Keynote Paper. AIENG95 Conf, Udine, Italy, July 1995

Artificial Intelligence in Engineering: Past, Present and Future

Professor George Rzevski
Centre for the Design of Intelligent Systems, Design Discipline
The Open University, Milton Keynes MK7 6AA, UK

Abstract

This is a review paper which sets the scene by defining some fundamental concept, such as intelligence and intelligent systems, and then discusses current trends in applications of artificial intelligence in engineering. The author reviews five key paradigms of artificial intelligence in engineering: knowledge-based systems, neural networks, genetic algorithms, fuzzy logic and intelligent agents.

Introduction

Among a plethora of concepts and methods of artificial intelligence (AI) in engineering, in my opinion, the key paradigms are:

1. *Knowledge-based systems*
2. *Neural networks*
3. *Genetic algorithms*
4. *Fuzzy logic*
5. *Intelligent agents*

My prediction is that these paradigms will yield important results for many years from now and that *the future will be dominated by the paradigm of intelligent agents*.

I shall begin by providing my personal definitions of some fundamental concepts of artificial intelligence in engineering. This will help the reader to understand the intellectual platform on which I stand while observing and reviewing the world of AI in engineering.

Some Fundamental Concepts

Intelligence

Intelligence is one of those elusive concepts, very much like quality and excellence that appears to be impossible to grasp fully. Nevertheless, one should start somewhere. I focus on the interaction of a system with a changing world in which it operates. For the purposes of my research [1, 2, 3] I define *intelligence* as *the capability of a system to achieve a goal or sustain desired behaviour under conditions of uncertainty.*

I have arrived at this definition by studying the phenomenon of intelligence in biological systems where, in operational terms, one can argue that intelligence helps them to cope with unpredictable changes in the environment. A comprehensive discussion of intelligence in biological systems can be found in [4]. I am painfully aware of philosophical problems associated with intelligence [5] but nevertheless believe that considerable progress can be made in solving engineering problems under conditions of uncertainty without reference to intentionality.

Intelligent systems have to cope with the following sources of uncertainty.

- The occurrence of unexpected events, such as an unpredictable change in the world in which the system operates (eg, the occurrence of a fault, a change in order priority, or a late modification of design specification).
- Incomplete, inconsistent or unreliable information available to the system for the purpose of deciding what to do next. This uncertainty may be caused by the speed at which unexpected events occur (eg, a brief appearance of an intruder's face within the viewing range of a security camera) or by inadequate data presented to the system (eg, fuzzy user requirements).

I distinguish between two classes of intelligent systems:

- *Intelligent decision support systems* ie, AI programs that advise and support engineers via human-computer interfaces, as exemplified by intelligent computer-aided design systems and intelligent fault diagnosing systems, and
- *Intelligent machine systems* ie, machines and interconnected machines with embedded AI which are capable of operating autonomously, as exemplified by intelligent machine tools and intelligent robots.

It is helpful to contrast intelligent systems ie, systems that can make decisions under uncertainty, with systems that are programmed to make only deterministic decisions.

Deterministic Behaviour
Deterministic behaviour is exhibited by artefacts capable of achieving specified goals or sustaining desired behaviour only under *predictable conditions*. Data processing systems, conventional robots, production lines and computer controlled machine tools are examples of such systems. Major strengths of this type of behaviour are precision and repeatability. The major weakness is its inability to cope with unexpected events. For many years *automation* was synonymous with the economy of scale and mass production. It is now increasingly difficult and costly to construct and maintain stable operating environments such as rigid production lines, required to implement automation. Therefore the demand for machines with deterministic behaviour is steadily declining. Under volatile market conditions an important asset is flexibility which automated systems do not have.

Proto-Intelligent Behaviour
Proto-Intelligent behaviour is exhibited by artefacts and biological systems (such as plants) capable of achieving specified goals or sustaining desired behaviour under *well defined variable conditions*. Many artefacts, from thermostats to auto-pilots, and biological systems such as plants, can cope with such conditions. I have used the term Proto-Intelligence to describe *self-regulation*, the most elementary behaviour that may appear externally as intelligent. It denotes the capability of a system to achieve and sustain the desired behaviour when working in an environment which changes in time in a limited way. The characteristics that change, the range of measurable changes, and the way in which the system should respond to any particular change are known in advance. Only the timing and magnitudes of changes are not known.
In general, for the purposes of self-regulations a system may monitor one or several measurable physical characteristics, called variables, such as position, distance from a given object, direction of movement, speed, acceleration, pressure, liquid level, thickness and composition. Whatever the variable or the set of variables, the mechanism of self-regulation is always the same: the feedback loop. Demand for proto-intelligent machines is steadily increasing. Sensors are now being built into a variety of machines which were previously constructed or programmed to behave in a strictly predictable fashion.

Intelligent Behaviour
Intelligent behaviour is exhibited by artefacts and biological systems capable of achieving specified goals or sustaining desired behaviour under conditions of uncertainty *even in poorly structured environments* ie, environments in which variable characteristics are not measurable, where several characteristics change simultaneously and in unexpected ways, and where it is not possible to decide in advance how the system should respond to every combination of events (eg, a situation in which a mobile robot must distinguish between a person and a piece of furniture in a workshop in which it operates, or a novel failure pattern that a diagnostic system is expected to deal with). Intelligent behaviour is characterised by a number of externally recognisable features such as those discussed below [6].

