Trans.I.Mar.E., Vol. 101, pp. 17–41

Expert systems and marine applications

P. S. Katsoulakos, B.Sc., M.Sc., Ph.D. **and C. P. W. Hornsby,** B.A., M.Sc., D.Phil.
Lloyd's Register, Croydon

SYNOPSIS

This paper reviews the background to the current state of the marine industry. It then describes the history and nature of artificial intelligence (AI), and presents some details of the structure and functioning of an expert system. Expert systems are classified and some significant issues in knowledge representation and reasoning are discussed. The current state of research in AI at Lloyd's Register is described, and a review is presented of current and possible applications of expert systems in marine technology. The implications of advanced computing techniques for ship and offshore operations are assessed.

INTRODUCTION

The past decade has seen a steadily increasing number of applications of artificial intelligence (AI) to every branch of science and engineering. Collaborative efforts by experts and AI specialists have produced systems which diagnose disease, evaluate military threats and even prospect for minerals, at a level of performance equalling that of human beings. The potential power of systems which not only replicate expensive or rare human knowledge, but also are capable of producing cumulative versions of it has stimulated worldwide efforts to develop and apply this technology. In time, expert systems will influence all areas of human activity where knowledge provides the means for solving problems.

During the same period of intense activity in AI research, fundamental changes have taken place in the world shipping industry. These led to a serious decline in shipbuilding activity and in the profitability of ship operation. Marine transport continues to be a low technology industry compared with the electronic, computer, communications and even air and road transport industries. This is one of the reasons for the 'vicious circle' of recession which has afflicted the industry for the past decade (Fig. 1).

However, it is inevitable that general progress in technology will lead to the application of high technology in shipping. Many projects currently in progress, including the 'efficient ship' schemes, have this goal in mind.

Marine transport encompasses shipbuilding and ship operation which, as in many other industries, continue to exist as completely separate functions. Both are highly competitive international businesses. Today, countries with high labour costs are seeking to improve their efficiency by developing and exploiting computer-based technologies (Fig. 2). The flexibility offered by automation in manufacturing makes robotic systems particularly suited to shipbuilding. The industrial robot can be used in a variety of complex manufacturing tasks and is highly developed for welding and plate-cutting operations.

To meet the present shipbuilding economic demands, the new generation of shipbuilding robots need to comply with two requirements. The first is enhanced awareness of the environment, for which new sensory systems are needed. The second is improved in-built intelligence for autonomous decision making. This means that future systems will have to replicate and apply autonomously what amounts to human expertise. To further this aim, 'knowledge engineering' can provide the technology to convert human knowledge into industrial power.

Dr. P. S. Katsoulakos obtained his B.Sc., M.Sc. and Ph.D. in marine engineering at Newcastle University. In 1981 he joined Lloyd's Register as a surveyor, initially involved in offshore platform design and finite element and fracture mechanics analysis. More recently, his work has focused upon applications of artificial intelligence (AI) to the marine field. He is responsible for the initiation and management of major AI-related research projects, including the EEC-funded project on residual fuel oils, the Condition Monitoring project under the U.K. 'Efficient Ship programme, and recent ESPRIT projects.
Dr. C. P. W. Hornsby obtained his B.A. in philosophy, politics and economics at Oriel College, Oxford, followed by an M.Sc. in computer science at University College, London, and a Ph.D. in African politics from St. Antony's College, Oxford. In 1986 he joined the University of Surrey, with responsibility for the development of an expert system to plan the rehabilitation of sewer networks, funded under the Alvey programme. He was also co-author of an expert systems development environment. In 1988, he joined Lloyd's Register with responsibility for artificial intelligence.

In the ship operation sector, the technical objectives are remarkably similar. Moves towards enhanced operational flexibility and computerized ship management will require the use of advanced electronics, computers, instrumentation and communications. 'Expert' or 'knowledge-based' systems provide a means of extending automation by making systems independently capable of diagnostic and predictive decisions, providing for example a coherent picture of the ship environment in uncertain and changing wind and sea states and operational requirements.

Knowledge in any subject is usually of two kinds, public and private. Public knowledge includes the facts and theories which appear in text books and references. Private knowledge consists largely of rules of thumb and generalizations that have come to be called 'heuristics'. Heuristics enable the human expert to make educated guesses, especially where information is incomplete, or where an unusual combination of conditions is involved. The ability to recognize and select promising approaches to problems with uncertain or incomplete data, and to check quickly whether results from complex analytical computational methods are realistic, is a very important part of

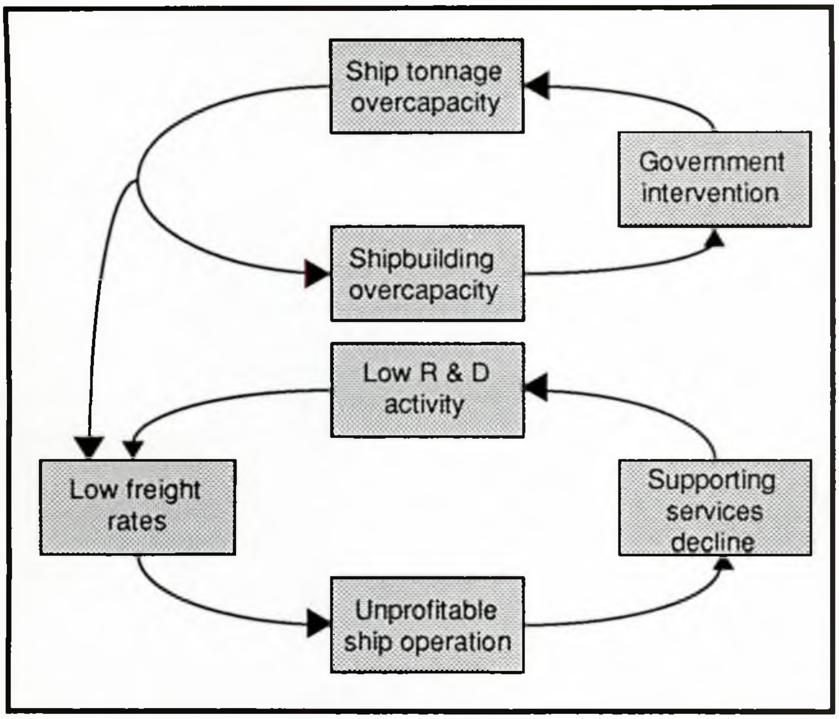

Fig. 1. An overview of the marine transport industry

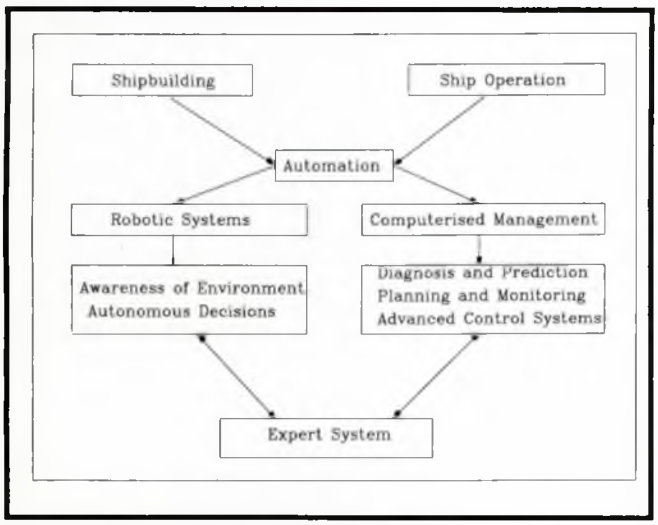

Fig. 2. Future sea transport features

human decision making in engineering. However, this expertise is increasingly becoming eroded by the use of computers to solve problems in a manner which is not fully understood by the decision maker. Transferring currently available knowledge into expert systems, to allow it to be used efficiently, is thus becoming an important issue.

The place of AI in ship transport and classification in the future is summarized in Fig. 3. However, a great deal of effort is still required both in resolving important difficulties in AI technology and in adapting it for practical use in the marine field. Within Lloyd's Register, experience has been gained in AI in general, and expert systems in particular, through the development of applications in several marine technology areas. Many of these developments have been carried out by the Performance Technology Department under three research programmes. The first is the Fuels Project, funded by the Energy Programme of the Commission of the European Communities (EEC) and the U.K. Marine Technology Directorate (MTD). The project has produced a fuel characterization expert system, with the active participation of 30 members from oil companies and engine manufacturers to research establishments. The fuels project, which is nearly completed, is closely linked to a second project, the Condition Monitoring Project, which is funded by the U.K. Department of Trade and Industry (DTI) and the MTD. A new generation of condition monitoring systems for diesel engines is being developed, incorporating fault diagnostic and predictive maintenance expert systems. The third project is the Danish-led ESPRIT Project 'Shipboard Installation of Knowledge-Based Systems' (KBSSHIP). (ESPRIT is the European Strategic Programme of Research in Information Technology.) KBSSHIP is aimed at the development of an integrated expert system to support the master and officers in optimizing the safe and economic operation of merchant vessels. A new ESPRIT project led by Lloyd's Register has also been recently approved, aimed at developing a tool kit which will assist in the maintenance and verification of existing software, which is expected to have a significant impact on future ship-related computer programs.

In the following text, the main principles of AI and the tools for building expert systems within the Society are described. A review of marine applications is then presented with emphasis on the prototype systems currently under development in the Society's research and development programme.

ARTIFICIAL INTELLIGENCE

Definitions

Most people today have some idea about what AI means, since the range of published applications in medicine, science and engineering is constantly increasing. However, it is difficult to state precisely what the field of AI covers. A picture of the discipline can be provided by the following definitions.

"AI is the science of making computers do the things which, if performed by humans, would be termed intelligent."

"AI is the study of how to make computers do things which, at the moment, people do better."

"AI is the study of computer techniques for solving problems by exploiting efficiently knowledge about a problem domain."

The first definition makes the essential point that AI is concerned with making 'smart computers' by engaging in human-like cognitive processes. The next definition says something about the goals of AI. They are, at least in part, directed at the development of a better understanding of the human thought process. A new field of study called cognitive science has emerged to investigate this area. As 'knowledge engineering' catalyses a global effort to collect, codify, and utilize applicable knowledge, clarification and expansion of the human knowledge process itself will also be achieved.

The last definition brings us closer to engineering reality. It highlights the fact that the key to intelligent problem solving lies in reducing the random search for a solution by the use of knowledge. Most problems can be cast in the form of a search for a path from some initial state to a desired final state which

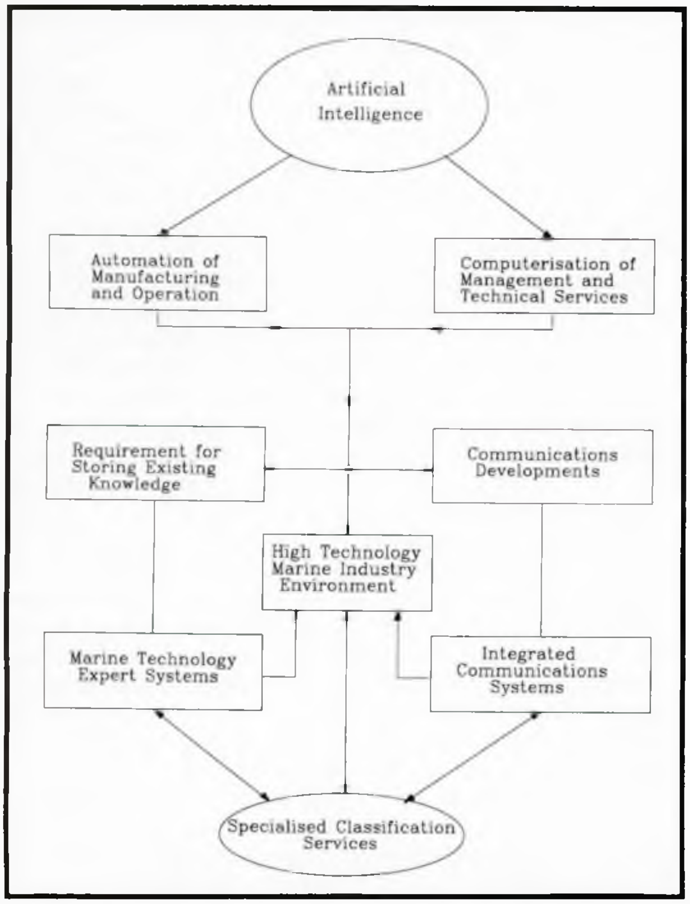

Fig. 3. The future impact of AI in the marine industry

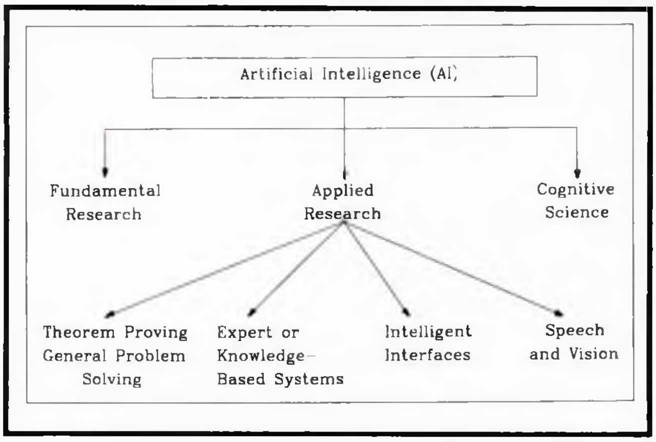

Fig. 4. The principal divisions of AI

is regarded as the goal. The network of possible routes leading from the initial conditions to the goal is viewed as a 'search space'. Both the definition of the search space and the guiding of the search process require the application of specialized knowledge. The central role of knowledge in intelligence explains why the most successful AI programs so far have been expert or knowledge-based systems which operate in specific subjects. In short, expert performance depends critically on expert knowledge and the effective handling of that knowledge.

There are three principal divisions of AI: fundamental research, cognitive science and applied research (Fig. 4). The first is directed at the development of theoretical techniques for progressing towards the AI goals mentioned previously. An example is the development of systems with the ability to learn from experience. Cognitive science is directed towards understanding the way in which the human beings use knowledge. In contrast, applied research is directed at the development of programs with some specific applied purpose, usually industrial, scientific or administrative. It is within this category that expert systems lie, the element of greatest interest to the engineering community.

Historical background

Some 40 years ago a British mathematician who had spent the war developing code-breaking techniques turned his attention to a concept which had fascinated men for ages. Having been actively involved in the development of early electronic computers, Alan Turing began to consider if it was possible for machines to behave intelligently and how this could be established or refuted. At that stage computers were in their infancy. However, Turing demonstrated that a certain kind of very primitive computing machine could compute anything that was computable. Scientists and philosophers began to wonder whether the brain was a 'Turing machine'. From these beginnings the initial AI objective, to provide a means of programming intelligence into machines, was set. In the following years researchers tried to build intelligent computers by imitating models of the brain as neural networks. What was overlooked was the fact that the human brain contains 10 billion neurons, each one an advanced form of analogue device. The proposed structuring of the neuron networks was shown to be an inadequate model of the brain.

The first period of AI research was dominated by a naive belief that reasoning models coupled with powerful computers would produce human-like or superhuman performance. However, by the late 1960s, the severely limited power of general purpose problem solving strategies was realized. Expectations were reduced, and attention focussed instead upon application problems. The lesson learned was that expert knowledge was the key to expert performance; the knowledge representation and inference schemes merely provide mechanisms for its use. It was from this background that expert systems emerged.

