CHAPTER 5

DOMAIN - EXPERT SYSTEM RELATIONSHIP

5.1. **Introduction**

The process of building an expert system is inherently experimental. In order to have successful development of an expert system in a domain, different potential issues have to be fixed up which demands a thorough analysis. This chapter is meant for fixing up such two potential issues, namely (i) why it is an expert system domain, and (ii) what requirements the domain lays on an expert system. These two potential issues should unfold some important matters relating to design and implementation.

This chapter has been organised as follows. In the next section, a brief discussion on AI and expert system technology has been provided. After a brief introduction to artificial intelligence (AI) some potential issues such as categories and application areas of expert systems, trends of applications of expert systems, typical architecture of an expert system, desirable features of an expert system, different stages of an expert system development, types of expert systems have been discussed. In section 5.3, we have tried to explain why the present problem domain may be considered suitable for an expert system domain. In section 5.4, an attempt has been made to find out what requirements the domain lays on an expert system. Lastly, a discussion has been provided.

5.2. **AI and expert systems technology : an introduction**

It is well known that several computer systems have been built over the past few decades that can perform tasks which are comparable to many human mental activities, such as writing computer programmes, doing some mathematics or engaging in common sense reasoning or understanding natural languages or even driving automobile. There are also computer systems that can diagnose diseases, plan the synthesis of organic chemical compounds, solve differential equations in symbolic form, understand limited amounts of human speech and natural language text, analyse electronic circuits or write small computer programmes to meet some formal specifications - we shall say that such systems possess some degree of artificial intelligence (AI). The field of AI largely has an emperical engineering orientation, drawn from a loosely structured but growing body of computational techniques. AI systems are developed, undergo experimentation, and are improved.

A second motivation for AI research is a large scientific goal of constructing an information processing theory of intelligence. If such a science of intelligence could be developed, it could guide design of intelligence machine as well as explicate intelligent behaviour as it occurs in human and other animals.

Artificial intelligence technique has gained tremendous importance in the past few years because of its crucial importance in connection with the research of Future Generation of Computer Systems (FGCS), particularly fifth generation computers as envisaged by Japanese scientists. The fifth generation computer research objectives are :

(1) The core function of inference, association, and learning have to be realised,

(2) AI software engineering is to fully utilised for the above mentioned functions,

(3) Intelligent management and retrieval methods from knowledge based machines, both hardware and software,

(4) Full exploitation of the advances made in the field of pattern recognition, natural language processing and image analysis to realise man machine interface, to reduce the alienation between human beings and the machines,

(5) Intelligent software to be provided by machines for the above mentioned uses in the society [1-5].

All these objectives are to be realised through what is known as knowledge based inference making methods.

In its three decades history one area of AI that can claim a large measure of responsibility for the current AI awareness in the world is expert system technology which are computer software that embody human expertise. Human expert in any field are a scarce commodity in our society, for example, a medical practitioner in some remote village may be competent but will be in great difficulty, when confronted by a patient with unfamiliar symptoms. If the specialist is not accessible the patient may have to settle for inadequate treatment. The scarcity of expertise exists in almost all fields, such as repairing automobiles, drilling for oil, managing a stock portfolio, or analysing chemicals. In all these and in many other cases there are times when access to the knowledge, experience, and judgement of an expert is an invaluable asset and unfortunately for citizens and fortunately for knowledge engineers there are more problems than experts. One solution to the dilemma is the expert systems technology, which can help with new approaches to organisation, productivity, expertise, knowledge, competence, and smart automatic equipments that can act as intelligent assistance to human experts as well as assisting people who otherwise might not have access to expertise. It is different from data base programme that retrieves facts that are stored while an expert system uses reasoning to draw conclusion from stored facts.

5.2.1. **Categories and application areas of expert systems**

We have mentioned that expert systems may be applied to any situation that normally requires human expertise. One can divide typical expert system applications into twelve functional categories [6,8] shown in table 5.1. In table 5.2, we indicate application areas for which some expert system has been developed [7,8].

Table 5.1

Generic categories of expert system applications

Category	Problem addressed and application types
Interpretation	Inferring situation descriptions from sensor data for speech and image analysis and surveyllance.
Prediction	Inferring likely consequences of given situations for weather forecasting and crop estimation.
Diagnosis	Inferring system malfunctions from observables for medical and electronic fields.
Design	Configuring objects under constraints for circuit layout and CAD.
Planning	Designing actions - automatic programming and military planning.
Monitoring	Comparing observations to expected outcomes - for power plant and fiscal management.
Debugging	Prescribing remedies for malfunctions for computer software.
Repair	Executing plans to administer prescribed remedies for automobiles / computers.
Instruction	Diagnosing, debugging and repairing student behaviour
Control	Governing overall system behavior for - air traffic control and battle management.
Prescription	Recommending solutions to system malfunctions.
Selection	Identifying the best choice from a list of possibilities.
Simulation	Modeling the interaction between system components.

Table 5. 2

Application areas for expert system

Agriculture	Physics
Chemistry	Process Control
Computer System	Space Technology
Electronics	Business
Engineering	Power Systems
Geology	Transportation
Information Management	Mining
Law	Communications
Manufacturing	Education
Mathematics	Environment
Medicine	Image processing
Meteorology	Science
Military Science	Aerospace

In Agriculture, different expert systems have been reported such as PLANT/CD, PLANT/DS, POMME. PLANT/CD [9], for example, predicts the damage to corn due to the black cutworm. The system uses knowledge about the particular field being studied, such as moth trap counts, field weediness, larval age spectrum, soil condition, and corn variety to predict the degree of damage the cutworm will cause. The system uses a combination of rules and a set of black cutworm simulation programs to produce the predictions. Knowledge is represented as rules accessed by a backward chaining control mechanism.

In Chemistry, different expert systems have been reported such as CONGEN, CRYSALIS, C-13, DENDRAL, GA1, META-DENDRAL, MOLGEN, OCSS, SECS, SEQ, SPEX, SYNCHEM, SYNCHEM2, TQMSTUNE. DENDRAL [10], for example, infers the molecular structure of unknown compounds from mass spectral and nuclear magnetic response data. The system uses a special algorithm developed by J. Lederberg to systematically enumerate all possible molecular structures; it uses chemical expertise to prune this list of

possibilities to a manageable size. Knowledge in DENDRAL is represented as procedural code for the molecular structure generator and as rules for the data-driven component and evaluator.

In Computer Systems, different expert systems have been reported such as CRIB, IDT, ISA, DART, MIXER, R1-SOAR, TIMM/TUNER, XCON, XSEL, YES/MVS. DART [11], for example, assists in diagnosing faults in computer hardware systems using information about the design of the device being diagnosed. The system works directly from information about the intended structure and expected behavior of the device to help find design flaws in newly created devices. The system has been applied to simple computer circuits and the teleprocessing facility of the IBM 4331. DART uses a device-independent inference procedure that is similar to a type of resolution theorem proving, where the system attempts to generate a proof related to the cause of the device's malfunction.

In Electronics, different expert systems have been reported such as ACE, BDS, CADHELP, COMPASS, CRITTER, DAA, DFT, EL, EURISKO, FG502-TASP, FOREST, IN-ATE, MESSAGE TRACE ANALYZER, NDS, SYN, PALLADIO, PEACE, REDESIGN, SADD, SOPHIE, TALIB, TRANSISTOR SIZING SYSTEM. SYN [12], for example, assists engineers in synthesizing electrical circuits. The engineer inputs partially specified circuit diagrams and constraints on particular circuit components, and the system combines this information with knowledge about constraints inherent in the circuit structure to specify the circuit completely (e.g. fill in the impedance of resistors and voltages of power sources). The system combines constraints by using symbolic algebraic manipulation of the formulas describing the circuit components.

In Engineering, different expert systems have been reported such as CONPHYDE, DELTA, NPPC, REACTOR, SACON, SPERIL-I, SPERIL-II, STEAMER. SACON [13], for example, helps engineers determine analysis strategies for particular structural analysis problems. The engineers can then implement this strategy with MARC, a program that uses finite-element analysis methods to simulate the mechanical behavior of objects. SACON identifies the analysis class of the problem and recommends specific features of the MARC program to activate when performing the analysis. SACON uses knowledge about stresses and deflections of a structure under different loading conditions to determine the appropriate strategy. Structures that can be analyzed included aircraft wings, reactor pressure vessels, rocket motor casings, and bridges. SACON is a backward chaining, rule-based system impliemented in EMYCIN.