Adaptability - the system is capable of achieving specified goals or sustaining desired behaviour in an environment characterised by unpredictable external changes.

Self-Maintenance - the system is capable of maintaining its own state of operational readiness, by means of self-diagnosis, preventive self-maintenance and self-repair by re-configuring, under conditions of unpredictable internal changes (faults).

Communication - the system is capable of exchanging information with other systems with a view to exercising control over, reporting to, receiving instructions from, or engaging in competition or collaboration with other systems.

Autonomy - the system is capable of acting independently (to a certain degree) from other systems, including human operators.

Learning - the system is capable of being trained to carry out certain tasks.

Self-Improvement - the system is capable of improving its own future performance based on past performance combined with learning from other agents or human operators.

Anticipation - the system is capable of predicting changes in its environment which may affect its operation.

Goal-Seeking - the system is capable of formulating and modifying tactical sub-goals with a view to achieving specified strategic goals. In highly volatile environments or environments about which system designers have inadequate knowledge there is a need for intelligent systems capable of learning about the new environment by means of interacting with it and then formulating achievable tactical sub-goals within constraints imposed by the overall strategy.

Creativity - the system is capable of generating new useful concepts, principles or theories, and conjecturing and testing methods and methodologies. Creative systems can operate successfully by interacting with humans. A fascinating possibility is to let a creative system interact with a section of the real world autonomously with a view to formulating new concepts, principles and theories about it.

Reproduction - the system is capable of creating replicas of itself. Whilst reproduction of hardware systems is not of immediate concern, the need for software reproduction is overwhelming. It is quite feasible to develop software genes loaded with instructions how and under which conditions to replicate themselves and form similar or identical programs.

At present there is no demand for comprehensive intelligent behaviour, that is, for behaviour which would encompass all the features described above. As our ability to design intelligent systems improves the requirements will no doubt change. The implication is that we should find ways of adding features of intelligent behaviour incrementally.

Intelligent Systems in Engineering

Intelligent systems will play an increasingly important role in engineering due to fundamental changes in economic conditions and rapid development of AI technology. Occasional doubts expressed about the long term future of AI are misplaced. Consider the following argument.

Markets for engineering products and services are now global rather than national or regional. As new countries and new engineering organisations join in, keen to gain an increased market share, they create a large surplus of supply over demand. As a result the nature of demand changes. To survive and prosper in the new economic climate vendors make effort to reduce concept-to-market lead times and design products and services to match as closely as possible the requirements and expectations of customers in identifiable market segments. This requires flexible decision support systems and flexible machines capable of achieving goals under conditions of perpetual change [7, 8]. The trend is clearly from automation and proto-intelligent systems towards flexibility provided by intelligent machines and intelligent engineering decision support. It is particularly strong in the design of machines and vehicles where digital technology coupled with fuzzy logic is steadily replacing mechanical engineering components, as exemplified by active suspensions, intelligent engine management systems, intelligent navigation systems and the like.

This trend is helped by the continuous decrease in the price/performance ratio of digital technology at a time when costs of manpower and materials are increasing.