The first successful expert system was the mass-spectrogram interpreter DENDRAL[1]. DENDRAL analysed chemical experimental data to infer the possible structures within a known compound. It employed an efficient variant of a simple methodology known as generate-and-test. Partial molecular structures consistent with the data were first generated, then elaborated in all possible ways. By systematically generating all plausible structures DENDRAL found candidates that human experts sometimes overlooked.

The best known expert system was MYCIN (1976), which addressed the problem of diagnosing and treating bacterial infections of the blood[2,3]. Several new features were introduced during its development which have since become hallmarks of an expert system. Its knowledge comprised approximately 400 rules relating possible blood conditions to associated interpretations. A scheme was devised based on 'certainty factors' to allow the system to reach plausible conclusions from uncertain or fragmentary evidence. MYCIN was also able to explain its reasoning processes. The user could interrogate it in various ways by enquiring why it asked a particular question or how it reached a conclusion.

Many other well-known expert systems were produced successfully in the late 1970s and early 1980s, mainly as research tools, but some also proved practically useful. They included PROSPECTOR in geology, which found a previously unknown deposit of the valuable mineral molybdenum, R1 for configuring DEC VAX computer systems, and HEARSAY II for speech understanding. A recent system, EURISKO (1982), which improves and extends its own body of heuristic rules through a complex learning process, made a breakthrough in very large-scale integration by inventing a 3-dimensional AND/OR gate.

Search and reasoning

Many AI systems make extensive use of search techniques. The 'state-space' system is an example of this approach. States are snapshots of the problem at different stages of a solution (Fig. 5). Each state is generated by the application

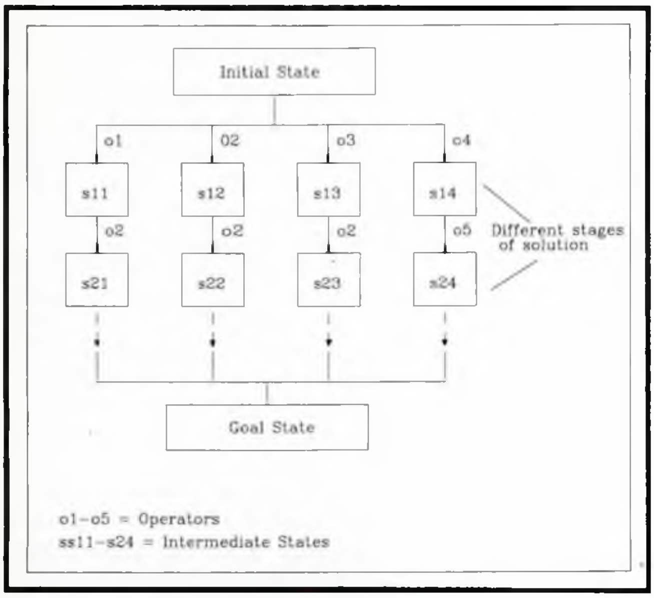

Fig. 5. 'State-space' search

of operators, first to the initial states, and then to the intermediate states. The aim is to find a sequence of operators that can be applied to the initial state in order to reach the goal. There are various search methods, including exploring all possible intermediate steps.

The size of the underlying search space of practical AI problems prohibits a systematic consideration of all the alternatives. For example, if the initial state of a problem was that a ship's speed is reduced, and the goal was to identify the cause, the possible number of combinations of engine, propeller and hull deterioration that would explain this reduction is enormous. The efficiency of the search process is thus crucial to AI applications. A number of techniques are available for efficient searching, optimized for specific categories of problem.

Search is sometimes formulated as the simple 'generate and test' process described earlier. Another technique is that of 'progressive charting refinement' of a subject which allows search to be carried out in a number of stages. In the first stage a global subject 'map' provides overall directions. The details can then be established in a 'local' map.

For many applications it is possible to apply specialized information to guide the search process. Such information can be a mixture of mathematical theory and heuristics. Heuristics can be expressed numerically or by rules. Numerical heuristics are functions estimating the closeness of a search path to a goal. To be useful, evaluation functions must characterize the solution space adequately, which requires a substantial amount of knowledge. Arithmetic functions with weighted coefficients can only partly represent such knowledge. Heuristic rules in symbolic computing are often necessary.

Reasoning is often based on creating assumptions or hypotheses and later revising beliefs in the light of new knowledge received or derived. The revision process can be facilitated if the dependencies amongst the current set of hypotheses are known. These dependencies can then be analysed to establish justifications before an assumption is confirmed or retracted. Dependency-directed reasoning can be extended to the problem-solving process itself, by introducing justifications about goals and constraints. Alternatively the solution process can be guided by fixed searching methods or heuristics.

Current difficulties in AI

Despite the significant achievements of the past decade, AI technology is still at a research phase. Some AI programs can be regarded as experiments which will be partly disregarded as soon as they have been developed and tested. The successful principles they embody do however provide the basis for further experimentation. Key issues which have been established during the relatively short history of AI are those of search and knowledge representation. Given a particular language for expressing knowledge, the task of the solution method is to explore the resulting search space. Many powerful knowledge representation systems, reasoning techniques, search mechanisms and supporting software have been developed.

The main difficulties in AI relate to those fundamental principles which normally provide a solid framework in any established engineering or scientific subject. Such principles include the definition and classification of the application, the theoretical foundations, and the assessment procedures for a solution.

(i) AI can be applied in almost every subject. As a result, the definition and classification of the AI application area is not clear. For example, in medicine, human illness is classified in terms of pathology, osteopathy, neurology and so on. Such a classification scheme for expert systems has not been generally accepted. A challenge to AI is to specify which knowledge representation schemes are suitable for different categories of problems, and which solution methods perform well with each representation.

(ii) The main theoretical basis of AI is logic. However, heuristics rather than theories are often the cornerstones of expert systems. There is a lively debate between those who prefer formal rigour, and those who believe practical experimentation is the best route forward. When fundamental principles for each class of application are established, certain methodological advantages will emerge.

(iii) It is also important to be able to predict the main characteristics of a solution, having adopted certain techniques to produce it. Such performance prediction in AI is not possible at present. For example, it is not always possible to estimate which knowledge representation and inference mechanism will give the fastest execution time in a specific application. Validation of an expert system also presents difficulties, as traditional methods cannot be used. The problems for which expert systems are being developed do not have a clear specification. Consequently, there is often no standard or reference against which performance can be judged[4].

EXPERT SYSTEMS

General description

An expert system is a computer program designed to embody organized knowledge of a specific domain of human expertise, and to simulate the performance of an expert in that field. The knowledge must thus be organized in such a form that the program can offer intelligent advice or make intelligent decisions. Expert systems have several distinguishing features. These include:

(i) Symbolic knowledge representation and inference.
(ii) Heuristic search and reasoning facilities.
(iii) Self-knowledge, employed to reason and rationalize the system's behaviour, and to provide explanations or justifications for conclusions.

The basic components of an expert system

The basic components of an expert system are shown in Fig. 6. The first component is an input/output module. Its main function is to provide a mechanism for communications between the user and the expert system. Problem descriptions are supplied by the user, and advice and explanations of decisions reached by the expert system are output. For expert systems operating in real time the input/output interface provides a filtering device for information received from sensors or other information sources. The output in the form of advice or warning is either sent to computer screens or can be linked to alarm or control systems.

The core of an expert system consists of a knowledge base, a data base (or working memory) and an inference mechanism. The knowledge base contains all the available knowledge about a particular subject. The database contains all the facts known or deduced about this particular problem. Inference strategies are then applied in order to derive a solution to this problem. It is one of the basic characteristics of an expert system that this separation takes place, allowing alterations and improvements to the knowledge without changes to the inference mechanism, and allowing common inference techniques to be applied to a variety of problems. It also provides transparency to the user and the domain expert. Both can see, if required, what the system is doing and why it is doing it.

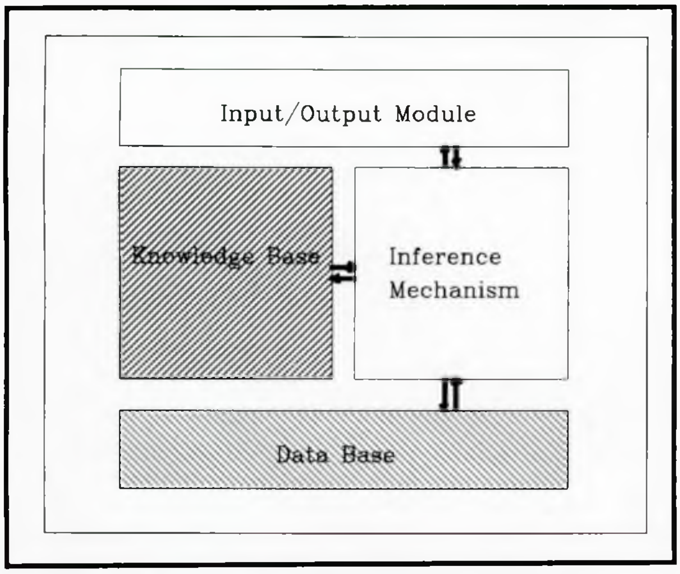

Fig. 6. Basic components of an expert system

The knowledge base

The knowledge base contains all the information required to solve the problems for which the system is defined. The main facilities for representing knowledge are rules, object definitions, relationships and procedures. Rules are the most familiar element of an expert system. They are generally in an IF...THEN... form, which states either that if this situation occurs, this should be the reaction, or that in order to prove that this is the answer, it is sufficient to prove that these conditions hold. Object definitions (or descriptions) are used to identify and differentiate different knowledge building blocks. Increasing numbers of expert systems now use them to define the basic components of the system, and their possible values. Relations express dependencies and associations between these building blocks, and hence between the facts defined within them. Procedures specify sequences of operations to perform when attempting to solve a problem.

When an expert system is running, rules and procedures generate new facts or new hypotheses in a database (or working memory space), describing the current state of the problem. The activation and scheduling of operations and the initiation of alternative solution routes can be co-ordinated by control knowledge.

The inference mechanism

The inference mechanism is a control scheme for the application of the knowledge contained in the knowledge base. The main inference techniques are known as forward chaining and backward chaining. Real systems may focus upon one or the other, or a combination of both. There are however several other techniques, ranging from hypothetical reasoning to pattern matching and statistical pattern recognition.

Forward chaining involves reasoning from data to conclusions. Forward chaining rules consists of a condition and an action part. If the condition matches the appropriate parameters in a database then the action is executed. With a rule which states IF *X* THEN *Y*, a forward chaining system will ascertain if *X* is true, and then deduce a new fact, *Y*. As this changes the current set of facts, new rules may become applicable.

Backward chaining attempts to find data to prove or reject a hypothesis by regarding it as a goal. The system first identifies the appropriate conditions that would be sufficient to achieve the specified goal. It then proceeds to try to establish these conditions, by examining the database to see if they are known to be true, and if not, by treating them themselves as goals. Thus the overall goal is resolved into a number of sub-goals; each sub-goal is then further partitioned into more sub-goals until a basic premise is reached, or the attempt to prove the goal fails. This topic is examined in more detail in the Knowledge Representation and Inference Techniques section.

The explanation system

An additional element within some expert systems is the explanation module. This justifies the system's advice and decisions to the user. It answers questions about why some conclusions were reached or why a competing hypothesis was rejected. The explanation module collects all the supporting evidence accumulated from the intermediate hypothesis refinement steps. Current research is aimed at producing 'causal models' which can be used to explain system decisions from first principles. Qualitative models, based on statements such as 'if temperature increases within an enclosed container, then pressure increases' are being specifically investigated for this purpose.

Table 1. Categories of expert system applications

Category	Function	Features		General applications		Marine applications
		Problem	System	Dialogue mode	Real time	
1. Classification	Infer system malfunctions. Categorize or clarify problems actions or items.	Selection from several possible answers.	Database of associations between features and categories causal models. Filtering of improbable answers.	Medical. Electronic. Mechanical. Taxonomic.	Electronic. Mechanical.	Fault diagnosis: - machinery - electrical - communications. Classification requirements.
2. General advice	Infer problem descriptions from specific requests. Apply appropriate information.	Large number of rules, constraints, regulations.	Information storage and retrieval. Database management. Interpretation by heuristics.	Administration. Law advice. Accounts. Design assessment. Information communication.	N/A	International conventions. Ship operating costs. Shipping information.
3. Design	Combine components to solve problem.	No limits on possible options.	Forward chaining to construct partial solutions, constraints and heuristics.	Computer configuration. Space planning.	N/A	Ship machinery designs.
4. Planning and scheduling	Generate and evaluate plans or actions to satisfy given constraints.	Multiple solutions. Concept generation and evaluation. Optimization of objective function. Quantitive constraints.	Prototype plans monitoring evaluation and replanning. Continuous reasoning.	Project management, process planning.	Robotics. Communication. Military planning. Resource allocation. Scheduling. Flexible manufacturing.	Voyage planning. Ship and offshore platform design. Maintenance and surveys planning.
5. Monitoring	Comparison of observations with key features of an adopted plan or specification.	Rapid response to deviations. Problem identification.	Hypothesis updating. Probabilistic techniques. Fuzzy logic.	Fiscal management. Project management.	Regulatory control. Nuclear power plants.	Fleet management. Business plan monitoring. Equipment monitoring.
6. Simulation and prediction	Infer likely consequences of given alterations.	No established theory. High number of significant variables. Parameters with variable interactions.	Parametric dynamic models. Generation of a range of answers based on different assumptions.	Weather forecasting. Crop estimations. Economic forecasts. Risk predictions.	Traffic prediction. Military threat prediction. Financial forecasts.	Predictive maintenance. Trading and freight rate predictions.
7. Identification	Infer descriptions from observables.	Multi-discipline. Different levels of abstraction. Different types of information.	Assignment of symbolic meaning to structure, form and properties of physical objects on situations.	N/A	Vision. Speech understanding. Image analysis. Surveillance.	Weather monitoring and surveillance. Navigation, position control.
8. Control	Integration of identification, diagnostic, predictive and monitoring functions within an overall control plan.	Dynamic task scheduling. Error recovery. Automatic supervision.	Integration of techniques from all other categories. Selection of control algorithms and parameters. Dependency networks.	Control system design.	Air traffic control. Business management. Missile control. Engineering control. Communication control.	Ship management. Bridge integrated control. Dynamic positioning.

A classification of expert systems

There are many different types of classification proposed for expert systems, but one view is that most expert system applications fall into the distinct types summarized in Table 1.

(i) Classification expert systems attempt to deduce which of a number of possible categories or classifications a problem or situation falls under. One of the main subgroups of these are diagnosis systems. Diagnostic expert systems explain why abnormal circumstances are present in a system. Problems of this kind are very common; they include diagnosing a patient's illness and localizing a fault in an electronic circuit. Diagnostic expert systems relate observed behavioural irregularities with underlying causes using a number of techniques. A common method is based on acquiring a database with real observations of associations between symptoms and malfunctions. Another technique is based on generating a description of the system behaviour corresponding to all possible candidate faults; then matching the simulated symptoms with real observations. Causal models which describe all the system components and their relationships could provide the basis of advanced diagnostic expert systems in the future.

(ii) General advice expert systems are used for retrieving information about specific problems. First, the problem is interpreted by heuristics. Then the appropriate information is selected from the knowledge base and presented to the user as advice. Although commonly used in accounts, administration and cost estimates, they can be applied to many engineering functions. Design appraisal, based on specific codes or an identified set of first principles, could usefully employ such methods. Advice on what international convention regulations to apply in certain situations could be obtained by interrogating an expert system. Regular information bulletins could also be stored in a form where the user is quickly guided to the appropriate data. To all intents these systems are capable of acting as a library with an efficient librarian, finding the correct data for the user.