In Geology, different expert systems have been reported such as PROSPECTOR, DIPMETER ADVISOR, DRILLING ADVISOR, ELAS, HYDRO, LITHO, MUD. PROSPECTOR [14], for example, acts as a consultant to aid exploration geologists in their search for ore deposits. Given field data about a geological region, it estimates the likelihood of finding particular types of mineral deposits there. The system can assess the potential for

finding a variety of deposits, including massive sulfide, carbonate lead / zinc, porphyry copper, nickel sulfide, sandstone uranium, and porphyry molybdenum deposits. Its expertise is based on 1) geological rules which form models of ore deposits, and 2) a taxonomy of rocks and minerals. PROSPECTOR uses a combination rule-based and semantic net formalism to encode its knowledge and bases its inferences on the use of certainty factors and the propagation of probabilities associated with the data.

In Information Management, different expert systems have been reported such as CARGUIDE, GCA, CODES, EDAAS, FOLIO, IR-NLI, PROJCON, RABBIT, RESEDA. GCA [15], for example, helps graduate students plan their computer science curriculum. The system gathers information about a student's academic history and interests and then acts as a faculty adviser by suggesting a schedule of courses for the student. GCA's expertise includes departmental and university regulations regarding graduate degree programs, course descriptions, and sequences of courses frequently taken by computer science students. The knowledge in GCA is organized as four interacting sub-systems under the direction of a manager program. These sub-systems determine 1) the number of courses the student should take, 2) the courses the student is permitted to take, 3) the best courses to take, and 4) the best schedule for the student. GCA's knowledge is encoded as rules with assoiated certainty factors.

In Law, different expert systems have been reported such as LRS, AUDITOR, DSCAS, JUDITH, LDS, LEGAL ANALYSIS SYSTEM, LRS, SAL, SARA, TAXADVISOR, TAXMAN. LRS [16], for example, helps lawyers retrieve information about court decisions and legislation in the domain of negotiable instruments law, an area of commercial law that deals with checks and promissory notes. LRS contains subject descriptors that link each data item to the subject area concepts the item is about. A semantic net containing more than 200 legal concepts built up from six primitive concepts (party, legal instrument, liability, legal action, account, and amount of money), forms the basis for this knowledge. The knowledge in LRS provides it with the ability to make inferences about the meanings of queries and to extend user queries to include terms that are implied but not mentioned by the user.

In Manufacturing, different expert systems have been reported such as ISIS, IMACS, PTRANS. ISIS [17], for example, constructs factory job shop schedules. The system selects a sequence of operations needed to complete an order, determines start and end times, and assigns resources to each operation. It can also act as an intelligent assistant, using its expertise to help plant schedulers maintain shcedule consistency and indentify decisions that result in unsatisfied constraints. Knowledge in the system includes organizational goals such as due dates and costs, physical constraints such as limitations of particular machines, and causal constraints such as the order in which operations must be performed. ISIS uses a frame-based knowledge representation shceme together with rules for resolving conflicting constraints.

In Mathematics, different expert systems have been reported such as MACSYMA, MATHLAB 68, ADVISOR. MACSYMA [18], for example, performs symbolic manipulation of algebraic expressions and handles problems involving limit calculations, symbolic integration, solution of equations, canonical simplification, and pattern matching. The system uses mathematical expertise organized as individual knowledge sources and chosen for a particular problem by sophisticated pattern-matching routines. MACSYMA achieves very high quality and efficient performance on the mathematical problems within its scope.

In Medicine, different expert systems have been reported such as AI/COAG, AI/MM, AI/RHEUM, ABEL, ANGY, ANNA, ARAMIS, ATTENDING, BABY, BLUE BOX, CASNET/GLAUCOMA, CENTAUR, CLOT, DIAGNOSER, DIALYSIS THERAPY ADVISOR, DIGITALIS ADVISOR, DRUG INTERACTION CRITIC, EEG ANALYSIS SYSTEM, EMERGE, EXAMINER, GALEN, GUIDON, HDDSS, HEADMED, HEART IMAGE INTERPRETER, HEME, HT-ATTENDING, INTERNIST-I/CADUCEUS, IRIS, MDX, MECS-AI, MEDICO, MED1, MI, MODIS, MYCIN, NEOMYCIN, NEUREX, NEUROLOGIST-I, OCULAR HERPES MODEL, ONCOCIN, PATHFINDER, PARTEC, PEC, PIP, PUFF, RADEX, RX, SPE, SYSTEM D, THYROID MODEL, VM, WHEEZE. At per the generic categories of expert system applications, some selected expert systems in medicine are shown in fig.5.1. BABY [19], for example, aids clinicians by monitoring patients in a newborn intensive care unit (NICU). The system attempts to find clinically important patterns in the medical and demographic data about NICU patients. It monitors on-line data in the NICU, keeps track of the clinical states of the patients, suggests further evaluation for important findings and answers questions about the patients. BABY contains neonatology medical expertise for interpreting the clinical and demographic data. BABY is a forward chaining, rule-based system that uses rules embedded in a PROSPECTOR-like network. The system handles certainty by using a Bayesian probabilistic method similar to that used in PROSPECTOR. MYCIN [20] assists physicians in the selection of appropriate antimicrobial therapy for hospital patients with bacteremia, meningitis and cystitis infections. The system diagnoses the cause of the infection (e.g. the identity of the infecting organism is pseudomonas) using knowledge relating infecting organisms with patient history, symptoms and laboratory test results. The system recommends drug treatment (type and dosage) according to procedures followed by physicians experienced in infectious disease therapy. MYCIN is a rule-based system employing a backward chaining control scheme. It includes mechanisms for performing certainty calculations and providing explanations of the system's reasoning process.