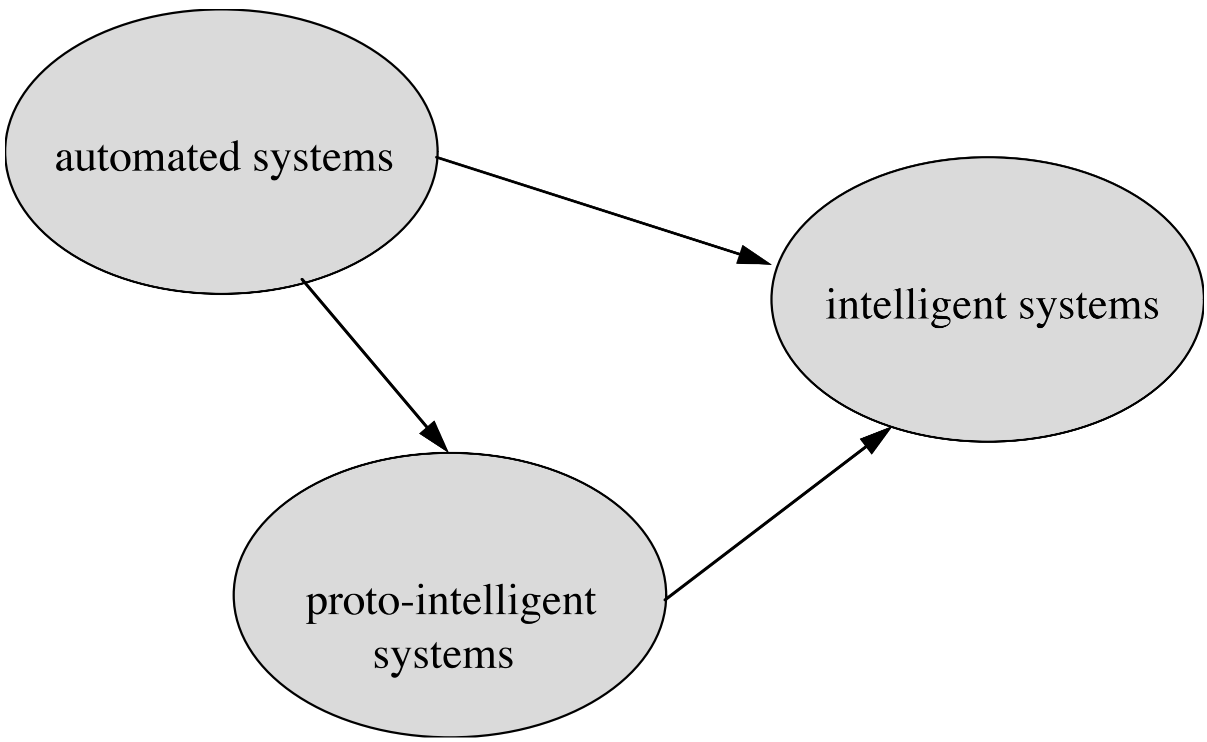

Fig. 1 Trends

Architectures of Intelligent Systems

Let us for a moment consider intelligent machine systems. To exhibit autonomous intelligent behaviour a machine must be capable of performing three fundamental functions named in [3] as:

- *Perception*,
- *Cognition* and
- *Execution.*

The two major roles of Perception are (a) to collect data about the world in which the system operates (this world includes the system itself and its environment) and (b) to process collected data (so called data fusion) with a view to assembling reliable information on the basis of which decisions can be made on future system behaviour. Perception is usually associated with building and updating models of the world in which the system operates. However intelligent behaviour may be achieved without an internal world model.

Cognition includes considering system goals and the current state of the world (possibly also likely future states) and, based on this information, planning future system actions.

Execution is about initiating and controlling a particular behaviour.

There are many possible ways of organising these functions to achieve autonomous intelligent behaviour. From an engineering perspective the key consideration is that the resulting system must be cost-effective to implement, sell, operate and service. User requirements are typically: acceptable cost and size, modifiability, reliability, safety, appearance, social and legal acceptability, ease of use and ease of servicing.

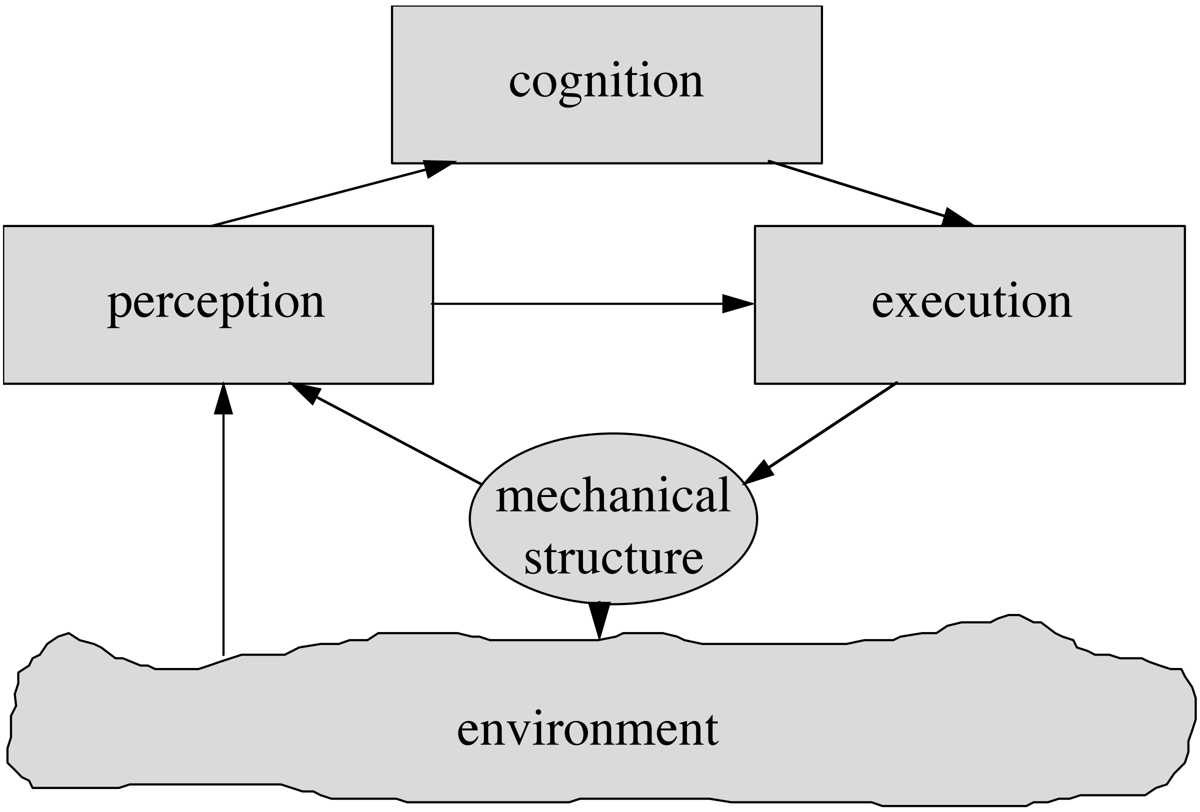

Fig. 2 Major functions of an intelligent machine system

Conceptually the most simple is a centralised architecture with perception, cognition and execution functions implemented as separate but interconnected subsystems. However, from the engineering point of view a centralised architecture is not feasible. For example, the complexity of a centralised perception subsystem for an intelligent factory would be difficult to imagine. Even for a single autonomous vehicle working in a factory such architecture is not really practical. Centralised architectures are on the way out even in decision support systems, which are less complex since they do not have to process sensory data.