A special category of general advice expert systems is that of intelligent front ends. These aid operators of complex computer systems, for example when dealing with a multitude of alarms in large, distributed control systems during a process upset. Potentially, intelligent front ends will become part of most simulation or other complex computer programs.

(iii) Design expert systems synthesize solutions to problems from a set of possible components, according to certain restrictions on possible configurations, and with general goals in mind. Deciding the best layout for a computer system or the allocation of space and determination of resulting aerodynamic properties for a car are obvious examples. Some of the hardest AI problems are design problems.

(iv) Planning and scheduling expert systems generate sequences of time-related actions that satisfy given constraints. A plan is often formulated by connecting the main entities of a problem in various relationships. Heuristics are used to evaluate alternative plans and suggest improvements. Constraints represent usually conflicting criteria, incompletely defined, and almost certain to change with time. Heuristics are necessary to re-order priorities according to the prevailing set of constraints. Applications include robotics, communication, military planning, resource allocation and manufacturing.

(v) Monitoring systems continually test and study certain features of a situation or product. They can also be combined with planning systems, where the monitoring is of plan execution. They compare observations with features that seem critical to a successful outcome, looking for conditions which might invalidate a plan or for potential effects which could violate a prescribed constraint. Possible applications exist in a wide range of sophisticated systems such as those employed in nuclear power stations, air traffic control and fiscal management aspects.

(vi) Simulation and prediction expert systems infer likely future consequences from given situations. Weather and economic forecasts are typical examples. Usually a dynamic model is employed in which future values are expressed as functions of selected parameters. The values of key parameters, appropriate for a given situation, are selected by heuristics. It is often more realistic to produce answers as a series of optional outcomes; these are based on assumptions which can be also chosen by heuristics.

(vii) Identification expert systems infer descriptions from observations. Problems are solved by storing knowledge of structure, form and properties of physical objects describing the state of a system. This category includes vision models, speech understanding, image analysis and surveillance. Vision and to a lesser degree speech understanding are amongst the most difficult areas of AI research.

(viii) Control expert systems can incorporate all the other systems. An expert control system governs the overall system behaviour. Control is based on formulating a plan and monitoring its execution to refine details. Identification, diagnostic and predictive functions are linked to the plan monitoring system. Air traffic control, management and process control are primary examples.

Real-time expert systems

In recent years, attention has increasingly focussed upon the use of expert systems directly linked in with pieces of equipment, serving fault diagnosis and monitoring roles. Two aspects make a system a real-time one; the system function is time-dependent, and the state of the world to which the system is being applied is changing constantly. There are certain problems in designing AI systems for continuous real-time operation. These relate to issues such as detecting change, speed of response requirements, interrupted reasoning and consistency of reasoning. If some events are more critical than others, the reasoning process must be interrupted to deal with the high-priority call. Changing data creates problems of consistency, and research in this area has established a number of reason maintenance methodologies[5], but additional work is required to improve their efficiency. Real-time planning is another area which is of relevance to many marine applications, such as voyage planning and maintenance scheduling. Here the challenge is to plan which goals, possible actions, time scales and resources may change and how to re-plan when such changes take place.

Collaborating expert systems

The principles of distributed AI were established in the early 1980s. There are two approaches to the problems associated with collaborating expert systems; task sharing and result sharing. Result sharing is especially suited for problems where decomposition into sub-problems is difficult. A supervisory system is used to communicate the problem to all sub-systems. Each system which is able to contribute communicates its results back to the common pool. Work on collaborating expert systems, where several small systems co-operate in order to achieve a given goal, or share data and knowledge in order to achieve their own separate goals, has focussed on what are known as blackboard architectures.

Distributed AI systems have the same advantages over ordinary AI systems that distributed systems always have; modularity, speed, reliability and re-usability. There are some intrinsic problems however, relating to stability, resource-sharing, communications bottlenecks and problem-decomposition.

Features of viable expert systems

In the previous sections the main areas for expert system applications have been identified. Some rules of thumb for choosing the 'right' type of application (one with a relatively high probability of success) are:

(i) problems in which specialized knowledge is needed but which do not require common sense;
(ii) problems in which experts exist, and in which knowledge is available and can be articulated;
(iii) problems in which there is rough agreement between experts (economic plans are not good applications);
(iv) problems which are not trivial and have generated general interest, but which are tractable;
(v) problems in which there is no algorithmic solution, and in which knowledge is heuristic and answers may be uncertain;
(vi) problems in which financial gain can be demonstrated to the organization involved, or an improvement in service to the customer achieved.

Assessment of expert systems

The assessment of an expert system can be based on the evaluation of its performance in three main areas.

(i) Completeness and correctness: the degree to which the knowledge in the domain is an accurate representation of the knowledge required to carry out the task, and is applied to produce high-quality consistent results. This depends amongst other things upon the quality of the knowledge representation and inference techniques used, and upon their successful use in modelling problem solving in the domain.
(ii) Reliability: the degree to which the system is fragile (crashing unexpectedly), unpredictable in its operation, inflexible (unable to function outside its limited area of expertise) and discontinuous in its reactions to similar inputs. Some systems handle a limited number of problems very well, but cannot solve problems even slightly outside a narrow system specification.
(iii) Usefulness and flexibility: the degree to which systems actually carry out the tasks desired by operators in the field, and their ability to be incrementally improved and debugged.

Knowledge acquisition

Knowledge acquisition is the process by which problem-solving expertise is obtained from some knowledge source in a form in which it can then be encoded within an expert system. Sources of information include textbooks and papers, manuals and the experts themselves, both theoretical specialists and practising workers in the field. Such knowledge elicitation involves the collection of specialized facts, rules of thumb and reasoning procedures in a narrow field of knowledge. The transfer of this information to a program is generally done by a 'knowledge engineer', who develops an understanding both of the domain and of the design and of the construction of the expert system. Knowledge acquisition has been recognized in the last few years as a major 'bottleneck' in expert systems development. However, tools are now becoming available to help in the automation of this process. An intelligent editing program can partially substitute for the knowledge engineer in simple applications, particularly when the expert is reasonably conversant with programming.

Another route towards overcoming this bottleneck in expert systems development concerns learning. The ability to learn, though presently in its infancy, makes a system capable of expanding and refining its own capabilities. An essential requirement before learning is the system's ability to reason about its own processes. This, in practical systems, is often implemented by tracing back the rules that played a part in a problem solving session. The rules then serve as their own justification. Specific research on computer learning has emphasized general inductive inference methods[6]. Induction is the process of generalizing from particular instances or examples to general rules or principles. It involves a search for common features in examples which can be subsequently used in classification rules or concept descriptions. In cases where the underlying rules are already known, induction is of less interest.

Rule induction from examples

A number of applications for which expert systems are being developed contain large volumes of data organized in databases. The absence of established theory and lack of experts then make these data a major knowledge source. In such cases the expert can be substituted for in the construction of rules by this data and the knowledge engineer by an induction program. Many marine technology problems fall into this category, making computer induction an important technique in this area. Engineering concepts can often be described by a set of positive and negative examples of problem instances. These are used to generate rules that provide decision trees and point to the most significant solution parameters. Induction is a technique heavily investigated and relatively well developed, with commercial tools to assist in it on the market.

The type of induction process used varies with the source of the examples. If the source is an expert, a sequence of examples can be selected to optimize convergence to the desired concept. When the source is the system itself, examples are generated from the knowledge base information, and an expert is then asked to approve them. More commonly, the examples come from data collected 'in the field'. In general, the induction packages available today are aimed at producing 'IF (condition) – THEN (action)' rules without reference to the underlying cause of this regularity. The EXTRAN (ID3), INDUCE, HGA and BEAGLE systems are examples of this category. They may provide adequate performance in the specified domain, but they are unlikely to identify the causal connections underlying the observed regularities.

KNOWLEDGE REPRESENTATION AND INFERENCE TECHNIQUES

Knowledge representation systems render knowledge accessible to computers. There are a variety of systems available, and they can be described in several ways, by their formal

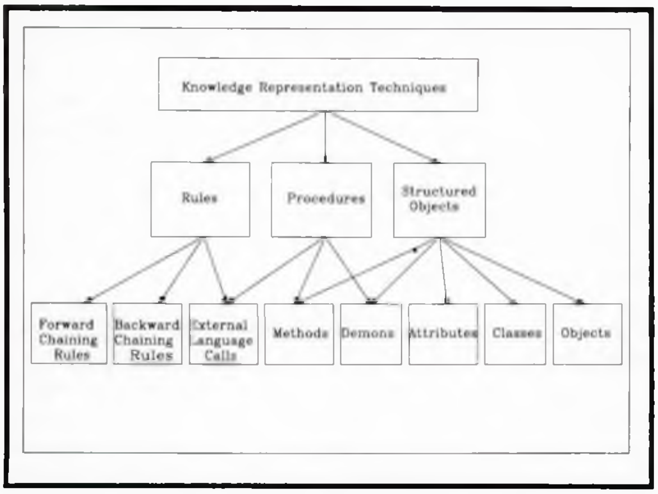

Fig. 7. Knowledge representation systems

characteristics, their user-friendliness and their descriptive power. Fig. 7 shows a categorization of the major knowledge representation techniques discussed below. Similarly, there are several different inference techniques available. We will review in more detail examples of each. In many cases, the two intertwine, since certain inferencing techniques will only be possible with certain knowledge representations.

Declarative and procedural knowledge

Knowledge can be viewed as being of two main types, declarative and procedural. Declarative knowledge states what a solution to a problem would consist of; it provides information which can be applied in several ways. Procedural knowledge embeds within itself the means to solve the problem. It does not state explicitly what a solution would consist of, but rather how to find it.

In pure declarative systems, it is assumed that knowledge representation can be formulated independently from considerations of the way that knowledge will be used. Consequently these systems consist of two components; a static part based on logic theory and an active inference mechanism. The classical declarative representational mechanism in AI research is predicate calculus. It can be used as both a knowledge representation language, and as an inference technique. The original basis of the language Prolog (PROgramming in LOGic) was a formalization and implementation of predicate logic on a computer.

Predicate calculus is a method of manipulating logically basic entities. Knowledge is represented by translating information into formulae of predicate logic. These formulae are added in the knowledge base as axioms. Unlike propositional calculus, where a statement 'Socrates is a man' is reduced to a single constant 'Socrates-is-a-man', in predicate calculus the full structure of the sentence is represented, with a predicate 'man' applied to a constant 'Socrates', producing man (Socrates). Inferences are formulae deduced from the axioms by inference rules. Thus given the simple axiom 'all men are mortal', which we could formalize as:

$$VX\ [\mathrm{man}(X) \text{–>} \mathrm{mortal}(X)]$$

(where 'VX' means for all X, and '–>' means implies), and the fact man(Socrates), predicate calculus will allow us to deduce mortal(Socrates).

The solution approach in predicate calculus is that the goal state is expressed by a formula and is regarded as a theorem which is to be deduced from the axioms by the inference mechanism. Some of the advantages of predicate calculus (and of declarative systems generally) are that the syntax and formal interpretation are well defined, there are clear inference rules and the knowledge base is modular and its application is flexible. However, the rules do not contain information about how to use knowledge, they are often less powerful, and they cannot represent such features as uncertainty or non-monotonicity, where the data being reasoned upon is changing or is inconsistent (as in reasoning over time). Research to improve the restrictive expressive power of predicate calculus is currently being undertaken, concentrating on advanced forms of logic and on integrating predicate calculus with other representations.

In procedural representations, emphasis is placed on the knowledge of how to solve a problem, which is placed within rules or procedures. Most traditional programming languages such as FORTRAN have been procedural, since they contain within them knowledge of how to solve problems, embedded within the program code. AI languages and expert systems combine both procedural and declarative elements, with different systems focusing more on one than the other. Often, primarily declarative systems have the advantage of clarity and rigour, but lack the facilities for complex operations and calculations. Thus, a declarative rule-based language may have the ability to call or have intermixed within it code written in another purely procedural language.

An example of a procedural knowledge representation technique is a method or procedure which is executed when a piece of information is needed. The system does not know what the fact required is, nor what it depends on. It does know how to get it, possibly by an arithmetic computation, possibly by requesting facts from elsewhere and combining them.

An advantage of procedural representation is that control knowledge supports straightforward processing of information for particular purposes. The expressive power of procedural systems is comprehensive; any other scheme can be implemented by a procedural one. Drawbacks include loss of transparency in comparison with declarative systems: knowledge can be attributed to a procedure as a whole and not to particular separate elements. Complex interaction between procedures creates the problem that adding or removing a procedure can lead to unexpected side effects.

Rules

Rules are the basic knowledge representation technique. They may manipulate either simple facts, such as 'if it is raining' then 'you should take an umbrella' or more complex object-based structures, as in if today (X) and raining (X) and person (Y) and going out (Y) then useful-possession (Y, umbrella).

Rules can be procedural or declarative, depending on their complexity and type. Those explicitly containing instructions to assert and retract items in a database, and to print statements and to carry out calculations, are primarily procedural. Others, which simply state that if certain facts are true, then certain other facts are also true (and which leave it to the inference mechanism to react to this) are more declarative.

Rules may be manipulated by a variety of inference mechanisms. Some systems have rules which can be used by both forward and backward chaining systems alike. Others have different syntaxes for each, in which case there is a direct relationship between forward chaining and forward production rules, and backward chaining and backward chaining rules.

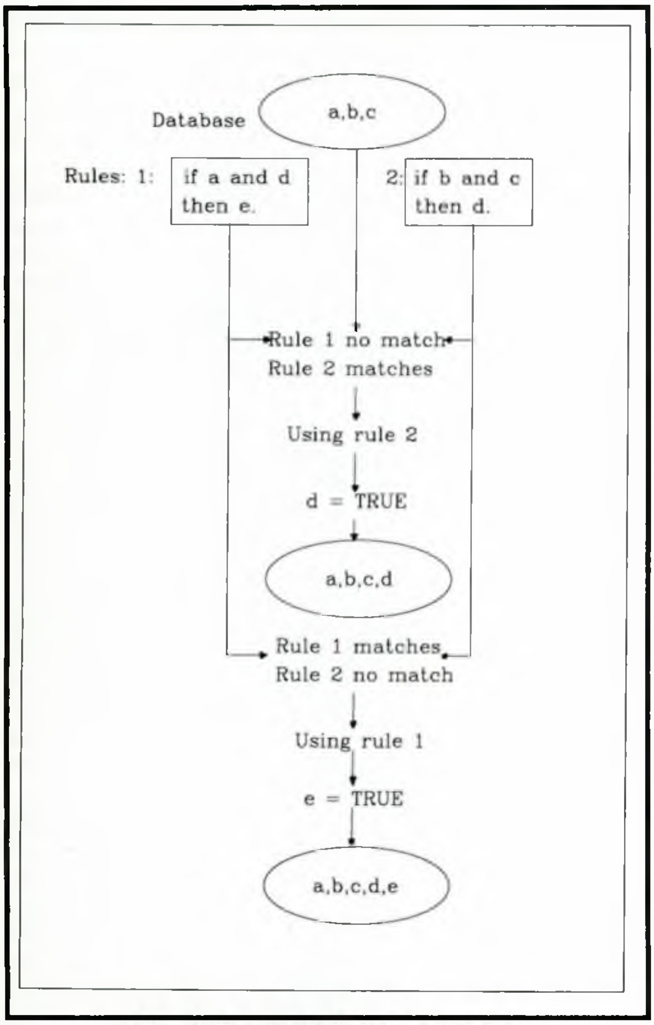

Fig. 8. Forward chaining example

Forward production systems and forward chaining

Forward production systems consist of three main components; a database (working memory), a set of production rules and an interpreter. The database contains the facts about a particular subject. Any changes in the database are monitored by the interpreter. The production rules contain specific knowledge about what additional facts can be deduced if certain conditions are satisfied (i.e. when facts in the database match with the condition of the rules). The action part of the production rules leads the deductions to be asserted into the database, either by the interpreter, or directly by instructions in the conclusion (depending how declarative the system is). New facts added to the database may trigger other production rules, which allow the deduction and assertion of more facts. An example of a forward production system is given in Fig. 8.