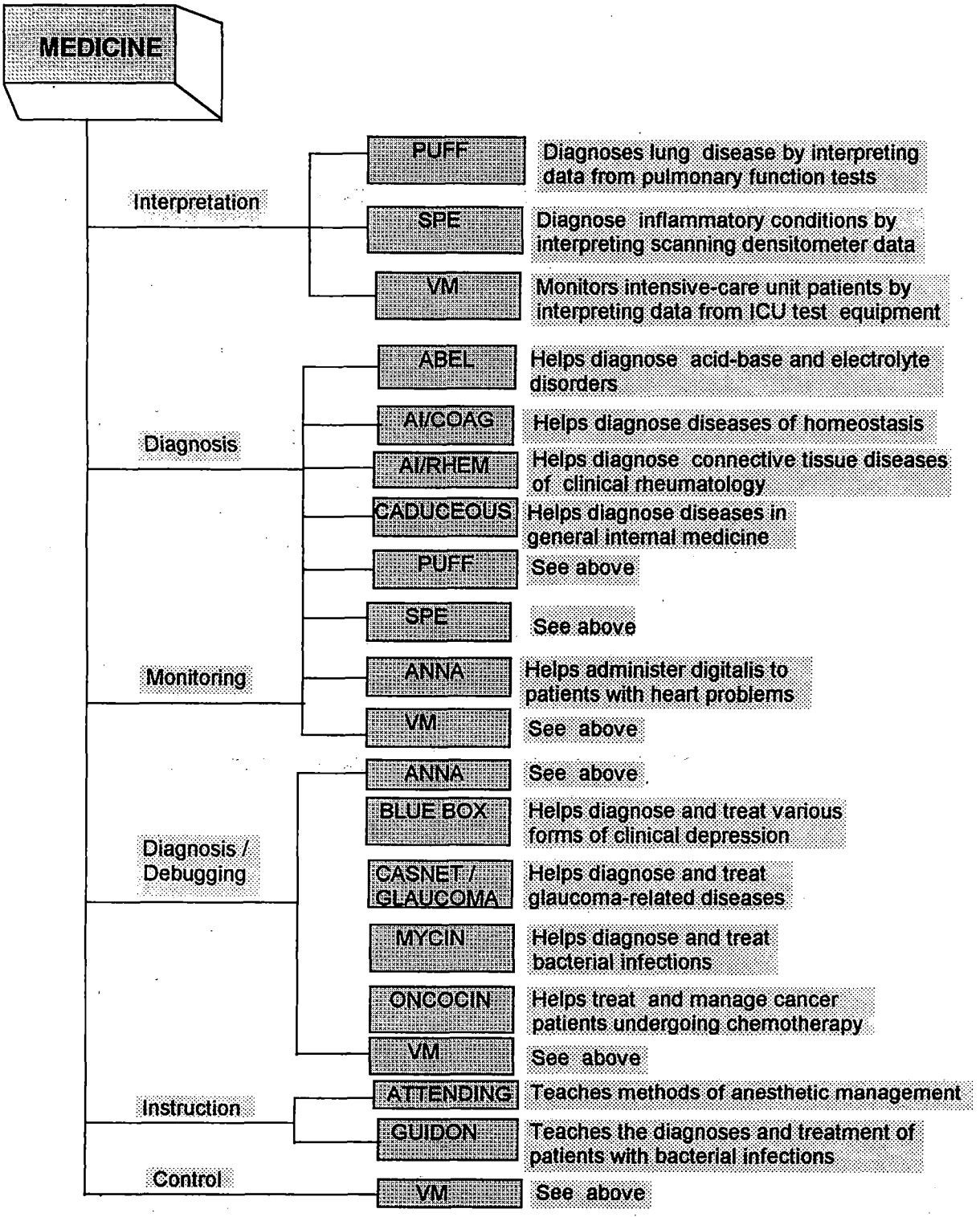


Fig. 5.1 Selected expert systems in medicine

In Meteorology, we may cite here WILLARD as an example of expert system. WILLARD [21] helps meteorologists forecast the likelihood of severe thunderstorms occurring in the central United States. The system queries a meteorologist about pertinent weather conditions for the forecast area and then produces a complete forecast with supporting justifications. The user may specify a particular geographical area for WILLARD to consider. The system characterizes the certainty of severe thunderstorm occurrence as "none", "approaching", "slight", "moderate", or "high", and each is given a numerical probability range. WILLARD's expertise is represented as rules generated automatically from examples of expert forecasting.

In Military Science, different expert systems have been reported such as MES, ACES, ADEPT, AIRID, AIRPLAN, AMUID, ANALYST, ASTA, ATR, BATTLE, DART, EPES, EXPERT NAVIGATOR, HANNIBAL, KNOBS, OCEAN SURVEILLANCE, RTC, RUBRIC, SCENARIO-AGENT, SIAP, SPAM, SWIRL, TATR and TWIRL. MES [22], for example, helps aircraft technicians diagnose aircraft problems. It is designed to overcome the shortage of technically qualified maintenance personnel by allowing less-qualified technicians to accurately assess problems with aircraft. MES contains knowledge taken from aircraft maintenance manuals, such as component weight and dimensions, ground operations, and troubleshooting and repair procedures. This knowledge is augmented by experiential knowledge from expert technicians. MES is a forward chaining, rule -based system.

In Physics, different expert systems have been reported such as MECHO and GAMMA. GAMMA [23], for example, helps nuclear physicists identify the composition of unknown substances by interpreting gamma-ray activation spectra produced when the substance is bombarded with neutrons. The system performs the identification by using knowledge about characteristic radiation energies and intensities emitted by different substances. Knowledge in the system is processed via the generate-and-test paradigm.

In Process Control, different expert systems have been reported such as FALCON and PDS. FALCON [24], for example, identifies probable causes of process disturbances in a chemical process plant by interpreting data consisting of numerical values from gauges and the status of alarms and switches. The system interprets the data by using knowledge of the effects induced by a fault in a given component and how disturbances in the input of a component will lead to disturbances in the output. Knowledge is represented in two ways - as a set of rules controlled by forward chaining and as a causal model in network form.

In Space Technology, different expert systems have been reported such as ECESIS, FAITH, KNEECAP, LES, NAVEX, RBMS and RPMS. ECESIS [25], for example, provides autonomous control of an environmental control / life support sub-system (EC/LSS) for use aboard a manned space station. The system decides how to shift the modes of the various EC/LSS sub-systems during the transition from shadow to sun. It also monitors

the EC/LSS, triggering actions in response to various events. Although ECESIS is intended to operate autonomously, it has a simple explanation capability to facilitate system demonstration. ECESIS has a bybrid architecture involving both rule-based and semantic net formalisms, and it uses the Bayesian scoring model developed for PROSPECTOR to handle uncertainty.

In Business, we may cite here SUTA as an example of expert system. SUTA [26], an expert system called Soviet Union Trade Advisor (SUTA) was developed by Deloitte and Touche, a large management consulting (and CPA) company. The major objective of the system is to provide advice on trade opportunities and licensing requirements for medium to high-technology products. The system is very user-friendly; it is based on simple sets of menus. The market is divided into twelve sectors with which potential products are matched. The system assesses the opportunities for general classes of products and then for specific ones. Then, potential buyers are identified together with procedures for making contacts. Explanations are provided on request. Several other types of valuable information are provided by the system.

In Power systems, we may cite here ENERGY MANAGEMENT as an instance of expert system. ENERGY MANAGEMENT [27] provides the utility needed a real-time ES integrated with a signaling system as well as with existing databases in hydroelectric plant. Links were done via communication lines. The system, which was developed on a Texas Instruments workstation (explorer), began as an initial prototype with 400 rules, 1500 objects, and 200 LISP functions. It was developed with LISP-related language and fully implemented after two years (in 1989). The system uses a model base in addition to the rule base. System capabilities include (1) detecting a disturbance when it happens, (2) locating the element of the fault, (3) tracking the location of the disturbance, and (4) recommending repairs (and justifying them).

In Transportation, we may cite here CARGEX as an example of expert system. CARGEX [28] is an expert system constructed to assist in making loading decisions. The German airline Lufthansa concentrates on airfreight consolidation process. Worldwide, all airfreight applications are routed into the cargo centre in Frankfurt. A "traditional" electronic data processing system makes automatic loading decisions, as long as (1) freight is standard and (2) some threshold values on loading an aircraft are not exceeded. Applications that cannot be handled by this system are routed on the screen to experts who decide (a) whether an application is acceptable, and if it is, at what price, and (b) on which specific flights the airfreight should be loaded. CARGEX actually supports the segments of the interface from Frankfurt, Atlanta, Dallas / Fort Worth, Houston, and Mexico. The system contains approximately 300 rules that are described in about 6,000 lines of code. The basic goal of the application of CARGEX is increased productivity of the consolidation system; it is expressed in two objectives :

1. Maximize the amount of kilogram chargeable per aircraft. This measure is computed as the sum of the actual weights of the goods and the virtual weights of voluminous goods.

2. Handle an increasing amount of airfreight business with a substantially smaller increase of consolidation personnel.

In Mining, we may cite here DustPro as an example of expert system. DustPro [29] is a small rule-based system developed by the U.S. Bureau of Mines. It includes about 200 rules and was developed with a Level5 shell on a micro-computer. It took 500 hours to develop the system. The system is now in operation in more than 200 mines. It is also successful that more than ten countries have requested permission to use the system in their mines. DustPro advises in three areas : control of methane gas emission, ventilation in continuous operations, and dust control for the mine's machines. The system is completely independent. Data on air quality is entered manually. The user interface is very friendly. The system is composed of thirteen subareas of expertise, and the average consultation time is ten to fifteen minutes.

In Communications, we may cite here COMPASS as an example of expert system. COMPASS [30] (Central Office Maintenance Printout Analysis and Suggestion System) analyzes maintenance printouts of telephone company control switching equipment and suggests maintenance actions to be performed. A central office telephone switch system connects thousands or tens of thousands of telephone lines to one another or external trunks (interconnections among central offices). Such a switch system can produce hundreds of maintenance messages daily. The massaages are analyzed to determine what actions are necessary to maintain the switch. COMPASS affords several potential advantages to users :

- It supports better telephone switch performance, yielding higher-quality service to telephone customers.
- It increases the productivity of experienced switch maintenance persons.
- It upgrades the performance of less experienced personnel.
- It provides guaranteed maintenance of existing switches by capturing expertise that may not be available in the future.