The usual approach to reducing complexity is to adopt a multi-level hierarchical architecture with perception, cognition and execution functions distributed at various levels of the hierarchy. Many systems of this kind are currently under development. However, hierarchies have a major disadvantage and that is their rigidity. Evidence is mounting that hierarchies are not suitable for worlds characterised by frequent changes.

A number of very successful prototypes of intelligent machines have been constructed using the so called layered architecture [9]. Brooks' original work in the field of robotics has opened new opportunities for the development of intelligent machine systems with direct links from perception to execution.

A very attractive alternative is to assemble a system from a number of autonomous intelligent agents connected in a network and capable of collectively generating desirable system behaviour. Intelligent agents may be designed to operate in *collectives*, organisations similar to colonies of ants, in which every constituent element obeys precisely defined rules of collaboration, or in *societies*, organisations similar to human societies, in which artificial intelligent agents negotiate, collaborate or compete among themselves.

I believe this architecture holds the key to the future of AI in engineering. Beginners should carefully read the introductory papers published in the Communications of the ACM Special Issue on Intelligent Agents [10].

Knowledge-Based Systems

Research into Artificial Intelligence can be traced back to the Second World War. Its origins are rooted in the work of Alan Turing in the UK and in Cybernetics, the science of control and communications in humans and machines, in the USA.

At the beginning, research effort went into the development of machines capable of solving difficult mental problems of any kind, including chess and draughts. Although some interesting results were achieved, in particular by Newell and Simon, who published the description of the General Problem Solver [11], no real progress was made in developing practical machines.

Slow progress was experienced until researchers realised that in order to solve a practical problem the problem solver must have, in addition to an ability to reason, a substantial amount of knowledge specific to the problem domain. General problem solving skills, although important, are not sufficient. Credit for this change in direction from general problem solvers to specialised intelligent Knowledge-Based Systems (KBSs), is often given to Feigenbaum who introduced the notion that the performance of KBSs critically depends on the amount of domain-dependent knowledge stored in the system [12]. In the UK pioneering work was carried out by Michie [13] and Aleksander [14].

The first knowledge-based systems appeared in the 70s, and because they contained high-level, domain-specific knowledge elicited from human experts, they were called Expert Systems (ESs).

The first significant *engineering* application was published in the 80s [15]. It was a knowledge-based system called R1 (later renamed XCON) used to configure DEC VAX computer systems.

Newell and Simon [16] developed the most widely used knowledge representation formalism ie, productions or rules, and a large number of KBSs used in engineering are designed to manipulate knowledge represented in this manner. However, in many engineering applications it is often convenient to represent knowledge in a variety of ways, including rules, frames, semantic nets, English sentences and mathematical expressions. Sloman in [17] strongly argues that we need a wide variety of ways of representing knowledge, primarily because the formalism used for knowledge representation determines the procedures that can be used to operate on it.

EMYCIN (empty MYCIN) published in 1979 by Van Melle [18] represented a new departure. It was the first knowledge system 'shell', that is, a knowledge system which is without any knowledge. It is empty. Shells were later heavily used in engineering applications enabling engineers without extensive computing skills to develop modest but nevertheless useful applications.

In order to develop a knowledge-based system it is necessary to acquire relevant knowledge and translate it into formalism such that it could be stored in a knowledge base. This activity ie, knowledge engineering, is generally considered to be the bottleneck for any KBS development project.

My research shows that it is possible to distinguish between the following categories of knowledge and skills required to operate a successful manufacturing business:

1. Creative skills, that is, skills in finding new ways of attracting customers, organising people, developing products, planning manufacturing processes, creating new markets and selling either existing or new products or services,
2. General business skills, which could be described simply as an ability to effectively manage resources and processes with a view to making money,
3. Expertise, that is, the capability to solve difficult problems in a very effective manner, often relying on experience, empirically derived rules and perceptive observations,
4. Technical skills, which include skills in performing effectively tasks that are not too demanding.

Current knowledge-based systems in engineering are aimed at supporting primarily experts and engineers carrying out technical tasks. I am not aware of a working engineering KBS which incorporates creative skills although the development of systems capable of invention has been reported [19].

An important part of knowledge analysis is to identify generic tasks that engineering decision makers carry out at all levels of the organisation. A partial list of these tasks is given below:

> *interpreting* data, text, images and voice messages; *selecting* methods, materials, ideas; *diagnosing* faults; *analysing* requirements; *assessing* proposals; *modelling* and *simulating*; *controlling* plants, projects, organisational units; *scheduling* resources; *planning* activities or processes; *specifying* systems, products; *configuring* systems; *performing conceptual design*, *embodiment design* and *detailed design*; *estimating* costs; *negotiating*; *training* colleagues.