Classical production systems have the desirable property of modularity. Production rules can be added or deleted without unexpected side effects, as procedures are initiated only by data and not by other procedures. They can thus be particularly suited for domains described by a large number of independent heuristics. The R1 system for configuring DEC VAX computers is a good example.

The drawback of pure production systems is that they do not contain control knowledge. The systems can be inefficient, as all the condition parts of all production rules are tested before a rule is activated. Inference is also unguided; every iteration depends on all the system variables. In order to alleviate the first problem, structural improvements of the database have been proposed. In order to ameliorate the second problem, control procedures can be applied where the rules are partitioned into groups and only examined according to a pre-specified agenda. Another possibility is to explicitly control the sequence of execution, with rules depositing additional control 'flags' in the database which will trigger certain other rules in subsequent cycles.

Backward chaining systems and backward chaining

In backward chaining systems, also known as goal-directed or goal-driven reasoning systems, inference is based upon the idea that there is a specific goal which is to be proved. The application of rules is not only dependent on the data in the database or control knowledge, but also on the given goal. As in production systems, backward chaining employs a database, a ruleset and an inference mechanism. It also maintains a stack of one or more goals. When the system is given a goal to prove, all the rules which conclude that this goal is proven are examined one by one to find what conditions would, if proved true, allow this deduction. Some of the conditions may be known as facts in the database already. Others may be unknown, in which case the system tries to prove these as new goals, by looking at the database and the rules once more. The result is a backwards move, from goal back to sub-goal, until

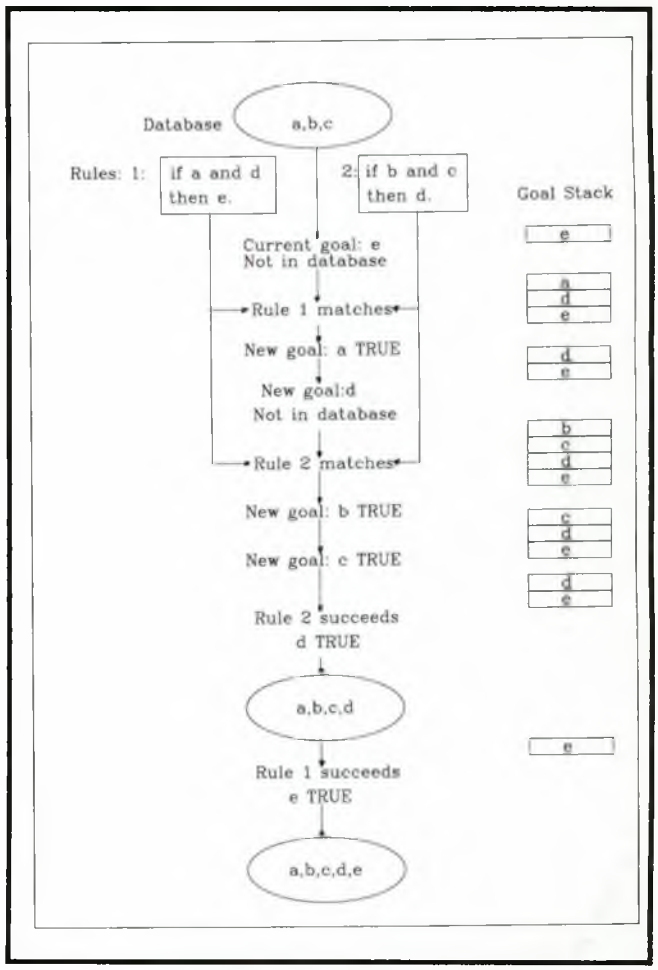

Fig. 9. Backward chaining example

either basic facts are reached which are known to be true, or the goal cannot be proved. An example of backward chaining, showing the deductive processes and the goal stack as it changes, is given in Fig. 9.

Backward chaining can be done in more than one way. All the rules can be examined and explored to a certain level before they are examined to a deeper level. This is known as breadth-first chaining. Alternatively (and more commonly) depth-first chaining is used, in which the first sub-goal (or rule condition) is explored and proved before the rest are considered. Prolog is a backward chaining language which makes use of depth-first search. MYCIN was also a backward chaining system.

Structured object, frame-based and object oriented systems

All these terms describe one very similar notion; that the world can be better described and reasoned about if a system has understanding not only of the connections between text strings such as 'it is raining' and 'you should take an umbrella', but also of the nature, structure and function of the elements or components about which the rules and other parts of the system are reasoning. Object-based or structured-object-based systems are characterized by the addition to the rule-base of a series of object descriptions, schemas or definitions, within which the types of entity to be reasoned about, their possible values, their ranges, and how their values may be obtained if unknown, are all described independently of any rules.

These object definitions may be organized in a hierarchy, in order to represent the real structure of the domain, and to make more efficient the storage of information. Thus a product tanker may be a class of tanker, and a tanker, a class of ship. It is then possible to state facts and general descriptions which must be true of all ships (in the absence of information to the contrary) which will be inherited by all tankers and all product tankers, without the need to state them explicitly. An example covering cars is given in Fig. 10.

Object definitions are now often known as frames, though the notion of a frame originally represented something slightly different. Frames were introduced into the AI research in 1975[7] to represent a possible means of human cognition. Prototypical situations are defined, to which the real one can be matched and the additional information contained in the frame about how to react, or about other facts which can be assumed to be true, can then be applied. Frame-based systems and object-based systems are now treated as essentially the same, as they both define prototypical objects, their characteristics, their common structure and attributes, and mechanisms to obtain data and react to changes.

Within object-based systems, three basic constructs are used to define the entities in the domain; classes, attributes and object instances. At the top level, classes denote the basic problem concepts. They may be organized in a hierarchy. The characteristic properties which can be attached to a class are referred to as attributes. The attribute values are different for each object of a class. Objects then represent particular instances of a class. The object may be thought of as the basic unit of the system. As an example, the frame for a 'ship's engine room' contains slots for the main propulsion system, generators and auxiliary machinery. Thus, in the absence of other information every 'engine room' will be assumed to contain these features or attributes. The value of the attributes is then determined for the individual object, possibly using default information. Objects can be created and destroyed, and can be tested and altered by rules.

Scripts are a special kind of frame and were developed to represent stereotyped sequences of actions and events. Scripts are more appropriate when dealing with dynamic environments. A sequence of scripts gives a description of changes with time (useful in applications such as weather forecasting or predictive maintenance).

An additional element often found within AI systems, and sometimes within conventional applications, is that of message-passing between objects. One method of object organization and manipulation involves attaching procedures to the object definitions, which state what should be done in certain situations, and which permit communication between objects. Objects encapsulate data inside the procedures which understand how to manipulate it. Thus, an object can be manipulated as data, and at the same time it describes the operations supported on that data. Objects are activated by 'messages'. Messages are requests for an object to perform one of its operations. The key feature of messages is that the requested operation is coded by a name describing what the programmer wants to happen, not how it is to happen. Thus a change to one object could lead to a message being sent to another, to alter a certain number of its values. The form of programming within which this technique is used is known as object-oriented programming. It is very similar again to the frame-based and object-based systems described above, with the addition of message-passing facilities between objects. The majority of frame-based systems today permit message-passing also.

The procedures describing the object response to messages are termed methods (described above) and reside in the class description. The different instances of a class utilize the same method. Consequently they respond differently only according to variations in the attribute values. A commonly implemented method is one which obtains a value in situations where it is desired but unknown.

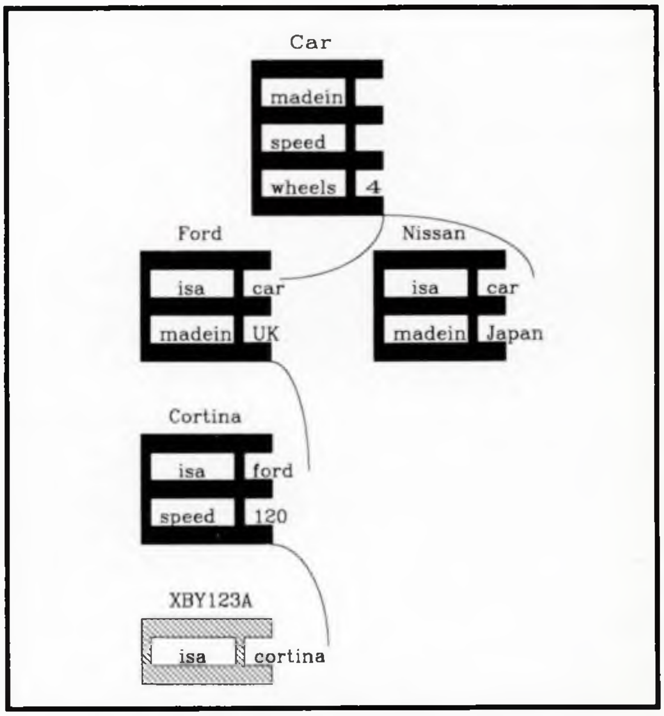

Fig. 10. Classes, objects, attributes and inheritance

Another form of procedure, which describes a reaction to an event taking place in the database, is the demon. Demons are generally attached to object descriptions, but their main goal is to monitor the database looking for examples of the creation, deletion or updating of a value of an object. They then react immediately (without the need to search through or examine rules).

The use of messages, methods, demons and inheritance are all forms of built-in inference, which take place without the intervention of rules. They thus straddle the boundary between knowledge representation and inference.

One last element which should be mentioned concerns hypothetical reasoning. Here, the system can explore different options simultaneously in situations where data is unknown or incomplete. It is then possible to 'split' the expert system's attention, and assume certain things to be true and false simultaneously, exploring the implications of each choice. A hypothesis will then be disregarded if one or more of the assumptions made prove to be wrong or lead to an inconsistency with known facts. This technique is implemented in several of the larger tool kits.

Table 2. Main characteristics of AI languages

Property	LISP	POP11	PROLOG
1. Symbolic representation	Y	Y	Y
2. Flexible representation			
(i) Lists	Y	Y	Y
(ii) Dynamic binding	Y	Y	Y
(iii) Untyped variables	Y	Y	Y
3. Functional language	Y	(Y)	–
4. Logic programming	–	–	Y
5. Self-manipulation	Y	Y	Y
6. Conventional syntax	–	Y	(Y)

BUILDING EXPERT SYSTEMS

Introduction

In building expert systems the principles and tools of knowledge representation and reasoning described above are brought together to solve problems requiring expert knowledge. There are three main categories of tool used to build expert systems: AI languages, shells and development environments.

AI languages

Current AI programming is basically concerned with the simulation of human intelligent behaviour through symbol manipulation. Symbols represent knowledge units which can be used to build complex knowledge structures. Languages which have the ability to reason with symbols rather than numbers are thus essential to practical AI applications.

The generation of intermediate problem states involves

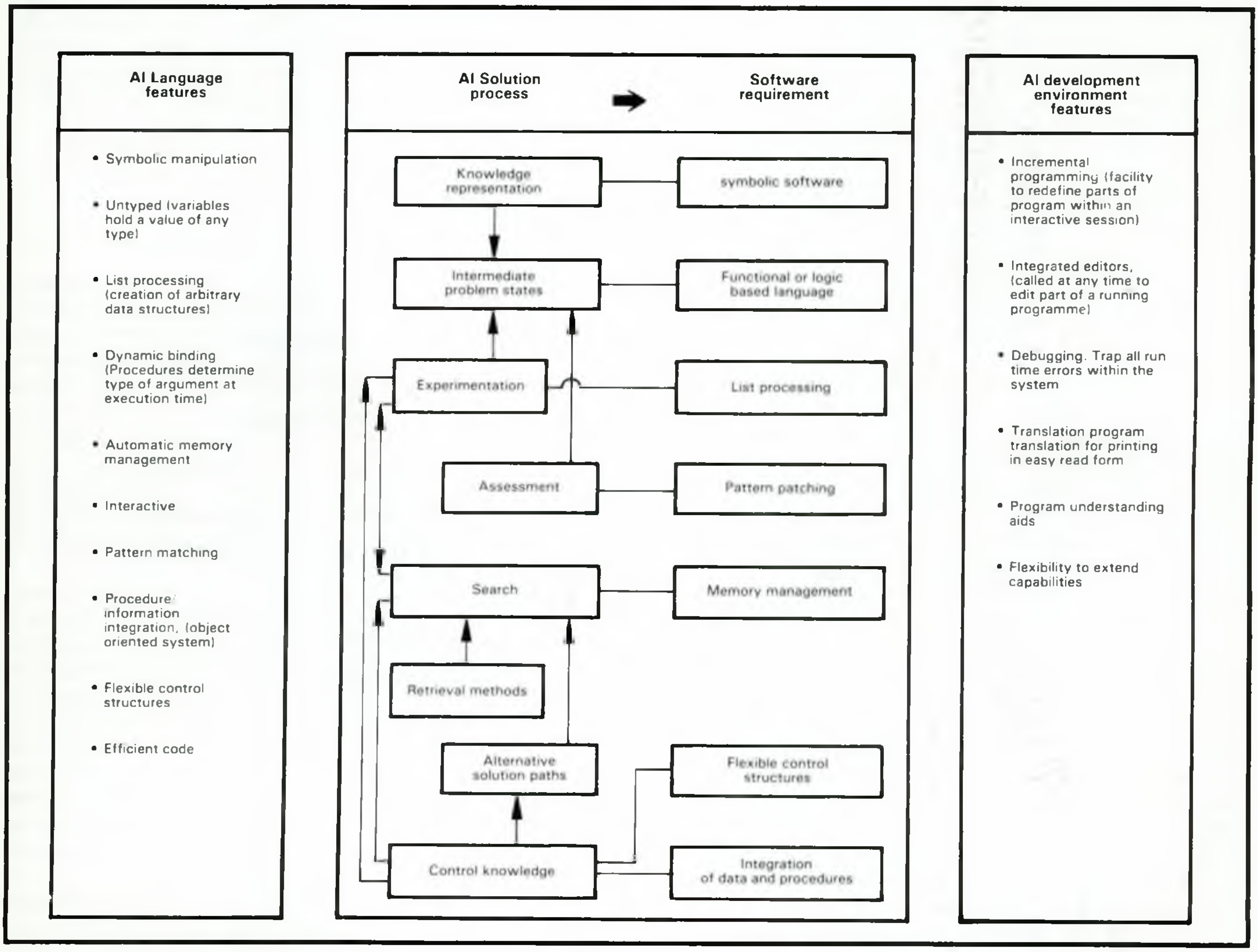

Fig. 11. AI languages

Table 3. Expert system 'manufacturing' tools

AI software	
AI languages	LISP, PROLOG, POP II
AI language environments	ART, KEE, MUSE, POPLOG...
Algorithmic languages	FORTRAN, C, PASCAL...
Programming method	Object oriented
Knowledge representations	Rules, objects, frame systems, logic
Reasoning modules	Forward/backward chaining rules
Control structures	Blackboards, meta-knowledge
Explanation routines	Causal models; dependency networks
AI hardware	
Development machine	Symbolics, Sun, Explorer...
Target machine	PC, VAX, MICRO's...
Parallel processing	Transputers

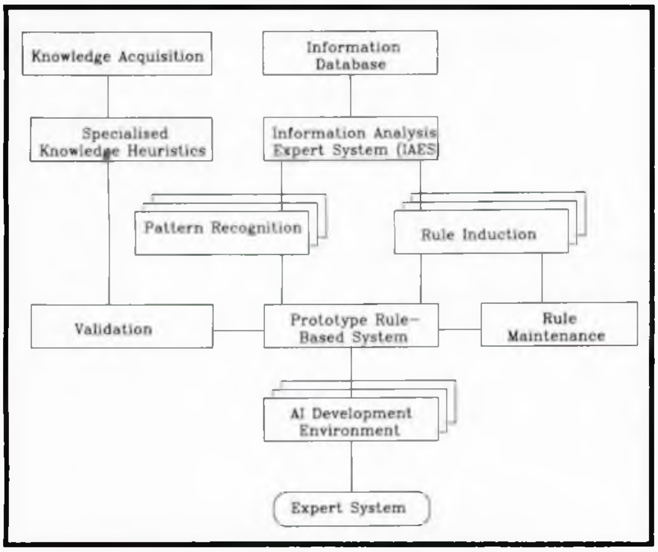

Fig. 12. Lloyd's Register framework for building expert systems

search for applicable knowledge, or for alternative solution paths. Search in some applications can lead to what is termed 'combinatorial explosion' (exponentially increasing number of choices). Although this can be resolved by the introduction of heuristics, automatic memory management is important. This requirement becomes more dominant as the body of heuristics grows. Control knowledge provides the management of both search and assessment and requires flexible control structures in the programming language. The flexibility requirement for a language is further emphasized by the experimental nature of the solution process in AI problems. List processing, untyped variables and dynamic binding are some of the features which have been introduced for this purpose in AI languages. Many of the features which are summarized in Fig. 11 are available in the main AI languages LISP, PROLOG and POP11.