In Education, we may cite here MIKE as an example of expert system. MIKE [31], developed by the Mandell Institute, is the automated admissions representative of Brandeis University. Built to expand the pool of high-quality applicants for the school, the system is designed to be used by high school seniors who are narrowing their choice of colleges. MIKE explains all the academic and extracurricular programs in which the prospective applicant has an interest. It also uses its video base to take the student

on a tour of the campus. Once the system has the student's interest, it even gives feedback concerning the likelihood of admission and the type and amount of financial aid. At this point, the student's name and address are taken for transmission to the campus-based computer for follow-up.

In Environment, we may cite here DustPro, an expert system for controlling environment in mines. This has already been discussed in the mining application.

In Image processing, we may cite here 3DPO [32] as an example of expert system. Image understanding is the process of establishing an interpretation of a scene based on one or more images of that particular scene. Traditionally, an image understanding algorithm is sub-dived into three steps, i.e. segmentation feature extraction or object description, and classification or interpretation. 3DPO stands for a 3-Dimensional Part Orientation system. The goal of this system is to recognize and locate identical castings jumbled together in a bin. In the 3DPO system, one of the important ideas is to extract as much knowledge as possible about the models, in order to simplify and speed up the interpretation process. Actually, knowledge, such as model information and heuristics, semantics, a priori information and / or expectations about the scene, plays an important role in image understanding. As a result, there is a justified trend in image understanding to pay closer attention to all kind of knowledge and to focus on it in the analysis of complex scenes. Those analysis systems are commonly called knowledge-based vision systems.

In Science, different expert systems have been reported such as DENDRAL, MOLGEN. DENDRAL [10], for example, provides a rule-based program for identifying chemical compounds from laboratory data and performs this task better than chemists because it tirelessly considers all possible candidates - even those that a human expert might initially rule out as unlikely. Developed by Stanford University in the late 60's, DENDRAL is now widely used by industrial and academic researchers. Another Stanford program (MOLGEN) plans experiments for determining the coding sequences of DNA molecules. Its knowledge base encodes the DNA synthesis and analysis procedures of some of the world's leading genetic engineers.

In Aerospace, we may cite here REX as an example of expert system shell. REX [33] an object-oriented, asynchronous real-time expert system shell to meet the challenges of the dynamic aerospace environment. The system's distinguishing feature is a semantically rich temporal representation facility for both data and knowledge. It handles the matching and firing of two rule types for encoding temporal knowledge : clock synchronized rules and spanning rules. To handle simultaneous, disjointed events, they have adopted a multiple-rule-firing model. Also in REX, rules are objects - and they are stored, retrieved, and managed exactly like data objects - unlike conventional object-oriented expert system shells where rules operate on top of objects and are treated

differently from objects. A knowledge-based system developed in REX can consist of multiple expert systems or agents.

Fig. 5.2 shows the major application areas that naturally developed - the breadth of the applications is remarkable. The fig. 5.2 also shows the number of developed systems for each area [8].