In many manufacturing organisations knowledge is not properly classified and documented. Records are often incomplete, inconsistent, even contradictory and out of date. It is therefore not surprising that many production activities are often carried out without much reference to documentation. Knowledge bases could be used for maintaining and refining manufacturing knowledge, formalising it and making it available to decision makers throughout the organisation. Since decisions are highly dependent on the quality of knowledge available to the decision maker, knowledge-based systems may:

- Improve quality and increase speed (and, in certain cases, reduce costs) of decision making and thus improve product or service quality, shorten lead times or reduce costs,
- Make new decision processes feasible and thus enable the introduction of new products or services, including services whose purpose is to increase customer loyalty or reduce the bargaining power of suppliers or those devised to create barriers to the entry of new competitors,
- Eliminate decision processes and information handling activities which are not essential and thus save time and money,
- Help to create an organisational culture conducive to innovation, learning and group decision making,
- Create the know-how necessary for an effective transition to a knowledge economy.

One of the main advantages of knowledge-based systems is that they can utilise empirical knowledge which is normally not available in books. It is however widely recognised that the elicitation of empirical knowledge from experts and technical personnel is a very difficult task. Empirical knowledge tends to be implicit, a constituent part of a person's skills and to make things worst, experts usually disagree with each other and provide contradictory advice. The personality of the expert may have a considerable influence on the success of knowledge elicitation. Many consider protocol analysis to be an effective way of eliciting knowledge. Subjects are given microphones and asked to describe what they are doing as they are doing it. If resources are available the additional use of video cameras and court-room shorthand experts is recommended. Wright describes in [20] a comprehensive approach to protocol analysis used during knowledge elicitation from manufacturing technicians and craftsmen. It provides an insight into ways in which operators use visual and aural information to control production.

A diametrically different view is advocated by Wielinga as described, for example, in [21]. His team considers knowledge acquisition to be a modelling activity. Knowledge engineers collect empirical and theoretical knowledge which may be useful for the problem solving and then, based on this knowledge, construct a model of the reasoning process that will generate desired system behaviour.

A considerable help can be obtained from effective knowledge engineering tools, as described for example by Rajan, Motta and Eisenstadt [22].

Large KBSs may be inefficient. Considerable research effort is therefore directed to distribute knowledge among different knowledge systems that co-operate in some manner. Early research into distributed knowledge systems for manufacturing carried out at the Open University is described briefly in [23]. An invited contribution to the fourth AIENG Conference by Findler provides a comprehensive review of distributed knowledge-based systems in manufacturing [24]. Nevertheless, the work on large engineering knowledge bases continues. Perhaps the most interesting research in this area is described in [25].

Mark Lee’s The Knowledge-Based Factory [26] is a paper that should be read by all those interested in intelligent manufacturing. It provides a lucid and comprehensive assessment of the kind of knowledge that is required for manufacturing activities. It considers the advantages and limitations of model-based and qualitative reasoning and suggests how much computational power can be reduced by strategically employing human creativity and decision making skills.

The separation of domain-specific knowledge from domain-independent reasoning is the key to success of KBSs in engineering problem solving. Whilst in conventional systems the designer must decide in advance on the search strategy and must build all search steps into the application program using a procedural language, in the case of a KBS, the knowledge engineer enters into the knowledge base the description of the problem domain (problem space) and heuristics, which guide the search through the problem space, but leaves the inference engine to decide on the exact search steps. This accounts for the relatively rapid development of KBSs and their wide appeal to engineers. The separation of the domain-specific knowledge, which is likely to change in time, from domain-independent reasoning mechanism, offers another important advantage: the ease of modification. Also, and most importantly, KBS architecture ensures that data and operations on data are kept together and thus the meaning of data, which derives from the context of their usage, is preserved.

Whilst at present a majority of engineering applications rely on heuristics, it is very likely that the emphasis will change. I expect much greater use of deep engineering knowledge, which is well formalised, reusable and accessible. There are still many important unresolved issues. For example, have we found the most effective way of representing deep engineering knowledge? What are the ways of structuring knowledge bases to facilitate search and update? How can we combine the power of deep, formalised knowledge with engineering heuristics?

Neural Networks

It is quite natural that an early attempt to create intelligent artefacts was made by emulating the behaviour of brain cells. The first model of a biological neuron was remarkably close to the mark but it failed to find applications primarily because at that time artificial intelligence researchers apparently could not perceive that the power of a neural network derives primarily from the neuron *links,* rather than from the neuron itself. Consequently the approach was severely criticised and, unfortunately, research was more or less abandoned. The change in neural network fortunes came only in the late 80s through seminal work of Rumelhart and McClelland, Kohonen, Hopfield, Aleksander and others. Now we have over fifty different types of neural networks most of them applied in one form or other to a variety of engineering problems. Pham [27] surveyed the field very thoroughly in his invited paper presented to the ninth AIENG Conference and there is no reason for me to repeat the exercise here. Instead, I shall consider some broader issues related to the use of neural networks in engineering problem solving.