Brief descriptions of some features of LISP, PROLOG and POP11 are given in Table 2. LISP was developed as a functional language based on the Lambda Calculus and provides the highest level in self-manipulation. Everything, including both data and program, is coded in LISP, and building other software layers (for example a reasoning module) on top of LISP is very easy. POP11 is similar to LISP, with fewer functional capabilities. It has a simpler and more conventional syntax, an important goal during its development. PROLOG is a logic programming language. The control mechanism in PROLOG is based on backward chaining. In each language, however, the practical versions available today have veered from their initial pure nature in response to the practical needs of programming.

Shells and environments

One characteristic aspect of AI programming is that the language is often only used as part of a development environment. The task of many AI problems is not one of coding up a solution but of exploring the problem and its possible solutions. To satisfy this requirement, development environments such as KEE, ART, KnowledgeCraft and MUSE have been constructed to support the evolution of a program as the problem understanding develops through experimentation. Some prominent tools currently available for building expert systems are listed in Table 3. Many software facilities such as languages and knowledge representation systems are combined in AI development envir-onments. Such environments are well established today and new 'expert system builders' are being developed, as a consequence of the Alvey and ESPRIT programmes.

Simpler software packages aimed at assisting users to build expert systems for specific applications are termed 'shells'. Often running on personal computers, these specify certain restricted knowledge representation and inference techniques, in order to speed the process and increase the simplicity of applications development. Some, such as Leonardo, also provide built-in uncertainty handling techniques. The cost is reduced flexibility and power.

EXPERT SYSTEMS IN LLOYD'S REGISTER

An overview of the approach

The general approach adopted in the Lloyd's Register research programme for the development of expert systems is outlined in Fig. 12. The knowledge acquisition phase incorporates both knowledge elicitation and induction from examples. The approach is based on a two-stage prototyping arrangement aided by an Information Analysis Expert System (IAES). The underlying concept in this architecture is that of 'Prototype Formulation and Refinement'. Producing a prototype working model is a common approach in engineering developments. In constructing expert systems this becomes almost essential due to the complexity of the problems and the experimental nature of solutions. Human experts also find it easier to criticize a working system than to say what is needed at the initial stages of development. The two-stage approach satisfies the flexibility and experimentation requirements and gives some modularity in the architecture. Furthermore the knowledge base can be incrementally developed and tested.

The first stage of the construction process is concerned with a 'prototype rule-based' system. This first prototype contains rules, typically arranged in a decision tree format, to provide solutions for a simplified model of a problem. Validation and improvement of the rules by experts can be consequently undertaken. A special software tool has been developed in LR for this purpose.

The second stage of development involves the selection and implementation of a knowledge representation system and a control strategy in an AI development environment.

MUSE, an integrated set of tools for developing real-time AI applications, is the main environment currently in use within Lloyd's Register. It provides a range of knowledge representation techniques to access a wide range of applications and to cater for development flexibility. MUSE is also linked to the IAES in a manner that provides the second prototype refinement loop.

Information analysis expert systems: IAES

The majority of marine applications for which expert systems are being developed contain large volumes of data organized in databases. The traditional solution is to manually select the relevant information from the database, specify a model in suitable terms, 'solve' the problem and then compare the 'fit' against solution criteria. This process forms a prototype working model which is subsequently subjected to the test–evaluate–refine loop until the solution criteria are satisfied. This is both inefficient and time-consuming, especially when there is either a large volume of data or a large number of models to be tested.

To overcome this difficulty, recourse is made to an information analysis expert system, IAES[8]. An IAES controls the extraction of information from the database according to the 'initial problem model' defined interactively by the user. The first function of IAES is to provide a software link between a database (with all the available information on the subject) and the expert system development tools.

To assist in defining the initial problem model, IAES prompts the user to define the type of the problem (for example diagnostic, predictive or planning). A list of the main problem concepts is then established interactively and advice is given on possible solution methods. Following the selection of an initial plan, the problem is resolved into sub-problems and the appropriate classes and attributes are defined. A specific IAES database is then produced. This reflects the partitioning details of the problem and the data organization under the appropriate classes and attributes.

IAES hierarchical network model

The next function of IAES is to facilitate the automatic generation of rules or other knowledge structures using a hierarchical network model. Each sub-problem is represented as a node defined by classes and attributes. Rules are generated using induction and statistical techniques, supplemented by heuristics. The following steps are used in specifying the network model.

(1) The problem is initially divided into sub-systems which can be analysed independently.
(2) For each sub-system, different levels are identified in which information contributing to different aspects of the solution sequence is grouped together. At each level the relevant information is distributed into a number of nodes which represent the objects which are being considered within each sub-system.
(3) The first level of analysis generates the initial hypotheses.

Table 4. Results from fuel 'indication' analysis

IV.I Example file

	Attribute									Class
	FSS1	FSS2	FST3	FST5	FSP1	FSP2	FSP3	FSF1	FSF2	
1	1	2	1	4	1	5	7	1	3	101
2	1	3	4	5	2	3	4	7	4	102
3	1	1	6	5	3	6	7	7	5	103
4	1	3	1	4	3	3	7	1	4	104
5	3	3	5	1	1	1	7	7	6	105
6	4	3	2	5	4	6	6	1	7	106
7	3	1	1	1	1	1	1	1	1	ISFFUE
8	1	3	1	1	1	1	1	1	1	ISFFUE
9	1	1	4	1	1	1	1	1	1	ISFFUE
10	1	1	1	4	1	1	1	1	1	ISFFUE
11	1	1	1	1	3	1	1	1	1	ISFFUE
12	1	1	1	1	1	3	1	1	1	ISFFUE
13	1	1	1	1	1	1	4	1	1	ISFFUE
14	1	1	1	1	1	1	1	7	1	ISFFUE
15	1	1	1	1	1	1	1	1	3	ISFFUE

IV.II Rule file

```
(FSP3) :                                        First consider FSP3
   1–4 :  (FST5)  :                             If FSP3 is 1–4, consider FST5
             1–4  :  ISFFUE                     If FST5 is 1–4, then ISFFUE
             5–7  :  102                        If FST5 is 5–7, then 102
   5–7 :  (FSS1)  :                             If FSP3 is 5–7, consider FSS1
               1  :  (FSS2)   :                 If FSS1 is 1, consider FSS2
                           1  :  103            If FSS2 is 1, then 103
                           2  :  101            If FSS2 is 2, then 101
                         3–7  :  104            If FSS2 is 3–7, then 104
             2–3  :  105                        If FSS1 is 2–3, then 105
             4–7  :  106                        If FSS1 is 4–7, then 106
```

(4) In the subsequent levels the implications of the initial hypotheses for pertinent problem concepts is considered. The consequences of a given set of hypotheses on dependent physical processes, and the second order (cascade) inter-action between sub-systems, are examples of aspects considered at two different levels. In another level, the pattern of events over time can be compared with anticipated trends.

Rule induction – EXTRAN (expert translator)

EXTRAN is a FORTRAN-based software package for building expert systems. It contains an inductive learning module, which induces 'If–Then' rules from examples. The induction process is based on Quinlan's ID3 algorithm, adapted to also handle numerical values[9]. The ID3 algorithm:

(i) selects a random subset of the database;
(ii) applies the CLS (concept learning system) algorithm to form a classification rule;
(iii) scans the entire database to find exceptions to the rule;
(iv) if there are exceptions, includes some of them in the considered subset and repeats step (ii).

The CLS algorithm repeatedly partitions data according to the variable with the greatest discriminatory power. ID3 is most suitable for reliable data, such as those obtained from experiments. It precludes comparing qualitative variables, which means that numerical descriptors must be defined for the analysis.

A simple example is given in Table 4 of an application of EXTRAN to the problem of fault diagnosis within the fuel sub-system of an engine (see Marine Applications section for more details). An EXTRAN problem is defined by a set of examples, in which factors that influence the solution (attributes) are linked to one of a range of possible outcomes or solution parameters (named classes in EXTRAN, but with no relation to the object classes mentioned above). Nine attributes of the engine have been identified in Table 4 as changing as a consequence of failures of the system, and 15 examples are given. The examples include six cases of specific component faults that can occur in the fuel sub-system. The others (ISFFUE) are examples of sensor failure. The value of each of the attributes indicates the absolute magnitude of the change, where 1 indicates no change from reference condition, and 7 is maximal. The resulting rule and explanation is given in the Rules section.

It is also necessary to consider the issues of rule validation and maintenance/updating with regard to induced rules. Rule validation is needed at two levels: to ensure the consistency and correctness of the data used (input validation) and to ensure the general applicability of the induced rules (output validation). Input validation is undertaken manually by experts. For output validation, a special rule validation software package is being developed as an extension to EXTRAN. This is used to establish whether the information used to produce the rules is incomplete, and if so, to identify the under-specified set of conditions. These conditions are then generated and tested against the decision rules, with experts asked to endorse or reject the decision values obtained. Thus the examples are extended, and the induced ruleset validated or altered.

For rule maintenance, it is necessary to retain the example set used to induce the original rule, in order to allow incremental changes to this set and consequent modifications to the rules. Software has been developed to automatically carry out the maintenance process after changes to the example-set.

AI development environment (MUSE)

The AI development environment MUSE[10] is currently the main tool available in the Society for developing expert systems. MUSE is an integrated set of software tools designed to support prototype development of experimental AI applications, particularly in real-time domains. A key feature of MUSE is that it provides a mechanism for packaging up prototype solutions and delivering them on a specified target machine[11].

The main MUSE components are summarized in Fig. 13. Poptalk is the main language, used for all the procedural code and for directly assessing the object system. It is an object-oriented language based on POP11[12] and Smalltalk[13]. It provides control structures such as loops and data structures such as lists, strings and variables. Poptalk is an untyped language (i.e. variables can hold a value of any type). Functions control the execution of a Poptalk program and assist in structuring it. Poptalk also provides a data capture facility consisting of a series of data channels. These channels can be linked to physical data sources such as sensors, to deliver current values of parameters to the system.

The rule system of MUSE consists of a Forward Production System (FPS) and a Backward Chaining System (BCS), whose function accords closely with that of the examples given previously. The MUSE FPS implements an OPS-style type 'match, resolve, execute' cycle[14]. The FPS is a collection of rules in the form 'if the condition is true then execute the action'. The condition part consists of a number of clauses. Each clause defines the conditions to be met before the rule can