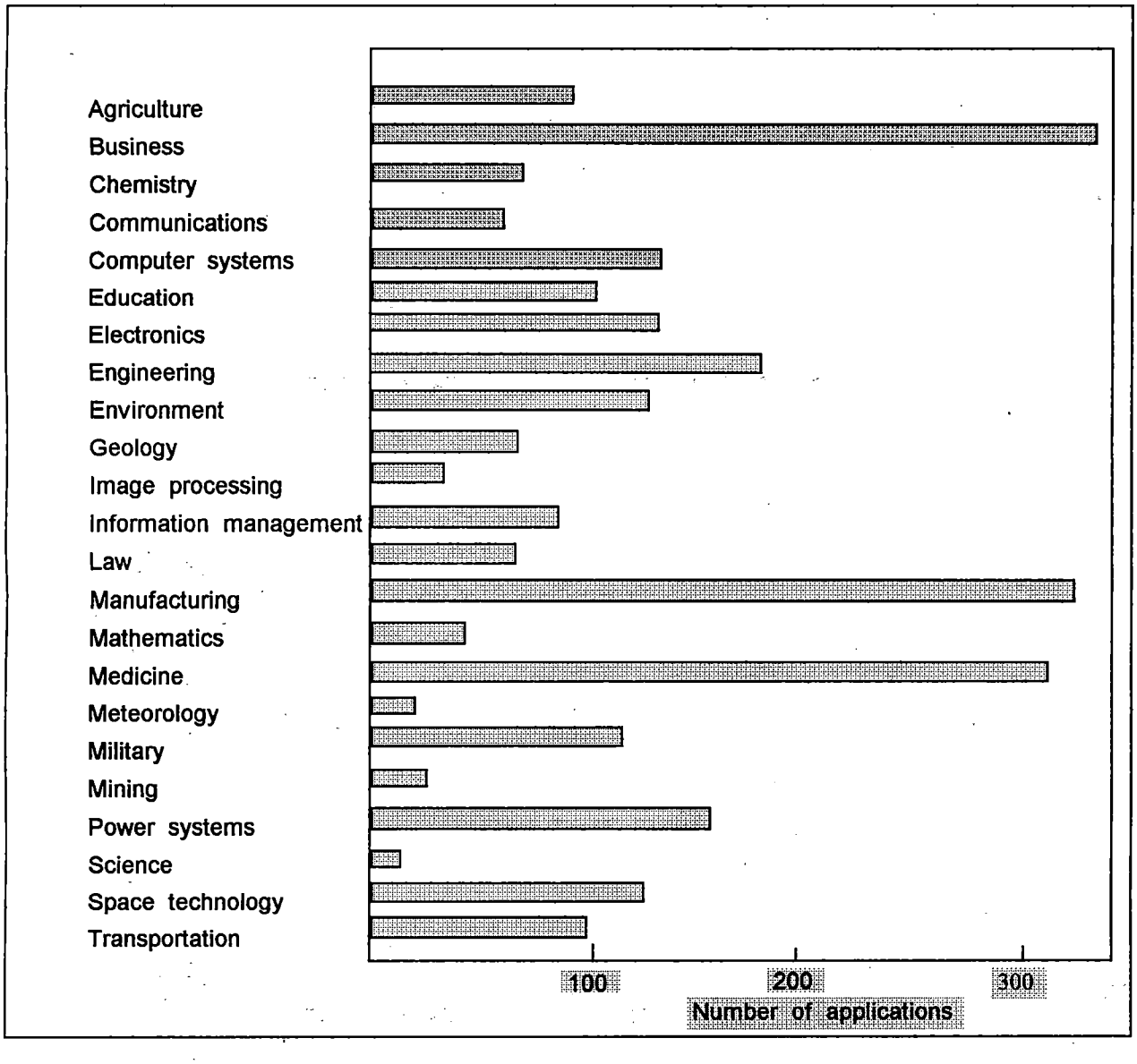


Fig. 5.2 The number of developed expert systems in various application areas

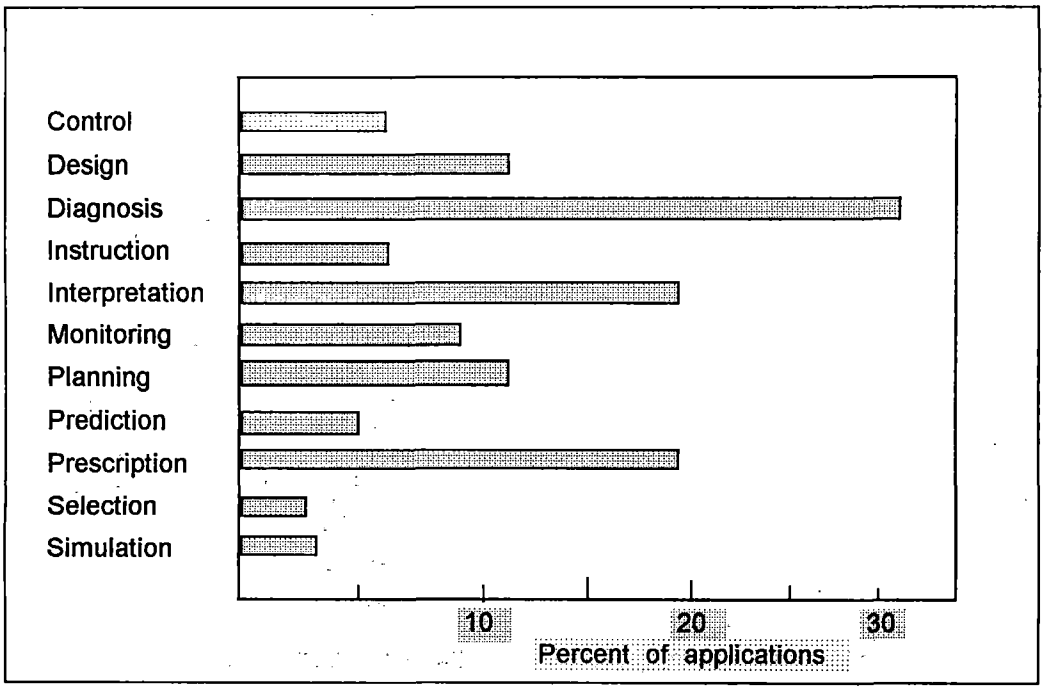


Fig. 5.3 The percentage of expert system applications by category

Fig. 5.3 shows the percentage of applications for each problem type in Table 5.1. Many applications employ more than one activity. For example, a diagnostic system might first interpret the available data, and later prescribe a remedy for the recognized fault.

As fig. 5.3 illustrates, the predominant role of expert systems has been diagnosis. One reason for the result is that this is the role most experts play. Fields such as medicine, engineering, and manufacturing have many individuals who help diagnose problems. Another reason for the large percentage of diagnostic systems is their relative ease of development. Most diagnostic problems have a finite list of possible solutions and a limited amount of information needed to reach a solution. These bounds provide an environment that is conducive to effective system design.

The large percentage can also be traced to the practical considerations of introducing a new technology into an organization. Most organizations prefer to take a low-risk position when considering a new technology. So, they prefer projects that require minimum resources and have the maximum likelihood of success. Because diagnostic systems are relatively easy to build, they are attractive to firms venturing into the field.

The drop-off from the large number of diagnostic applications to that of some other problem types is dramatic. Two reasons help explain this result. First, tasks such as design and planning are difficult to implement in an expert system framework because

their steps vary greatly between application areas and it is often hard to precisely define these steps. Second, tasks such as instruction, control, and simulation, although they are excellent areas for expert system applications, are relatively new ventures [8].

5.2.2 **Application trends**

Fig. 5.2 showed that expert systems have primarily been used in business, manufacturing, and medicine. Fig. 5.4 shows the number of developed expert systems per year in these areas. However, one may find in fig. 5.5 the dramatic increase in the number of developed expert systems during recent years.

In the early eighties, medical expert system applications dominated the scene. This is primarily due to the diagnostic nature of these applications and the relative ease of developing such systems. However, as we moved toward the mid-eighties, more difficult problems were approached. It was also time to develop systems that benefited the commercial sectors. Unfortunately, initial attempts frequently met with limited success. Three primary reasons help explain this result.

First, early applications of expert systems in industry often over-challenged the technology, leading to poor results. Many designers tried to build systems to solve problems that were beyond even the best experts. The thinking was "Well, we can't solve this problem, so let's try throwing AI at it". Second, other designers often took on a project whose scope was so broad that completing it in a reasonable time frame was impossible. Third, some designers developed remarkably intelligent systems; but failed to meet up the client's need to integrate the system into existing hardware and software. As a result, the powerful finished products were left on the shelves to collect dust.

With the few successes being produced during this period, coupled with earlier glowing promises of the technology, critics crept out of the bushes and quickly pounced on the situation. Journal and conference papers, newsletters, and the national media were swift to point to the shortcomings. For example, Forbes [34] asked, "What happened to those 'expert' systems that were supposed to transform the world of business forever"? Expert system designers began to realize that finding a place for the technology can be as tedious as matching the glass slipper to Cinderella's foot [8].

The turning point came in the mid-eighties when designers began to focus on very narrow, well-defined, and sometimes even mundane tasks. They also took the time to look at where the technology would be embedded. Although the systems developed from this focus might have seemed unimpressive to the AI researcher viewing the scene from the ivory tower, they were well received by managers in industry because they produced commercially worthwhile results.

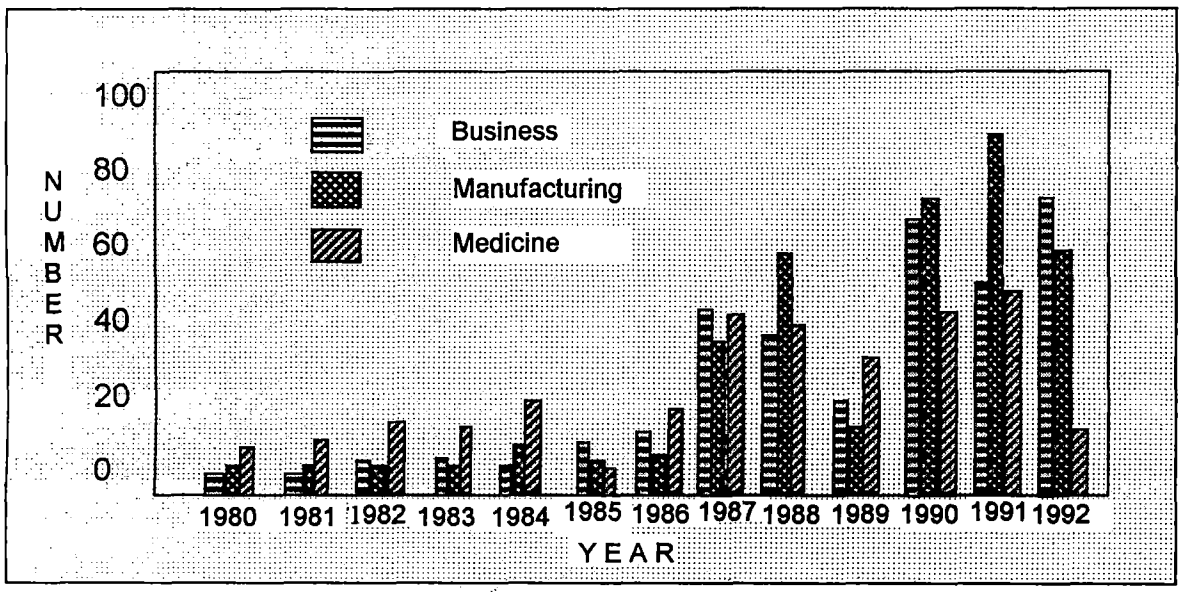


Fig. 5.4 The number of expert systems developed per year for business, manufacturing and medicine

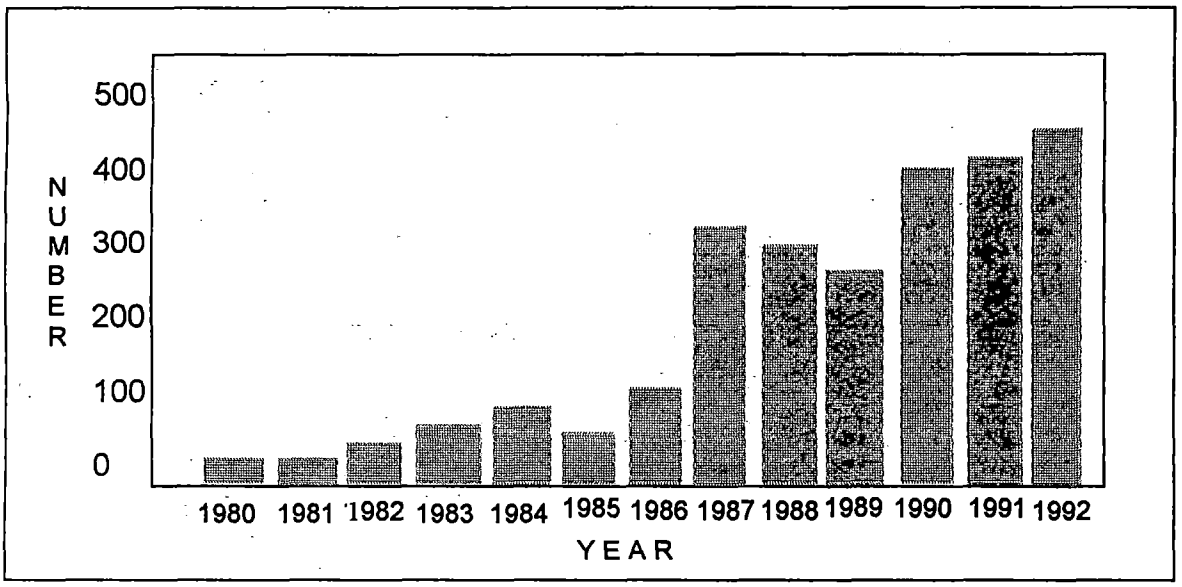


Fig. 5.5 The number of expert systems developed from 1980 - 1992

One can observe a dramatic swing toward commercially viable systems, and applications for business and manufacturing began to pick up steam. Medical applications continued to grow, but not at a similar rate.