Judging by the total number of published papers, neural networks are, in general, considerably more popular as a research topic than knowledge-based systems. In engineering, however, the situation is quite different - neural networks appear to lag behind in terms of practical applications. This may well change in the near future. The key features of neural networks that are very significant for engineering applications are their capabilities to:

- Learn from examples,
- Store information in a distributed fashion, and
- Recognise partially specified patterns.

Perhaps the most interesting feature is that neural networks solve problems by *pattern recognition.* That is very close to how engineers and factory floor technicians work. Wright and Bourne describe this process very perceptively in their excellent book on manufacturing intelligence [28].

Neural networks are likely to be used widely in data fusion, data analysis and classification, vision and learning.

Genetic Algorithms

Genetic algorithms are a computational equivalent of evolution, of the survival of the fittest [29]. An excellent review paper by Goldberg presented at the sixth AIENG Conference [30] stresses the value of this method as a model of conceptual design.

Perhaps the most interesting feature of genetic algorithms is their ability to expand the search space, to diverge, as well as converge. For this reason they are quite effective as search algorithms, particularly for solving optimisation problems with a large number of local minima. Applications in engineering are increasing, predominantly for optimisation. I expect that they will be widely used in a much more interesting way - to generate feasible alternative solutions and select the fittest in the early stages of conceptual design.

Fuzzy Logic

If intelligence is about handling uncertainty, fuzzy logic must be a very appropriate formalism for describing and solving AI problems. The pioneering work was done very early by Zadeh [31, 32] and applied to control problems by Mamdani [33]. An informative review of applications of fuzzy logic to sensing and control was presented by Foulloy and Galichet at the eighth AIENG Conference in 1993 [34].

Fuzzy logic is at present the most widely used method for controlling intelligent machines. It is both simple and effective and applied not only in industrial situations but also for simplifying human-machine interfaces of appliances such as vacuum cleaners, washing machines and video recorders. My prediction is that fuzzy control may soon replace conventional control systems in the majority of practical applications.

Intelligent Agents

There are several interesting hypotheses about human intelligence which may be of use to us when we consider how to design intelligent machines.

The first one is proposed by Newal and Simon [17]. It was estimated that an expert holds in his/her long-term memory approximately 50,000 chunks of information relevant to their domain of expertise, probably in the form of cue-action pairs ie, productions. Short-term memory, where processing of knowledge occurs, has a very small capacity and represents a processing bottleneck. The hypothesis is that the brain solves problems by creating its symbolic representation ie, a problem space, and by conducting a search for a solution in this space. The search is guided by heuristics contained in productions stored in the long-term memory. Experts have access to a large quantity of domain-specific productions and are thus capable of solving problems much faster than non-experts. The above paradigm could be construed as somewhat restrictive because it is based on the assumption that intelligent behaviour is problem-solving oriented: that we always know what we want to do.

An alternative view has recently emerged which postulates that intelligence is the capacity of a system to interact with its environment without clearly defined goals, to learn from this interaction and, in an incremental fashion, to both articulate and achieve its goals. The system may attempt to impose controls on its environment with a view to changing it to suit its goals, or to adapt itself to the environment if it comes to the conclusion that the environment will not change.

Both hypotheses appear to me valid, each applicable to a different practical situation. Expert systems are, of course, based on the first hypothesis. Can we develop intelligent systems to behave according to the second hypothesis? A promising approach is to design the cognition function of intelligent systems as a society of intelligent agents.

The idea that centralised, hierarchical control could be replaced by a group of loosely connected agents (computer programs that are capable of communicating with each other, reasoning about received messages and collectively learning from experience) was inspired by Minsky's seminal work The Society of Mind [35].

The idea is currently being explored in many research centres around the globe. Let me describe briefly a plausible specification for such an arrangement.

- A team of agents is given responsibility for planning and controlling the behaviour of a particular system (such as a machine tool or a vacuum cleaner).
- An agent within this team is given responsibility for initiating and controlling a particular machine behaviour (eg, processing work-pieces, avoiding collision, vacuuming, navigating, or avoiding a failure during a critical operation). The implication is that such an agent would have to have access to both perception and execution functions.
- Another agent is given responsibility for scheduling the machine activities with a view to maximising effectiveness of the higher system of which the given system is a component (say, a factory, a household, or a vehicle fleet).
- Yet another agent is given responsibility for recording, keeping accounts and reporting on all system activities.
- All decisions are made by negotiation among Agent Stakeholders (those agents whose work may be affected by consequences of a particular decision).
- A protocol is established regulating negotiations and specifying non-negotiable categories such as safety.