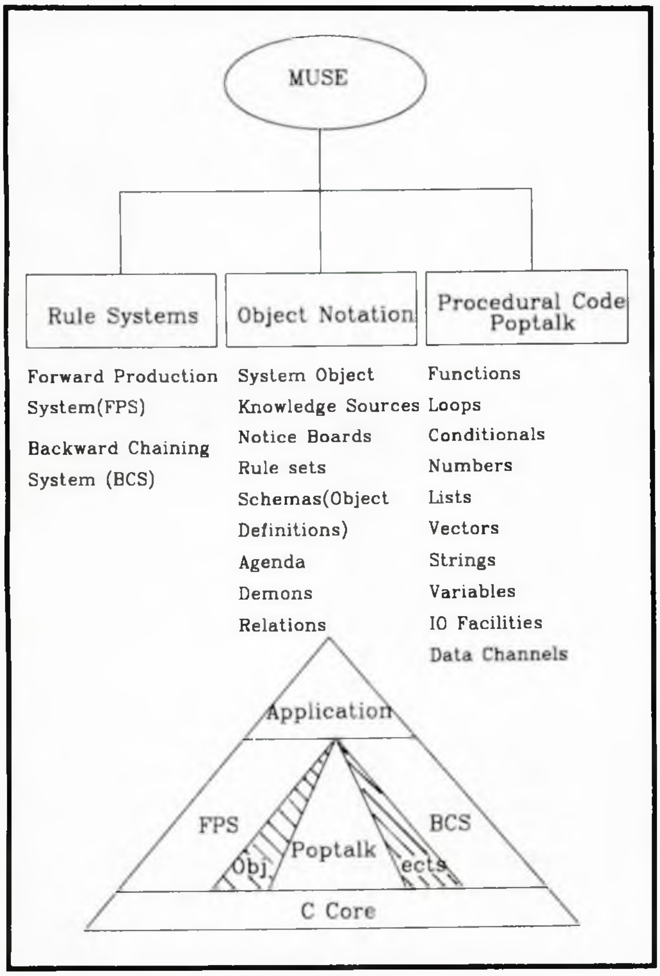

Fig. 13. MUSE components

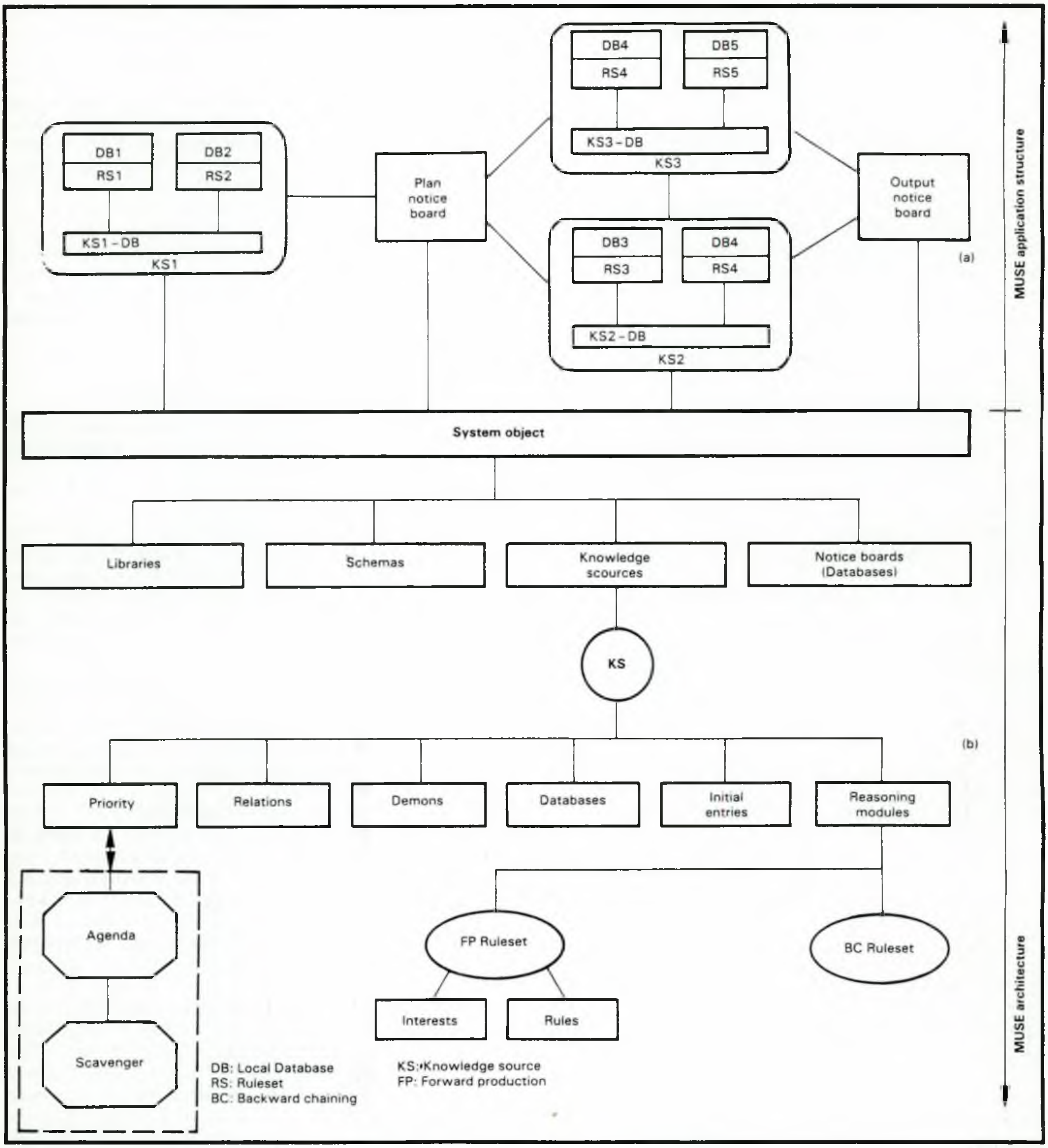

Fig. 14. Setting up an application in MUSE

be executed (such as the constraints on the values of certain attributes). The rule actions initiate the creation, modification or deletion of objects in one or more databases. The BCS system is similar in style and operation to Prolog. Thus, the BCS rule base contains statements of relationships between problem concepts in the form: X is true provided, $X1$, $X2$, $X3$ ... are true or X is true provided, $Y1$, $Y2$, $Y3$ ... are true.

The BCS program contains predicates which represent goals and are distinguished by having a unique name and a number of arguments. Each predicate consists of a collection of rules and facts. The rules are headed by a goal or conclusion and contain a set of clauses. Controlling the way that BCS applies the various rules or searches for relationships is not as explicit as in procedural languages. Depth-first search and chronological backtracking upon failure are the basic methods involved.

The third main element of MUSE is the object-based layer, which provides both access to object-oriented structures, and built-in facilities to provide many of the facilities necessary for the construction of advanced AI applications. Its components include demons, relations and facilities for system structuring and scheduling.

The function of demons has already been described. In MUSE, demons are essentially functions watching for the creation of instances of a class, or for changes to the attributes of an object. A number of different kinds of demon can be declared. 'Creation' demons are executed after an instance of a class has been created. 'Post-update' demons are executed after a particular attribute it has been monitoring has been updated with a new value. In MUSE, relationships between objects are specified by special objects called 'relations'.

Setting up an application in MUSE

The basic structure of any MUSE application is a set of reasoning modules termed 'knowledge sources' (KS) which communicate by shared access to particular databases. Every database is part of the 'system object', which represents the whole application.

The system object is an internally defined or built-in class which in MUSE is called a schema. It contains a number of built-in attributes or slots. The slots of the system object which are useful in structuring an application include 'knowledge sources', 'notice boards', 'schemas' and 'libraries' (Fig. 14).

The declaration of a knowledge source includes a number of pre-defined slots. It consists of a local database and one or more rule systems. The rule system consists of a set of forward or backward chaining rules plus interests in one or more databases ('notice boards') which the rules monitor and modify. Notice boards are employed as communication channels. All production rules monitor such databases for interesting changes. The blackboard approach previously described can be implemented in MUSE using notice boards.

In the example of a MUSE application given, three knowledge sources are utilized (KS1, KS2 and KS3). Each contains a central shared database (KS1-DB, KS2-DB, KS3-DB). These databases are visible to all the knowledge sources in the application and to Poptalk code. They can be specified in the 'interests' of any production rule set which will then monitor any changes. Each knowledge source has two rule sets (RS1/RS2, RS3/RS4, RS5/RS6). Each of these rule sets is either an FPS or BCS. Each has its own private database (DB1/DB6) which is only visible to each individual rule set.

When the initial information in a problem is processed in the knowledge source KS1, the results are communicated in the plan notice board. The plan notice board can trigger the two knowledge sources KS2 and KS3 which have access both to the plan and the final output notice board.

The order in which knowledge sources and other events are executed within the application is specified by the agenda. The agenda scheduling system is based on priority ordering. The agenda can be viewed as a fixed data structure with spaces representing different priorities. All the events which are ready to be executed are placed in the agenda by explicit calls from another part of the application or by the 'scavenger', which collects the production rule sets ready for execution when nothing else is running. The agenda also allows reasoning chains to be suspended when new external events occur, allowing critical or alarm events to be processed immediately.

MARINE APPLICATIONS

The category 'marine applications' primarily covers ships and platforms for drilling, construction and petroleum production. Expert systems for the ship and offshore platform sectors fall into two categories. Firstly, there are general advice dialogue-based systems. These can provide members of the industry, ranging from shipbuilding experts to a ship's crew and classification society surveyors, with the most updated information (rules, design standards and costs) in suitable marine technology and management areas. Interpretation of information and expert advice are the eventual goals of such systems[15,16]. The second category covers on-line expert systems designed to handle specific ship or platform functions. Such applications include voyage planning, engine fault diagnosis, and maintenance prediction. A control expert system could also handle the overall management of a ship.

A general marine technology expert system (MTES)

Marine transport technology can be broadly divided into ship design and ship operation. Both are multi-disciplinary tasks influenced by factors which are external or internal to the ship. External factors include environmental conditions such as winds and sea states, broad social and economic issues and the following operational constraints:

(i) legislative and charter party constraints;
(ii) environmental time-independent constraints such as port of call, arrival times and their elasticities and navigational restrictions;
(iii) environmental time-dependent constraints such as weather and sea state.

The environmental constraints have been recognized and treated as probabilistic problems. Certain operational requirements can be considered within the appropriate cost equations. The majority are either neglected or judged intuitively. Furthermore, the critical economic factors such as the level of world trade, individual government policies, oil prices, freight markets and financing are changing unpredictably, making any attempt at mathematical modelling virtually impossible.

Internal design factors are related to areas such as hydrodynamics, machinery, structures, stability, control, maintenance scheduling and management. In all these areas mathematical modelling can be applied, although in many cases the most serious obstacle is actually to define the problem. The necessary information for comprehensive mathematical modelling is often not available. Traditionally, preliminary solution parameters are derived and used sequentially as input for interconnected areas. The resulting initial design or operational parameters are then refined in an iteration loop. The widespread use of computer aided design (CAD) techniques makes an integrated synthesis approach suitable for this process.

Ship configuration changes such as damage or engine power reduction can produce operational constraints. Responses to motion are also in this category. The overall difficulty is that problems arising from the variable interaction of internal design or operation factors, according to criteria which are dependent on variable external factors, cannot be resolved satisfactorily.

Ship design and management have been based on computer implementations of the traditional process of analysis–synthesis–evaluation. These steps can be also accomplished by a Marine Technology Expert System (MTES) outlined in Figs. 15(a) and (b). The necessary input information will be made available from existing designs or operation procedures. The rules thus obtained, augmented by expert human knowledge, will form a ship technology knowledge base. This will be operated by an inference system to attain specific goals or to perform design synthesis.

An advantage of this approach is the adaptability of the system in formulating new solutions for changes associated with economic factors, different trading constraints and installation of new equipment. The effects of implementing each

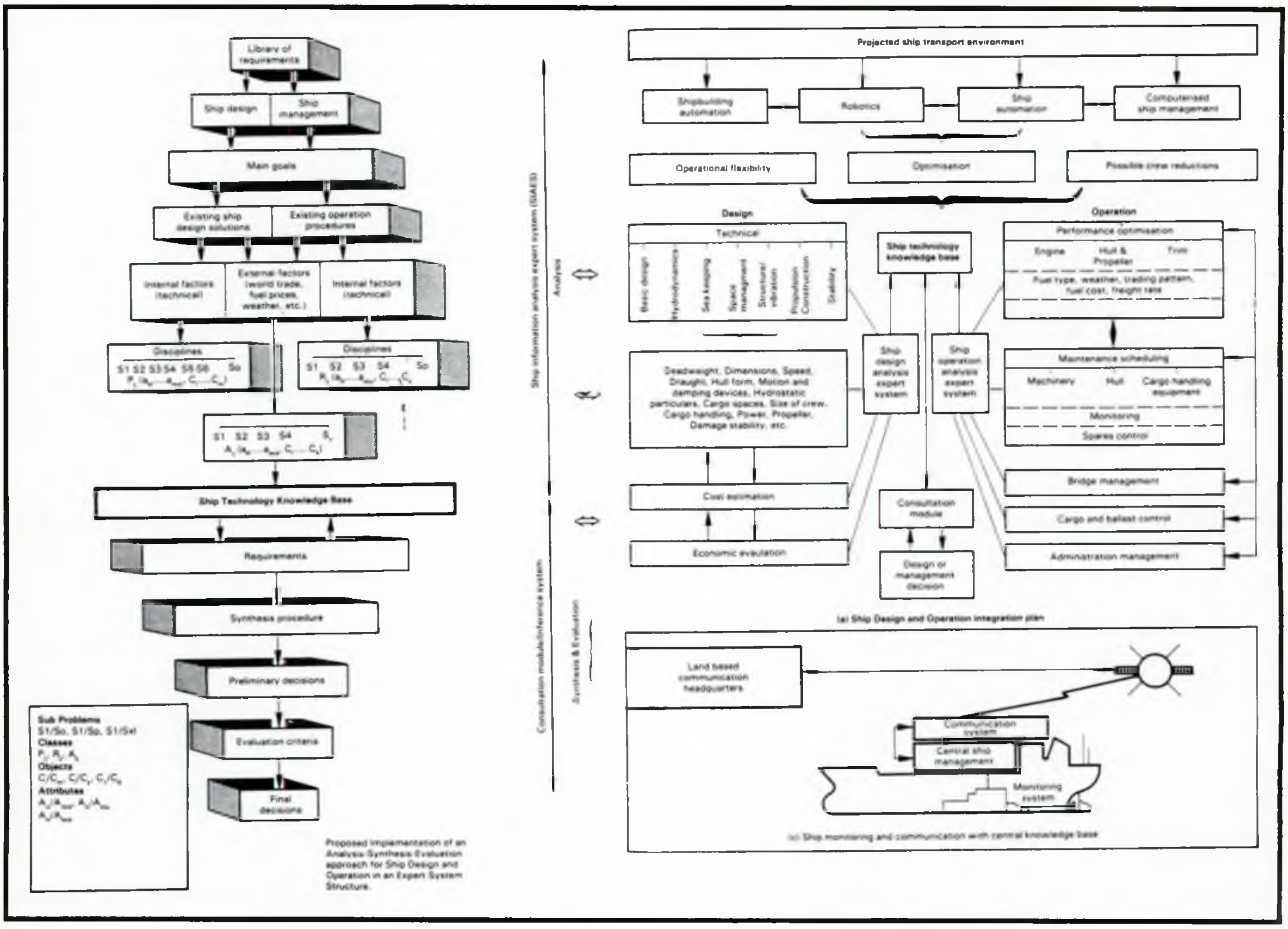

Fig. 15. Development of an integrated technology expert system

Table 5. Marine technology expert system applications

<u>Financial</u>	<u>Technical</u>
Investment decision	Fleet monitoring
Business plan	Voyage planning
Resource allocation	Maintenance planning
	Integrated bridge control
<u>Administration</u>	Ballast control
	Stowage plan
Inventory control	Fault diagnosis
Inventory schedule	Performance optimization
Crew	Design
Accounts	Design appraisal
	Classification
<u>Training</u>	Certification
Emergency procedures	
<u>Ship information</u>	
Weather	
Ship movements	

new decision can be monitored, and depending on the feedback, the overall plan can be refined.

Advances in communication technology allows monitoring of the effects of advice given by shore-based marine systems (Fig. 15c). In particular, feedback data on a ship's 'performance' is an essential input for reliability studies and for measuring goal attainment in the ship design and operational plan.

An area where an expert system such as the described MTES can be utilized by classification societies is in machinery, hull, electrical and refrigeration design appraisal and plan approval. The relevant classification rules and design standards can be identified by the system for any particular case. Where the expert system provides a real benefit is when it can provide advice on how to apply the rules and why certain formulae and appraisal procedures are considered to be applicable. Additionally, certainty factors (estimates of safety factors) can be attached to different calculation procedures. Expert systems which will advise on finite element analysis in relation to loading and component details are already under development.

Specific marine technology expert systems

A number of expert systems are currently being developed worldwide for specific marine applications. These are shown in Table 5. Some of them are briefly described below, including prototype systems under construction at LR.

Expert voyage planning. A number of commercial voyage planning programs are already available. Now, expert systems are being developed to add heuristic and learning

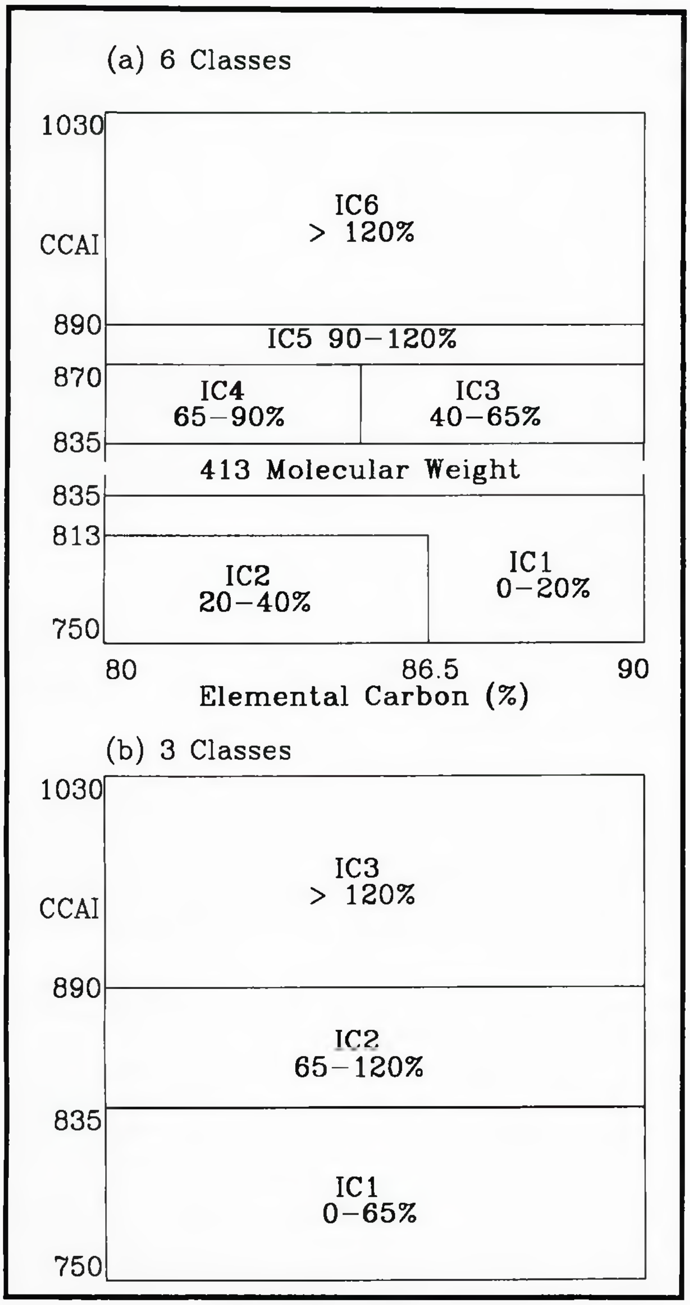

Fig. 16. IDP fuel maps

capabilities to such programs. The aim is to advise on an optimum voyage plan subject to constraints such as legislation, charter party details, route and weather, ship condition and ship responses to motion. Voyage planning is largely a determin-istic optimization problem with constraints for the expected motions (especially rolling), slamming, deck-wetness, structural response and the degree of propeller racing. The objective function is dependent on criteria such as arrival time, minimization of operating costs and minimization of rough weather damage. The expert system must thus identify the appropriate objective function, or objective function values, given information on port arrangements, bunkering ports, fuel costs, chartering details and future contracts. Heuristics can also be used to select a global track from knowledge base data. Digital maps and statistical weather forecasts will have a significant impact on future voyage planners.