Durkin [8] divided the application areas into two categories : commercial and scientific. The first category includes systems that produce economically beneficial products for organizations in business, manufacturing, power systems, and transportation. The second category includes systems that produce primarily scientific results for chemistry, geology, image processing, and space technology. Fig.5.6 shows the ratio of systems developed in the commercial category to those developed in the scientific category per year.

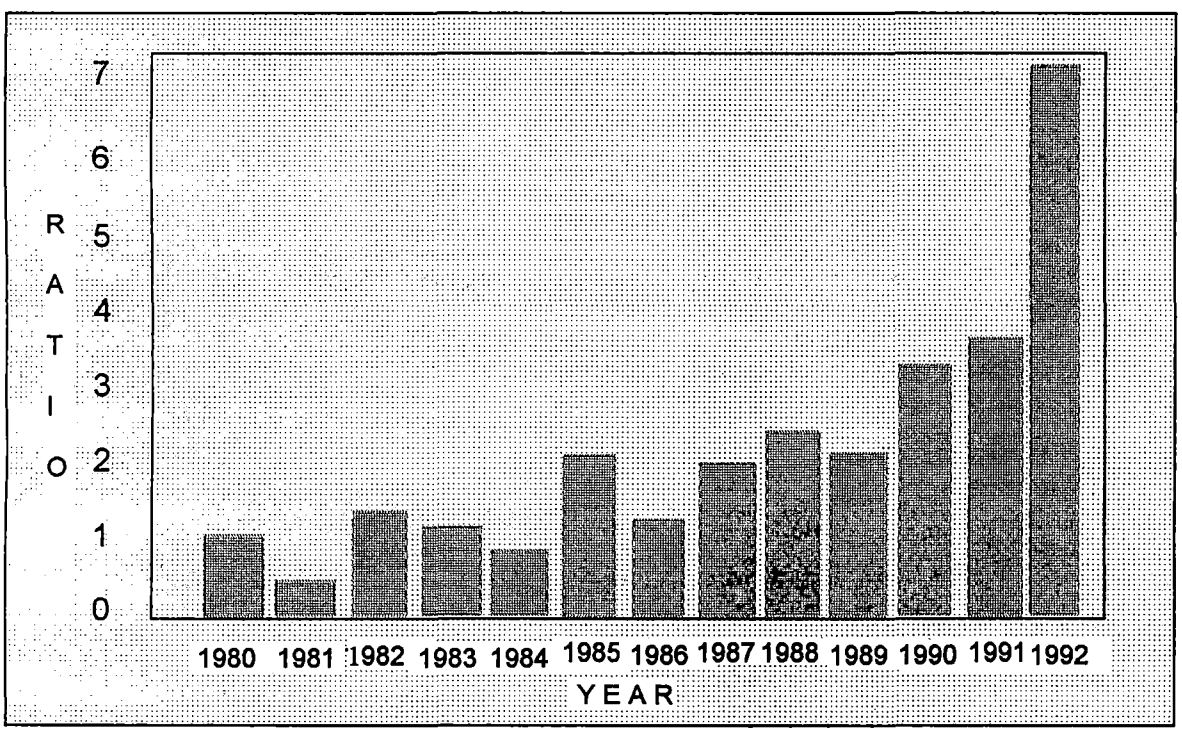


Fig. 5.6 The ratio per year of developed commercial expert systems to developed scientific ES

In the early eighties this ratio remained around one-to-one; that is, as many commercial as scientific applications. The mid-to-later eighties showed a two-to-one ratio, while the early nineties showed a dramatic increase toward commercial applications.

During the seventies AI was a cult activity-almost a religion. Researchers centered on producing intelligent general-purpose reasoning machines. The fascination of achieving this academic challenge drove their efforts. By the eighties, when the fuel for the advancement of the technology came from sectors that demanded a return on their investment, researchers began to realize that this is not a religious experience but an economic one. The trend in fig. 5.6 - from laboratory to industrial applications - is one measure we can use to judge the technology's value. But, however, we feel that there is the need of exploring more domains suitable for expert systems applications on the one hand and then developing practical systems on the other.

5.2.3. **Components of an expert system**

Though at present there is no such thing as a standard expert system, but most expert systems have a knowledge base and inference engine and a user interface. AI environments for expert system development are shown in block diagrams (fig.5.7 and fig.5.8) which are more or less self explanatory. Previously AI scientists thought that by emulating the process of human reasoning computer could solve problems without having access to large amounts of specific knowledge which ultimately proved to be unsuccessful. In mid 70s AI scientists realised that the problem solving power of a programme comes from the knowledge it possesses, not just from the formalisms and inference schemes it employs. The current concept is to make a programme intelligent provide it with lots of high quality specific knowledge about some problem area. This realisation led to the development of systems that were very successful in some narrow problem area. The component of the expert system that contains collection of the domain knowledge for the system is called its knowledge base. This element of the system is so critical to the way most expert systems are constructed that they are also popularly known as knowledge based system. The **knowledge base** of an expert system contains both declarative (facts about objects, events and situations) and procedural (information about courses of action) knowledge depending on the form of knowledge representation chosen that two types of knowledge may be separate or integrated. There are several knowledge representation schemes such as First-order-predicate calculas, Semantic Networks, Frames and Production Rules etc. which will be discussed in **chapter 6**.