For example, the operation of a machine-tool may be managed (planned and controlled) by five autonomous intelligent agents [3]. One agent is charged with the goal of achieving the optimal speed of cutting various work-pieces and another with the goal of maintaining the machine in the best possible working order. Under conditions of normal operation, the first agent will monitor speed of cutting and maintain it at the appropriate level whilst the second agent will be monitoring tool wear. When tool wear reaches a critical limit the second agent may decide that unless the cutting speed is reduced the tool will break in the middle of the current operation and will send a request for slow-down. Since the first agent has a preference for continuing at the same speed until the end of the current operation, there will be a conflict. The conflict is resolved by negotiation in accordance with relevant protocols (over a period of time agents may be allowed to modify these protocols with the aim of improving the overall system effectiveness). The third agent, quite independently from the first two, schedules the work load and is given the task to minimise idle time for the machine under its responsibility without reducing overall effectiveness of the factory. This agent negotiates the schedule with agents responsible for other machines in the factory. The fourth agents records all machine activities and commitments, and alerts other agents if the need arises. The fifth agent monitors the immediate environment of the machine and is responsible for avoiding collision with human operators, robots or vehicles transporting work-pieces. In such a scheme a factory is a society of intelligent agents negotiating with each other how best to achieve specified goals. Each intelligent machine, in turn, is controlled by a team of intelligent agents rather then a centralised control system.

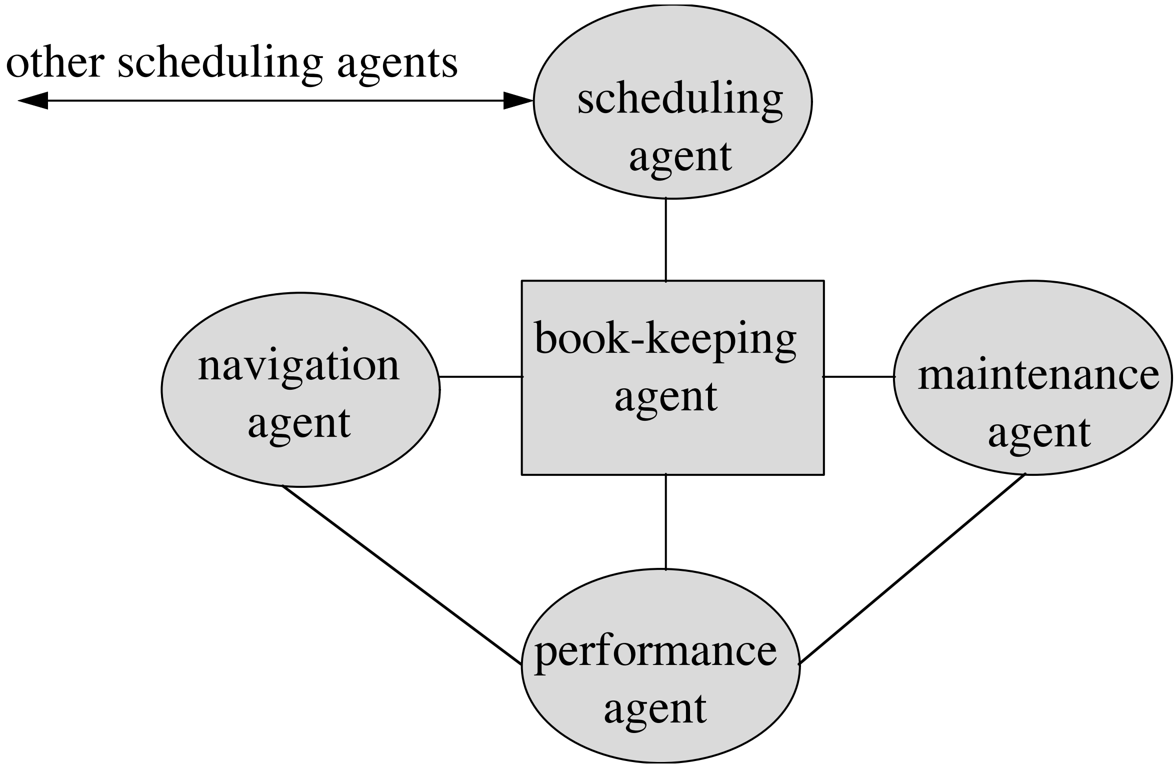

Fig. 3 A Society of Agents

The novelty of this approach is in replacing hierarchical architectures with network configurations in which nodes are capable of negotiating how to achieve specified goals without any centralised control. Negotiation protocols impose constraints on the freedom of agent actions.

We have the required technologies to design prototypes of societies of intelligent agents; these are knowledge-based systems, neural networks, genetic algorithms and fuzzy logic. One way to start is to imitate a colony of ants in which each individual works to a limited set of rules and the whole colony, as a result of application of these rules by individuals, exhibits intelligent behaviour - it copes with a limited uncertainty. Then we can attempt to design a society in which each agent has some limited intelligence as a result of which the society exhibits more advanced features of intelligence. It is important to remember that intelligence is the emergent property of groups of agents engaged in interaction. There exists here a parallel with multidisciplinary teams engaged in concurrent engineering. They tend to be able to handle unpredictable events more effectively than individuals.

References

1 Rzevski, G, Machine Intelligence: The Key Capability of Mechatronic Products. Proceedings of I Mech E Conference on Mechatronics, Dundee, 1992.
2 Rzevski, G, Intelligent Systems: Issues and Trends. Proceedings of Int Conf on Intelligent Manufacturing, Wuhan, China, 1995.
3 Rzevski, G, Mechatronics: Designing Intelligent Machines. Butterworth-Heinemann in association with The Open University, 1995.
4 Stonier, T, Beyond Information: The Natural History of Intelligence. Springer-Verlag, 1992.
5 Aleksander, I, Designing Intelligent Systems: An Introduction. Kogan Page, 1984.