A marine fuel oil characterization expert system (FOCES). A prototype fuel oil characterization expert system has been developed within an EEC-funded research programme currently near completion[17].

Any classification or ranking of fuel combustion behaviour can be based on the overall oxidation and pyrolysis routes of the three main hydrocarbon classes, saturates, aromatics and mixed polyaromatics[18]. However, non-linear 'interaction effects' between different hydrocarbons are present, producing a problem space of such complexity that traditional analytical treatment is virtually impossible. The engine response variations create additional complications. The statistical distribution of combustion parameters such as ignition delay and rate of pressure rise shows large deviations from the mean values. The magnitude of this variance is fuel-dependent[19]. An expert system approach was thus selected for this application.

FOCES is based on a two-stage characterization. The first stage is aimed at identifying a property area in a 'fuel map' within which a second-stage analytical procedure can establish a detailed correlation. This is similar to looking at a global country map to find the area for which details can be subsequently looked up on a local map. The fuel maps are produced as rules by computer induction and other statistical techniques for each engine area affected by fuel properties (such as ignition, combustion, fouling and wear). A set of typical ignition maps are shown in Fig. 16(a). The fuel classes IC1 to IC6 define set percentage ranges of change in the ignition delay period (IDP) from a reference condition (minimum IDP throughout the load range, gas oil). An important feature of the approach is the ability to simplify classification maps by broadening the ranges associated with the fuel classes. Only one fuel property is used in Fig. 16(b).

Diesel engine fault diagnostic expert system. A fault diagnostic expert system is under development in the condition monitoring research programme, co-ordinated by Lloyd's Register, within the 'U.K. efficient ship project'[20]. The fault diagnosis and fuel characterization systems will be integrated with a simulation module in an advanced condition monitoring system which is illustrated in Fig. 17[21]. The objectives of the diagnostic system, named DEEDS (Diesel engine expert diagnostic system) are:

(i) increased reliability and safety for diesel engines;
(ii) the ability to advise non-qualified personnel on the location, severity and causes of faults;
(iii) the provision of clear and correct system state information to support engineers handling abnormal operating conditions;
(iv) early warnings of component condition for maintenance planning.

The functional requirements of the system include:

(i) early detection of specific engine component or sensor faults;
(ii) diagnosis of faults (including multiple independent faults). (This can be linked with alarm systems to provide causes for reported alarms.)
(iii) a warning system for interfacing to maintenance planning;
(iv) identification of probable faults in a situation of fewer sensors;
(v) assessment of confidence levels for the diagnosis;
(vi) assessment of fault severity;
(vii) an interface to the operator, to permit requests for additional information, and to inform the user of the diagnosis and the implications for the monitored components.

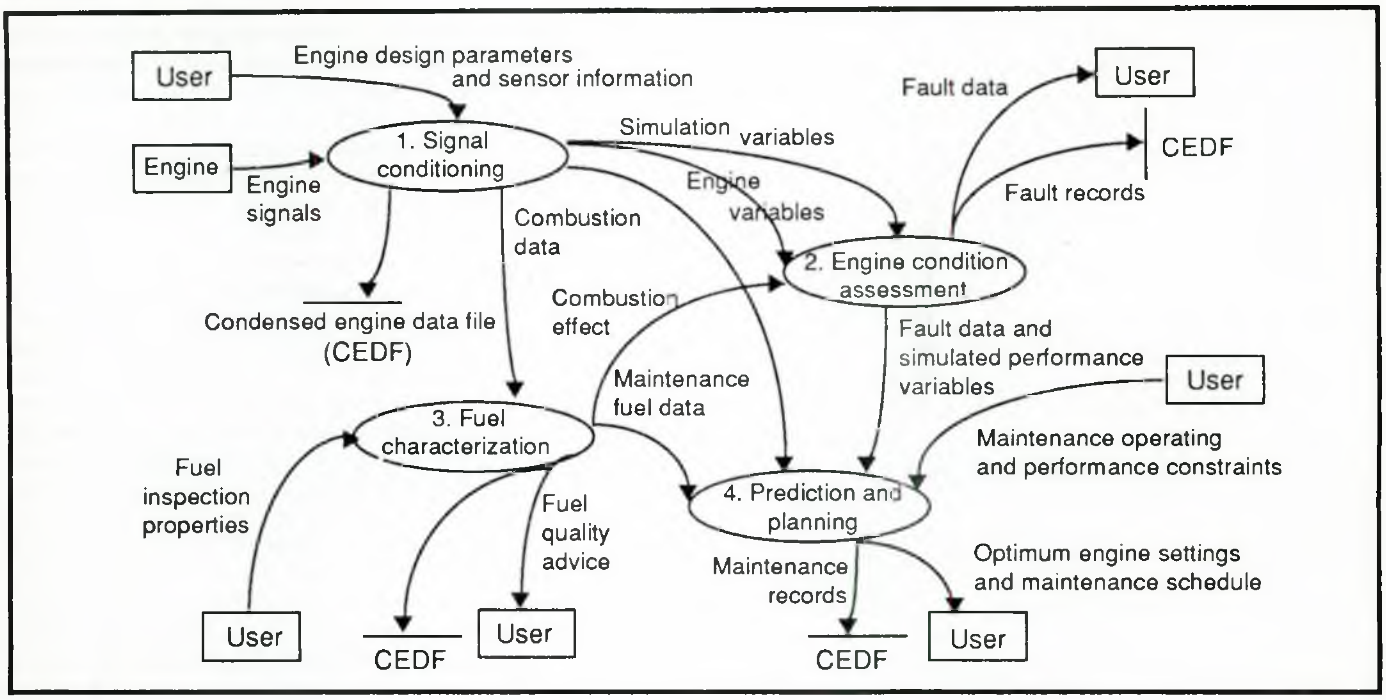

Fig. 17. Systems overview data flow

Table 6. Main stages of DEEDS diagnostic process

	Process	Technique
1.	Sensor diagnostics	Kalman filters, heuristics
2.	Fault detection	Generation of reference data by mathematical simulation and comparison
3.	Identify engine subsystem(s) with faults	Heuristics Causal model
4.	Identify all plausible faults in a subsystem	Modified 'set covering' model
5.	Generate and test plausible single faults	Pattern matching, fuzzy set techniques, heuristics
6.	Generate and test multiple fault hypothesis	Heuristics, reliability data, decision tree analysis, causal model, simulation
7.	Refinement of fault hypothesis	Temporal logic, causal model
8.	Fault hypothesis validation	Simulation
9.	Consequence analysis	Heuristics, temporal logic, causal model

The development of this expert system is based on obtaining an experimental database of associations between symptoms and engine faults (Fig. 18). An extensive engine test programme is being undertaken at several research centres, in which 80 fault conditions are artificially imposed into engine components, and the resulting changes in sensors' signals from the engine are recorded.

In the final application, the fault diagnosis expert system is closely linked with an engine simulation model. This responds to changes in engine speed, load, ambient conditions and fuel oil quality by predicting the expected engine performance. When sensor data indicate significant deviations from these simulation reference values (outside the expected range of accuracy of the sensor), a set of deviations is generated and passed to the fault diagnosis system. Fault hypotheses are generated and tested, and then confirmed if sufficient supporting evidence can be found in the reactions of other subsystems, and in the combustion process. The simulation model also provides final confirmation for any fault identified by the expert system. The main stages of DEEDS are given in Table 6 and are described in some detail in ref. 22.

Expert maintenance system. Expert maintenance systems both for diesel engines and for the whole ship structure are under development within LR, the latter under the KBSSHIP programme. Their objective is to produce condition-based maintenance schedules which maximize ship availability, subject to operational constraints. Optimization of maintenance schedules for hull and machinery produces benefits in ship availability, crew requirements, direct maintenance costs and spare parts inventory control. The knowledge base of the maintenance expert system must contain the following three sets of information.

(i) Planned or reference maintenance schedules for the appropriate items.
(ii) Knowledge of factors which produce increased maintenance activity (such as fuel quality and heavy weather).
(iii) Heuristics and/or models specifying deterioration patterns for each component for each of the factors in (ii).

Initial estimates of when maintenance is required will be provided during the commissioning of the equipment from the manufacturers' planned maintenance schedules. These will be constantly updated by demons, triggered by factors responsible for increased maintenance activity [list (ii)]. The organization and scheduling of predictive maintenance itself is a form of planning problem. Heuristics can be employed to select appropriate parameters and/or models for estimates of change in maintenance schedules. It is envisaged that it will be necessary to monitor actual deterioration trends (for example, engine lubrication condition and accumulated debris) to refine the predicted maintenance plan.

Integrated ship control. Systems integration on ships refers usually to engine room control and cargo handling and

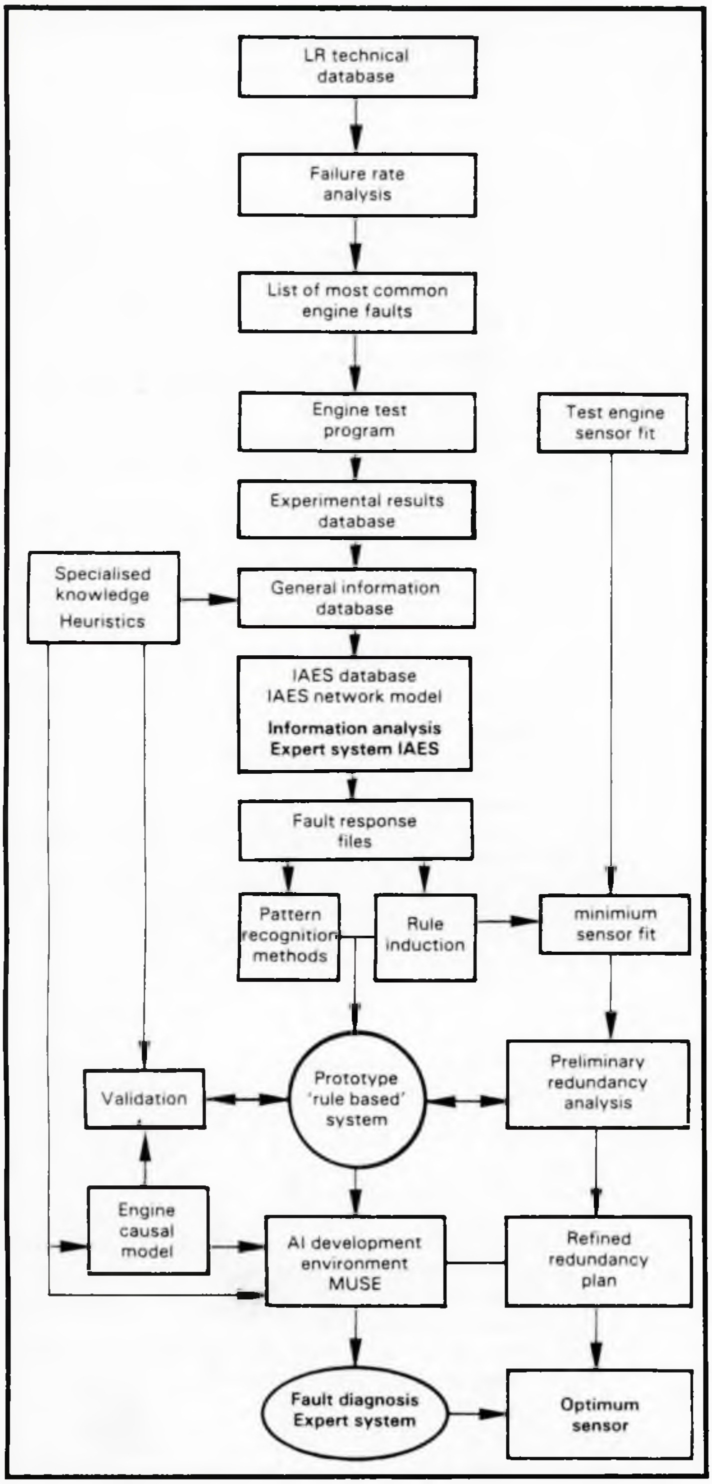

Fig. 18. Development of a fault diagnosis expert system

sometimes communication, navigation and other systems. Total integration is, however, the goal of many research and development activities and particularly of the many national 'efficient ship' projects.

Developments to integrate bridge control systems are on the way. The typical bridge operational system consists of the steering control, magnetic compass, engine room telegraphs, radar, radio, gyro, echo-sounding and satellite equipment. Electronic charts are also finding their way into the bridge area. Integration of the bridge control with the machinery control, cargo handling and surveillance systems into a central ship management area will eventually take place, but can impose overwhelming demands on both the computer hardware and the operators.

Conditions can change rapidly during an emergency situation, so that a course of action, initially correct, could moments later be disastrous. Assimilation and interpretation by the bridge officers of all the information available by all the above systems is almost impossible. Furthermore, no computer, regardless of its speed, can evaluate quickly enough all possible outcomes presented by an emergency. Efficient inference mechanisms augmented by expert heuristics can offer a solution to this problem. A number of expert systems for intelligent process control in real-time land-based industry have already been developed and marketed. They usually act as alarm management systems, by monitoring large numbers of process variables and alarm signals.

Dynamic positioning expert systems. A dynamic positioning system is a computer- and thruster-assisted manoeuvring system. It provides a means of controlling the position and heading of a vessel or mobile offshore unit within pre-defined limits. Thrust is applied to overcome disturbances from wind, currents, tides and waves. Different types of thrusters and/or a controllable-pitch propeller are employed for this task.

Controllers for dynamic positioning application are based on established multi-variable control theory. Normally, because ship motions have significant non-linearities, it is necessary to change control parameters (gain and phase) to obtain satisfactory control over the power range of a ship. Expert systems have been developed in this area to act as supervisory controllers. Their role is to interpret prevailing conditions and to select a control mode with the appropriate controller parameters.