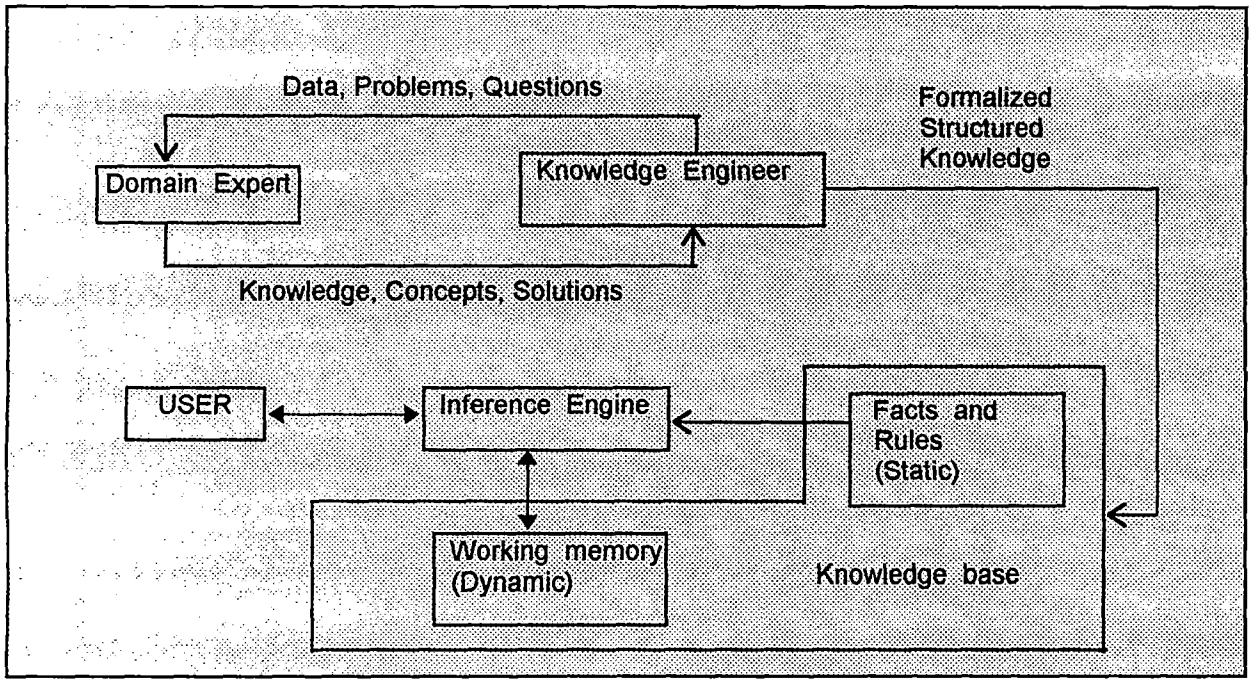


Fig. 5.7 An architecture of a typical expert system

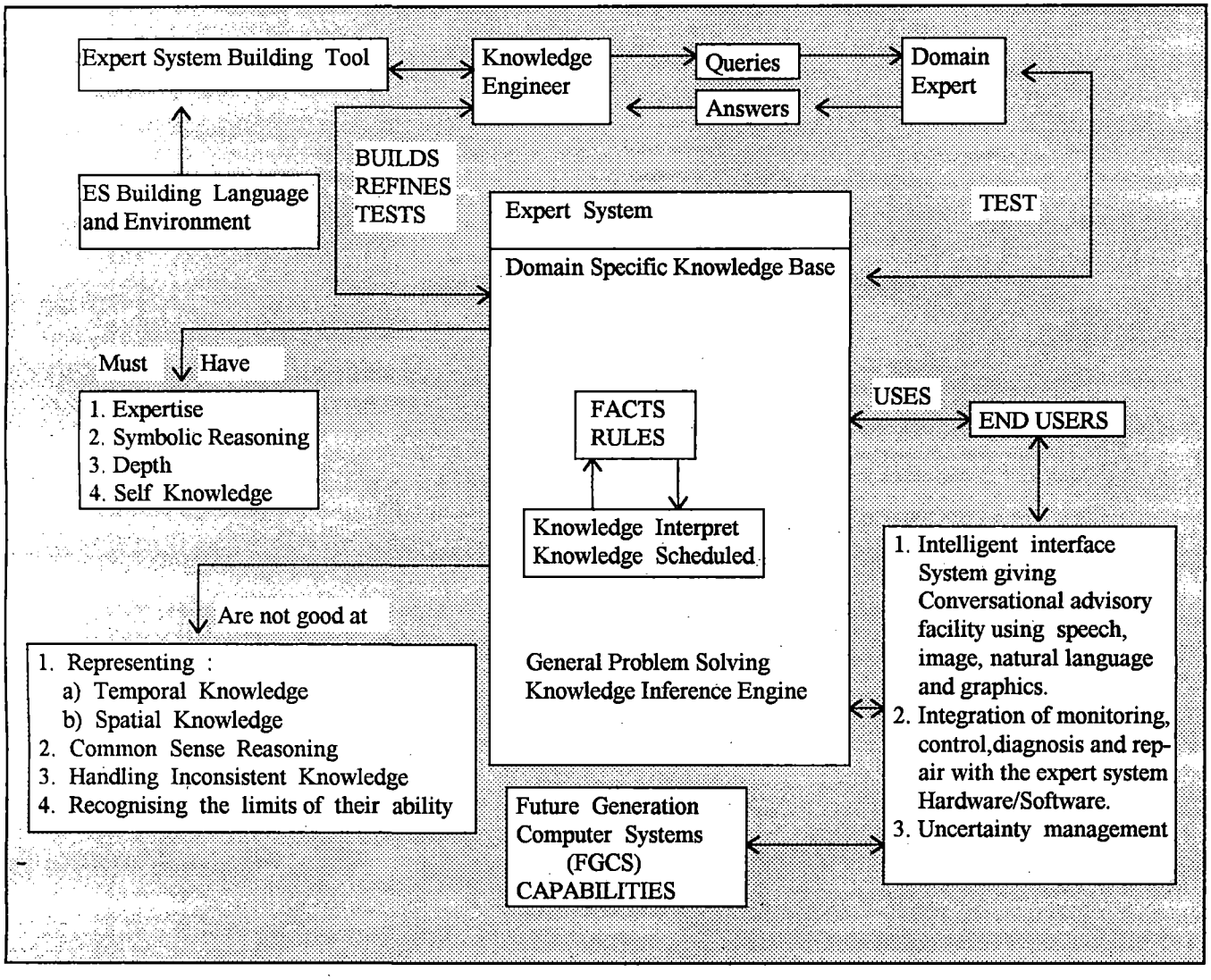


Fig. 5.8 A block diagram indicating ES-development environment

Simply having lot of knowledge does not make one an expert. The system must know how and when to apply the appropriate knowledge. So having a knowledge base itself does not make an expert system intelligent. The component that is responsible to direct the implementation of the knowledge is known variously as the control structure, the rule interpreter, or the inference engine. The **inference engine defines** which heuristic search techniques are used to determine how the rules in the knowledge base are to be applied to the problem. As a matter of fact the inference engine runs an expert system determining which rule are to be invoked accessing the appropriate rules in the knowledge base executing the rules and determining when an acceptable solution has been found. The knowledge in an expert system is not interwined with the control

structure. As a result of which an inference engine that works well in one expert system may work just as well with a different knowledge base. For example, the inference engine of one of the most famous medical expert system MYCIN is available separately as EMYCIN (essential MYCIN). EMYCIN can be used with a different knowledge base to create a new knowledge system eleminating need to develop a new inference engine.

Next important component is the **user interface** that enables user to communicate with an expert system. The communication performed by a user interface is bi-directional. At the simplest level the user must be able to describe his problem to the expert system and the system must be able to respond with its recommendations. The user may also ask the system to explain its reasoning or the user may ask the system for additional information about the problem. The system may also ask the user for additional information about the problem. In fig. 5.8 different features that are desirable for the end-user interface is described. As a matter of fact the capabilities of using speech, natural language, pictures and graphics are the most important features of the fifth generation systems also.

5.2.4. **Typical features of an expert system**

Table 5.1 indicated the functional categories for expert system applications in a tabular form. Inspite of the fact that each expert system is unique in some sense, certain features are desirable for any expert system. Some authors suggested 7 criteria as pre-requisites for acceptance of an expert system by its intended users. These are :

(a) The program should be useful which means the expert system should be developed to meet a specified need for which it is recognised that assistance is needed;

(b) The programme should be usable which means it should be designed so that a less experienced computer user can use it;

(c) The programme should be educational and appropriate which means a non-expert can use the system and increase his own expertise by using the system;

(d) The programme should be able to explain its advice, which means the reasoning process of the system should be transparent so that the user is able to decide whether to accept the system's recommendations;

(e) The programme should be able to learn new knowledge, which means the system should ask questions to the user to gain additional information and incorporate it to the system, if necessary;

(f) The programme's knowledge should be easily modifiable so that the knowledge base of an expert system can be revised easily to correct errors or to add new information.

5.2.5. **Major stages of expert system development**

In fig.5.8 an attempt has been made to present an expert system development environment in its totality in one diagram that is self explanatory [35]. From this figure it is clear that knowledge engineers and domain experts, the two main categories of people that are responsible and must work together to design an expert system. The knowledge engineers develop the expert system and domain experts develop the information for the knowledge base. A knowledge engineer is an AI specialist who can be from any of the conventional disciplines. A domain expert is an individual who has significant expertise in the domain of the expert system being developed. Domain expert may not understand AI at all.

Expert system development can be viewed as five highly interdependent and overlapping phases : identification, conceptualization, formalization, implementation and testing [7]. Fig. 5.9 illustrates the stages of an expert system development.