6 Rzevski, G, On Behaviour and Architectures of Autonomous Intelligent Agents: An Engineering Perspective. First International Round-Table on Abstract Intelligent Agents, ENEA, Rome, 25-27 January 1993.
7 Rzevski, G at al, Computer Integrated Manufacturing: Trends. Proceedings of CIM-Europe Annual Conference, Amsterdam, 1993.
8 Rzevski, G, Strategic Importance of Engineering Design. Journal of Design and Manufacturing, Vol 2, No 1, 1992.
9 Maes, P (ed), Designing Autonomous Agents. MIT/Elsevier, 1990.
10 Communications of the ACM, Special Issue on Intelligent Agents, Vol 37, No 7, 1994.
11 Newell, A and Simon, H A, GPS, a program that Simulates Human Thought. Feigenbaum, E A and Feldman, J (eds), Computers and Thought. McGraw-Hill, 1963.
12 Feigenbaum, E A, The Art of Artificial Intelligence: Themes and Case Studies of Knowledge Engineering. Proceedings IJCAI 77, Cambridge, MA, 1977.
13 Michie, D, Expert Systems. Computer Journal, Vol 23, pp 369-376, 1980.
14 Aleksander, I, Burnett, P, Reinventing Man: The Robot Becomes Reality. Kogan Page, 1983.
15 McDermott, J, R1: an Expert in the Computer Systems Domain. Proceedings of AAAI 80, 1980.
16 Newell, A and Simon, H A, Human Problem Solving. Englewood Cliffs, Prentice-Hall, 1972.
17 Sloman, A, Why We Need Many Knowledge Representation Formalisms. Bramer (ed), Research and Development in Expert Systems. Cambridge University Press, 1984, pp 163-183.
18 Van Melle, W, A Domain-Independent Production Rule System for Consultations Programs. Proceedings of IJAA 79, pp 923-925, 1979.
19 Langley, P, Simon, H A, Bradshaw, GL and Zytkov, J M, Scientific Discovery: Computational Explorations of the Creative Process. The MIT Press, Cambridge, MA, 1987.
20 Wright, P K, Linguistic and Somatic Knowledge Engineering for the Automation of Small-Batch Machining. Rzevski, G (ed), Artificial Intelligence in Manufacturing. Springer-Verlag, 1989, pp 203-223.
21 Wielinga, B and Schreiber, G, Future Directions in Knowledge Acquisition. Shadbolt, N (ed), Research and Development in Expert Systems VI. Cambridge University Press, 1989.
22 Rajan, T, Motta, E and Eisenstadt, M, Acquist: A Tool for Knowledge Acquisition. Kelly, B and Rector, A, Research and Development in Expert Systems V. Cambridge University Press, 1988, pp 113-126.
23 Rzevski, G, Distributed Intelligent Knowledge-Based Systems for Cell Scheduling and Control. Proceedings of 19th Int Symp on Allied Technology & Automation, Monte Carlo, 1988.
24 Findler, N, Distributed Knowledge-Based Systems in Manufacturing. Rzevski, G (ed), Artificial Intelligence in Manufacturing. Springer-Verlag, 1989, pp 17-34.
25 Tyugu, E, Large Engineering knowledge Bases. Artificial intelligence in Engineering, Vol 8, No 4, pp 264-270, 1993.
26 Lee, M H, The Knowledge-Based Factory, Artificial intelligence in Engineering, Vol 8, No 2, pp 109-125, 1993.
27 Pham, D T, Neural Networks in Engineering. Rzevski, G, Adey, R A and Russel, D W (eds), Applications of Artificial Intelligence in Engineering IX. Computational Mechanics Publications, 1994.
28 Wright, P K, Bourne, A B, Manufacturing Intelligence. Addison Wesley, 1988.
29 Holland, J H, Adaptation in Natural and Artificial Systems. University of Michigan Press, 1975.
30 Goldberg, D E, Genetic Algorithms as a Computational Theory of Conceptual Design. Rzevski, G and Adey, R A (eds), Applications of Artificial Intelligence in Engineering VI. Computational Mechanics Publications, Elsevier Applied Science, 1991.
31 Zadeh, L, Fuzzy Sets. Information and Control, Vol 8, pp 338-353, 1965.

32 Zadeh, L, A Theory of Approximate Reasoning, Machine Intelligence. Ellis Harwood, 1979.
33 Mamdani, E H, Application of Fuzzy Algorithms for Control of Simple Dynamic Plant. Proc IEE, Vol 121, No 12, pp 1585-1588, 1974.
34 Foulloy, L, Galichet, S, Fuzzy Sensors and Controllers. Rzevski, G, Pastor, J and Adey, R A (eds), Applications of Artificial Intelligence in Engineering VIII. Computational Mechanics Publications, Elsevier Applied Science, 1993.
35 Minsky, M, The Society of Mind. Heinemann, 1985.