Offshore platform design. The design of an offshore platform is probably the most complex of all offshore design problems. It may require up to 10 million man-hours and involve hundreds of experts. The overall objective in developing a field is to exploit a petroleum reservoir in the most economical manner. Suggested development concepts consistent with a production plan are reviewed and refined in the light of environmental, technical, and economic constraints. These constraints include weight, cost and time estimates, phasing of activities and field development economy. Techniques developed for planning and monitoring expert systems are particularly suitable here. Management, communication and monitoring aspects can also benefit from the use of expert system technology. There are several research projects to develop expert systems to assist in offshore platform design.

CONCLUSIONS

The industrial revolution has been succeeded by the information technology era – a mixture of computers and communications encompassing knowledge-based systems. Information technology can be used to raise the technological level of ship transport to enable the generation of new products and services in a technically competitive environment. The use of this technology in a coherent and disciplined way requires comprehensive knowledge of its advantages and limitations. Such knowledge is also necessary for the independent assessment of its impact on safety and ship integrity.

Future developments towards integrated ship management systems rely on developing new techniques for diagnosis, prediction, planning and control and for interpreting information which is unclear, uncertain and unreliable. Expert systems are particularly suitable for these problems and will inevitably become an essential element in marine technology. Another advantage of such uses is the accessibility of the knowledge base, allowing easy updating when parameters change due to the inherent dynamism of the ship transport and offshore platform environment. Expert systems technology also serves to provide a means to codify and preserve scarce and/or expensively gained knowledge, storing the expertise of retiring personnel, and making available to less experienced individuals at least some of the wisdom gained by the best practitioners of a craft.

The benefits of expert systems in marine applications have now been successfully demonstrated. The development of a number of expert systems and of a methodology for their construction has awakened interest within the marine community, and will be an important contribution to the long-term revival of the shipping industry.

ACKNOWLEDGEMENTS

The authors would like to thank the Department of Trade and Industry, the Science Engineering Research Council, the Council of the European Communities and many industrial and research participants for their support in the Society's research programme. The authors would particularly like to thank Cambridge Consultants Limited, their colleagues within Lloyd's Register and the members of the KBSSHIP and condition monitoring projects for their participation and advice and the Committee of Lloyd's Register for permission to publish this paper.

REFERENCES

1. R. K. Lindsay, B. G. Buchanan, E. A. Feigenbaum & J. Lederberg, Applications of Artificial Intelligence for Organic Chemistry: the DENDRAL Project, McGraw–Hill (1980).
2. E. H. Shortliffe, Computer-based Medical Consultation: MYCIN, American Elsevier (1976).
3. B. G. Buchanan & E. H. Shortliffe, Rule-based Expert Systems: THE MYCIN Experiments of the Heuristic Programming Project, Addison–Wesley (1983).
4. F. Hayes-Roth, D. A. Waterman & D. B. Lenat (eds.), Building Expert Systems, Addison–Wesley (1983).
5. J. Doyle, A Truth Maintenance System, *Artificial Intelligence*, vol. 12 (1979).
6. R. S. Michalski, J. Carbonell & T. M. Mitchell, Machine Learning: An Artificial Intelligence Approach, Tioga Press (1983).
7. M. Minsky, A Framework for Representing Knowledge in Mind Design (J. Flangelnad, ed.), The MIT Press (1981).
8. P. S. Katsoulakos, J. Newland & T. Ruxton, Monitoring Databases and Expert Systems in the Development of Fault Diagnostics, International Conference on Computers in Engine Technology, I. Mech. E., Oxford (1987).
9. J. R. Quinlan, Discovering Rules by Induction in Expert Systems in the Microelectronic Age (D. Michie, ed.), Edinburgh University Press (1979).
10. D. Reynolds, From Development to Delivery: System Integration, Knowledge-Based Systems (KBS) 1986, Online Publications (1986).
11. D. Reynolds, MUSE, A Toolkit for Embedded, Real Time, AI, Cambridge Consultants Ltd. (1986).
12. R. Burstall, J. Collins & R. Popplestone, Programming in POP-2, Edinburgh University (1977).
13. A. Golderg & D. Robson, Smalltalk-80: The Language and its Implementation, Addison–Wesley (1983).
14. C. Forgy & J. McDermott, OPS: A Domain Independent Production System Language, Proc. 5th IJCAI (1977).
15. P. S. Katsoulakos, Expert Systems in Ship Technology, Seminar: Applications of High Technology to Shipping, Athens (1986).
16. P. S. Katsoulakos, Expert Systems in Current and Future Ship Technology Development, Motorship International Propulsion Conference (1987).
17. P. S. Katsoulakos, C. Pontikos & J. Stansfeld, Integration of a Fuel Characterization Expert System with Diesel Engine Simulation, CIMAC Conference, Warsaw (1987).
18. P. S. Katsoulakos, C. Pontikos, C. M. R. Wills, G. Armstrong, D. Richardson & T. Piyasiri, Assessment of the Combustion Performance of RFOs, International Congress on Combustion Engine, CIMAC, Oslo (1985).
19. D. Richardson, D. Armstrong, C. Pontikos & P. S. Katsoulakos, The Analysis of Marine Fuels with Reference to Diesel Engine Performance, *NECIES Transactions* (1987).
20. J. T. Stansfeld, J. L. Buxton & P. S. Katsoulakos, Condition Monitoring and Classification: The Lloyd's Register Approach, Fremtidens Skibsdrift Conference, Copenhagen (1987).
21. J. T. Stansfeld, P. S. Katsoulakos, T. Ruxton & L. Isaias, The Application of Simulation Techniques in Advanced Engine Monitoring and Control, Eighth Ship Control Symposium, The Hague, The Netherlands (1987).
22. P. S. Katsoulakos, C. P. W. Hornsby & R. Zanconato, DEEDS: The Diesel Engine Expert Diagnostic System, Intl. Conf. on Maritime Communications and Control, *Trans.I.Mar.E.(C)*, vol. 100, conf. 1, paper 22 (1988).

Discussion

M. Houlder (Postgraduate, University of Sheffield) You are at least looking towards safety-critical applications. Can formal methods provide the expert system software validation that is, therefore, required? What concerns me is the indeterminacy intrinsic to expert systems coming from the use of statistical techniques and confidence levels and so on.

P. S. Katsoulakos (LR) This point is quite correct. Indeterminacy at present does cause problems in safety-central applications. However, the problems of verification are very much broader than this, and few real safety-critical systems existing today could be said to have been verified formally at all. We look forward to the time when the verification and validation techniques being developed today can be applied in the expert systems area, and in a related research programme (the REDO project), Lloyd's Register is attempting in part to address this problem.

R. V. Thompson (President, I.Mar.E.) Could the authors underline the fundamental advantages and disadvantages of using:

1. real time simulation techniques;
2. expert systems in, for example, engine control systems.

Is the name 'expert' system misplaced in that the activity defined essentially consists of a 'normal' operating control system upon which is superimposed a series of hierarchal logic queries or functions?

P. S. Katsoulakos (LR) The advantages and disadvantages of using on-line simulation models or expert systems for engine monitoring and control are as follows.

1. Simulation
   a. Advantages. This provides a good baseline for detection of symptoms of a fault, which varies with operating conditions and content environment. Also on-line simulators can be used to model decisions postulated by on-line expert systems, thus providing a validation facility.
   b. Disadvantages. Speed.
2. Expert systems
   a. Advantages. Expert systems can supplement engine control systems by providing an 'intelligent' filter for input information. In particular it is useful when multi-variable inputs need to be evaluated in order to establish control input variables.
   b. Disadvantages. Safety-criticality and verification. Expert systems are far more general than engine control systems. The main function of expert systems in control is to provide intelligent interfaces.

F. D. Petit (Crown Agents for Overseas Government Administrations) In times of cheap fuel (such as now) the shipowner tries to make some money on profitable operations. How can you persuade the shipowner to invest now in expert systems?

In times of expensive fuel (perhaps by 1992/3) the bad times may return and the shipowner will need the full help of expert systems – but by then it may be too late for him to invest in them.

If fuel costs become very great then perhaps other factors will again come to the fore – very efficient hull design, sail-assisted ships, Flettner rotors and other alternative power systems.

In the light of the above how do you propose to interest the shipowner in the appropriate expert system for each new development?

P. S. Katsoulakos (LR) Mr. Petit's question reflects the priorities of the marine industry and their possible relationship with expert systems. Expert systems can be applied for two distinct purposes:

1. Integrated ship control leading to crew reduction.
2. Advisory expert systems coupled to different ship functions such as fuel characterization, maintenance planning, loading, and others. Such systems will be aimed at optimizing performance, reliability and safety. Their commercial development will depend on market demand.

It is likely that each new future development will make use of 'intelligent front ends' for communication with the central ship control and the operators. In essence Mr. Petit's question can only be answered by demonstrating first at a research level, and then in practice, that the systems will provide some commercial benefit to their users, whether in terms of reduced costs or improved service.

D. R. Cusdin (Shell Seatex) I would like to congratulate the authors on a good, informative paper which provides a welcome introduction for non-experts into expert systems in the marine industry.

Although my company is developing computer programmes as an aid to voyage optimization and thus to improving profitability, we have not yet approached the stage where the computer and its expert systems take full control of the ship handling operations. If such systems are intended to replace experienced crew members, no doubt an experienced backup programmer/engineer will still be needed in the case of black-out.

My company has however developed an expert system called SEAFUEL,[1] which is a computer program built around a knowledge base associated with the storage, handling and pre-treatment of marine residual fuel oil on board a ship.

This system can be used for on board trouble shooting or as a training aid, and it probably falls between categories 1 and 2 in Table 1 of the paper.

Our experience shows that the system must be 'user friendly'. I do not know if this is the same as the author's terms of 'usefulness' and 'flexibility'. Simple uncluttered screens must be used with simple dialogue and a simple, logical approach. Fig. 1 over the page shows such a typical screen. 'Help' screens aid the user, and Fig. 2 (over the page) shows the help commands available. The 'WHY advice' and 'HOW X' have proven to be very useful.

The authors have explained the differences between forward and backward chaining, and I would like to ask in which application the authors consider these should be used?

Again, I would like to thank the authors for a most interesting paper.

Reference

1. R. Ives & I. Archibald, SEAFUEL - An Expert System for Residual Fuel Oil Handling, International Conference on Maritime Communications and Control, *Trans.I.Mar.E.(C)*, vol. 100, conf. 1, paper 25, pp. 231–236 (1988).

```
The ESP ADVISOR. Vers 2.05 Copyright (c) 1984
Expert Systems International
GOAL:  system        SECTION:  control
-------------------------------------------------
In which part of the plant has the fault occurred ?
    (1)  –   Storage tank
    (2)  –   Centrifuge
    (3)  –   Filter
    (4)  –   Pump
    (5)  –   Heaters
-------------------------------------------------
Enter the number of the relevant entry   : 1
-------------------------------------------------
```

Fig. 1. A simple user-friendly 'Prompt' screen

P. S. Katsoulakos (LR) Mr. Cusdin's comments indicate that expert systems have already found practical applications in the marine industry. The SEAFUEL system is a good example of perhaps the most useful type of expert system today; the 'Dialogue mode' as referred to in Table 1 of the paper. Most of the expert system features described in the paper are for 'real time' applications. Such systems are being constructed in the Society's research programme for decision support rather than to take control of the ship handling operation. It is the opinion of the authors that the latter situation will not be realized in the foreseeable future.

Forward and backward chaining systems are combined in the majority of applications. In general backward chaining systems are used when we need to prove a well-defined goal, for example, when there are a limited number of possible solutions, one of which is likely to be the case. In the diagnostic system that we are developing the forward chaining system is used to arrive at a set of possible faults by examination of dynamic data.

C. Levantis (Det Norske Veritas) With respect to definitions of artificial intelligence, exactly how intelligent are these expert systems? Is it not really a comparison between a mathematical model and real monitoring?

P. S. Katsoulakos (LR) A number of definitions have been included in the paper to differentiate the meanings of the fundamental concepts in AI. Expert systems simulate the use of expert knowledge to solve problems in a specific domain. It can be safely assumed that very few can be termed 'intelligent'. Even learning capabilities in expert system are in their infancy. Interpretation of the differences between a mathematical model and actual data from monitoring processes can be a suitable application for expert systems, but this is not the only function of these systems.

R. J. Clements (The Marine Technology Directorate Ltd.) I would like to congratulate and thank Dr. Katsoulakos for an interesting and instructive paper. I have found it most useful and look forward to understanding better these 'black boxes' that all the experts are talking about for the future.

An impressive range of expert systems has been described which are being developed within LR but these are not the only ones and many companies and universities are developing their own in different areas. Can they be made intelligent enough to talk to each other and consequently be used to shorten the time required to produce the overall model or the ultimate expert system?

P. S. Katsoulakos (LR) The use of advanced information processing and communication tools offers endless possibilities. Collaborating expert systems have been briefly described in the paper. Distributed AI offers the advantages and disadvantages of any type of distributed systems. It is possible to divide marine technology into many application areas for which expert systems can be developed in selected centres of expertise. A supervisory system could then communicate with all the sub-expert systems providing the 'ultimate marine system'. The notion of combining many expert systems in the same application to produce a more advanced expert is probably unrealistic, due to the complexities of data management and the variety of knowledge representation and inference strategies used in different systems.

```
The ESP ADVISOR. Vers 2.05 Copyright (c) 1984 Expert Systems International
GOAL: tank-type     SECTION:  control
--------------------------------------------------------------------------------

              TABLE        OF        AVAILABLE        COMMANDS

YES / NO      —  Question  TRUE/FALSE       TRACE    on / off —  Proof trace on/off
UNKNOWN       —  Answer is not known        PRINT    on / off —  Advice print on/off
EXPLAIN / ??  —  Explain the question       LOG      on / off —  Logging mode on/off
WHY   N       —  Why qstn. to depth  N      SAVE   X          —  Save session to X
WHY advice    —  Why last advice given      STATUS            —  Status of beliefs
HOW   X       —  How param. X proven        STATUS   X        —  Status of param. X
SHOW          —  Display current param.     SHOW   X          —  Display parameter X
VOL    X      —  Volunteer parameter X      RECAP             —  Recap advice so far
CHANGE   X    —  Change parameter X         DIRECTORY         —  Directory of params.
HELP   X      —  Help with command X        HELP  user        —  Display user commands
QUIT          —  Quit current section       EXIT              —  End of consultation

--------------------------------------------------------------------------------
Enter the number of the relevant entry    :  help
--------------------------------------------------------------------------------
                          PRESS ANY KEY TO CONTINUE
```

Fig. 2. SEAFUEL 'Help' screen

M. Phillips (Mobil Oil Co. Ltd.) An expert system is a computer-based knowledge system which contains a large number of rules and facts from which it can infer solutions to questions put to it by myself (user).

What problems exist in terms of:

1. handling inconsistencies;
2. systems knowing their own limits;
3. the ability to build in common sense?

P. S. Katsoulakos (LR) Mr. Phillips has recognized some real problems within the current generation of expert systems. Many methods of handling inconsistencies are in use. The handling of inconsistencies can be viewed from two different perspectives. When the information provided by the user (or by sensors in real-time control) is inconsistent, procedures can detect this (at a cost in processing power) and react by identifying the likely confidence in each fact (e.g. from its RECENCY or its SOURCE), and deleting the least likely inconsistent data item. They can also (in some cases) identify and remove all logical dependencies from this data item. However, in commercial systems, this is generally an extremely costly and time consuming exercise.

Expert systems operate within their own limits and some have the ability to communicate that they cannot find a solution when they run outside those limits. Users must be aware that, as with experts, performance degrades as they reach the boundaries of their knowledge.

Today it is almost impossible to build in common sense into expert systems. This relies on a vast mass of tacit knowledge, developed by experience and communication with others over many years. Attempts to build very large knowledge bases are being made, but their success is still uncertain.