A. P. (1988) Knowledge Elicitation Techniques in Classification Domains. ECAI-88: Proceedings of the 8th European Conference on Artificial Intelligence.

Chandrasekaran, B. (1988) Generic tasks as building blocks for knowledge-based systems: the diagnosis and routine design examples. Knowledge Engineering Review.

Dawkins, R. (1986). The blind watchmaker. Longman

Engelman, C. (1971). The legacy of MATHLAB 68. Proceedings: Second Symposium on Symbolic and Algebraic Manipulation, pp. 29-41

Gaschnig, J. (1982). PROSPECTOR: an expert system for mineral exploration. In Introductory Readings in Expert Systems, ed. D. Michie: Gordon & Beach

Ginsberg, M.L. (1987) Readings in non-monotonic reasoning. Morgan Kaufman

Hammond, P. (1982). APES: a detailed description. Report 82/10, Computing Dept., Imperial College, University of London, 150 Queen's Gate, London

Martin, W.A. & Fateman, R.J. (1971). The MACSYMA system. Proceedings: Second Symposium on Symbolic and Algebraic Manipulation, pp. 59-75

McClelland, J. L & Rummelhart, D. E (1985) Parallel Distributed Processing: Explorations in the Microstructure of Cognition Vol 2 Psychological and Biological Models. Cambridge, Mass: MIT Press.

McDermott, J. (1980). RI: an expert in the computer systems domain. Proceedings: AAAI-80

van Melle, W. (1979). A domain-independent production-rule system for consultation programs. Proceedings: IJAI-79, pp. 923-925

van Melle, W, Shortliffe, E.H. & Buchanan, B.G. (1984). EMYCIN: a knowledge engineers tool for constructing rule-based expert systems. In Rule-based Expert Systems, ed. B. Buchanan & E. Shortliffe, pp. 302-328, New York: Addison-Wesley

Pinker, S. & Prince, A. (1988) On language and connectionism: analysis of a parallel distributed processing model of language acquisition. Cognition, 28.

Reboh, R. (1987). Knowledge engineering techniques and tools in the PROSPECTOR environment. SRI Technical Note 243, SRI, 333 Ravenswood Avenue, Menlo Park, California

Rummelhart, D. E & McClelland, J. L (1985) Parallel Distributed Processing: Explorations in the Microstructure of Cognition Vol 1 Foundations. Cambridge, Mass: MIT Press.

Shadbolt, N.R. & Burton A.M (1989) Empirical Studies in Knowledge Elicitation. ACM-SIGART Special Issue on Knowledge Acquisition

Shortliffe, E.H. (1976). Computer-based medical consultations: MYCIN. New York: Elsevier

Shortliffe, E.H., Axline, S.G., Buchanan, B.G., Merigan, T.C. & Cohen, S.N. (1973). An artificial intelligence program to advise physicians regarding antimicrobial therapy. Computers and biomedical Research, Vol.6, pp. 544-560

Sloman, A. (1984) Why we need many knowledge representation formalisms. In Research and development in Expert systems, ed. M.A. Bramer, Cambridge University Press

Smith, B. & Kelleher, G. (1988) Reason maintenance systems and their applications. Ellis Horwood.

Szolovits, P. & Parker, S.G. (1978). Categorical and probabilistic reasoning in medical diagnosis. Artificial Intelligence, 11, pp. 115-144

Steels, L. (1986). Second generation expert systems. In Research and development in Expert systems, ed. M.A. Bramer, III, Cambridge University Press

Wielinga, B.J. & J. Breuker, (1986). Models of expertise. Proceedings: ECAI'86, pp. 306-318

Weiss, S.M., Kulikowski, C.A., Amarel, S. & Safir, A. (1978). A model-based method for computer-aided medical decision-making. Artificial Intelligence, 11, pp. 145-172