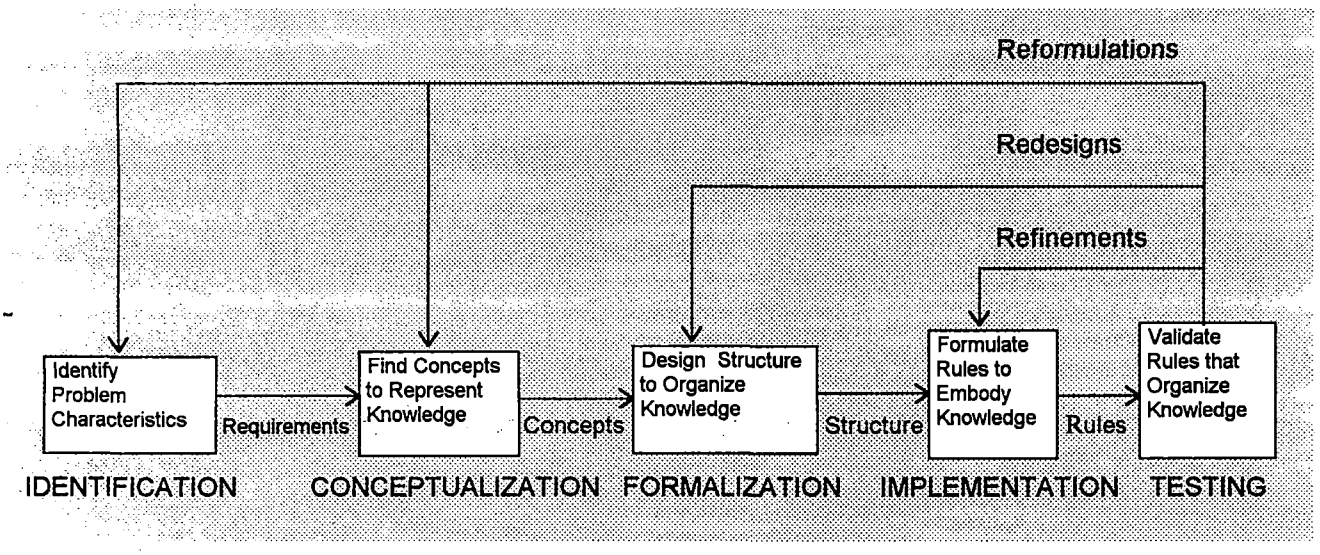


Fig. 5. 9 Stages of expert system development.

- **Identification**

The knowledge engineer and expert determine the important features of the problem. This includes identifying the problem itself (e.g. type and scope), the participants in the development process (e.g. additional experts), the required resources (e.g. time and computing facilities), and the goals or objectives of building the expert system (e.g. improve performance or distribute scarce expertise). Of these activities, identifying the problem and its scope gives developers the most trouble. Often the problem first

considered is too large or complex and must be scaled down to a manageable size. The knowledge engineer may obtain a quick measure of this complexity by focusing on a small but interesting sub-problem and implementing routines to solve it.

- **Conceptualization**

The knowledge engineer and expert decide what concepts, relations and control mechanisms are needed to describe problem solving in the domain. Subtasks, strategies and constraints related to the problem-solving activity are also explored. At this time the issue of granularity is usually addressed. This just means considering at what level of detail the knowledge should be represented. The knowledge engineer will normally pick the most abstract level of detail (coarsest grain) that still provides adequate discrimination between key concepts. A word of warning - the developers must avoid trying to produce a complete problem analysis before begining program implementation. They will learn much from the first implementation that will shape and direct the conceptualization process.

- **Formalization**

Formalization involves expressing the key concepts and relations in some formal way, usually within a framework suggested by an expert system building language. Thus the knowledge engineer should have some ideas about appropriate tools for the problem by the time formalization begins. For example, if the problem seems amenable to a rule based approach, the knowledge engineer might select ROSIE as the system building language and gather expertise in the form of IF-THEN rules. If a frame-based approach seems more appropriate, the knowledge engineer might instead select SRL and work with the expert to express domain knowledge as a large network.

- **Implementation**

The knowledge engineer turns the formalized knowledge into a working computer program. Constructing a program requires content, form and integration. The content comes from the domain knowledge made explicit during formalization, that is, the data structures, inference rules and control strategies necessary for problem solving. The form is specified by the language chosen for system development. Integration involves combining and reorganizing various pieces of knowledge to eliminate global mismatches between data structures and rule or control specifications. Implementation should proceed rapidly because one of the reasons for implementing the initial prototype is to check the effectiveness of the design decisions made during the earlier phases of development. This means that there is a high probability that the initial code will be revised or discarded during development.

- **Testing**

Testing involves evaluating the performance and utility of the prototype program and revising it as necessary. The domain expert typically evaluates the prototype and helps the knowledge engineer to revise it. As soon as the prototype runs on a few examples, it should be tested on many problems to evaluate its performance and utility. This evaluation may uncover problems with the representational scheme, such as missing concepts and relations, knowledge represented at the wrong level of detail, or unwieldy control mechanisms. Such problems may force the developers to recycle through the various development phases, reformulating the concepts, refining the inference rules and revising the control flow.

5.2.6. **Types of expert systems**

5.2.6.1. **Based on reasoning**

Rule-based reasoning

A type of knowledge representation in which the knowledge about a domain is expressed in rules that define relationships between facts. Rules provide a formal way of representing recommendations, directives or strategies. They are often appropriate when the domain knowledge results from empirical associations developed through years of experience solving problems in an area.

Case-based reasoning

A knowledge base for case-based reasoning [36] is a set of relevant examples rather than general rules. These cases are applied to new problems by an analogical reasoning process. This is another response to the complexities encountered in trying to handcraft a knowledge base of general rules that will cover all situations. Proponents of case-based reasoning argue that this is closer to human reasoning. Case-based approaches have played an important role in expert programs in law and medicine.

Table 5.3

Comparison of Case-based and Rule-based reasoning [37]

Criterion	Rule-based reasoning	Case-based reasoning
Knowledge unit	Rule	Case
Granularity	Fine	Coarse
Knowledge acquisition units	Rules, hierarchies	Cases, hierarchies
Explanation mechanism	Backtrace of rule firings	Precedent cases
Characteristic output	Answer, plus confidence measure	Answer, plus precedent cases
Knowledge transfer across problems	High, if backtracking Low, if deterministic	Low
Speed as a function of knowledge base size	Exponential, if backtracking; Linear,if deterministic	Logarithmic, if index tree balanced
Domain requirements	Domain vocabulary Good set of inference rules Either few rules or Rules apply sequentially Domain mostly obeys rules	Domain vocabulary Database of example cases Stability - a modified good solution is probably still good Many exception to rules
Advantages	Flexible use of knowledge Potentially optimal answers	Rapid response Rapid knowledge acquisition Explanation by examples
Dis-advantages	Computationally expensive Long development time Black-box answers	Suboptimal solutions Redundant knowledge base

Frame-based reasoning

Reasoning with frames is much more complicated than reasoning with rules. The slot provides a mechanism for a kind of reasoning called expectation-driven processing. Empty slots (i.e. uncofirmed expectations) can be filled, subject to certain conditioning, with data that confirm the expectations. Thus, frame-based reasoning looks for confirmation of expectations and often just involves filling in slot values.

Perhaps the simplest way to specify slot values is by default. The default value is attached loosely to the slot so as to be easily displayed by a value that meets the

assignment condition. In the absence of information, however, the default value remains attached and expressed.

The reasoning process that takes place with frames is essentially the seeking of confirmation of various expectations. This amounts to filling in the slots and verifying that they match the current situation. With frames, it is easy to make inferences about new objects, events, or situations because the frames provide a base of knowledge drawn from previous experience.

The reasoning in frames can be executed in different ways. Two most common ways are using rules and employing hierarchial reasoning.

Advantages and **dis-advantages of frame-based reasoning**

In frame-based system, it is always easy to see the order and relationship of the elements. Frame-based systems assume that the hierarchical relationship of the objects is relatively static. If the order is dynamic, using a frame-based system becomes difficult.

Frame-based systems are most applicable to biological classification systems, and similar types of systems, in which a static hierarchical classification is a part of the knowledge.

Model-based reasoning

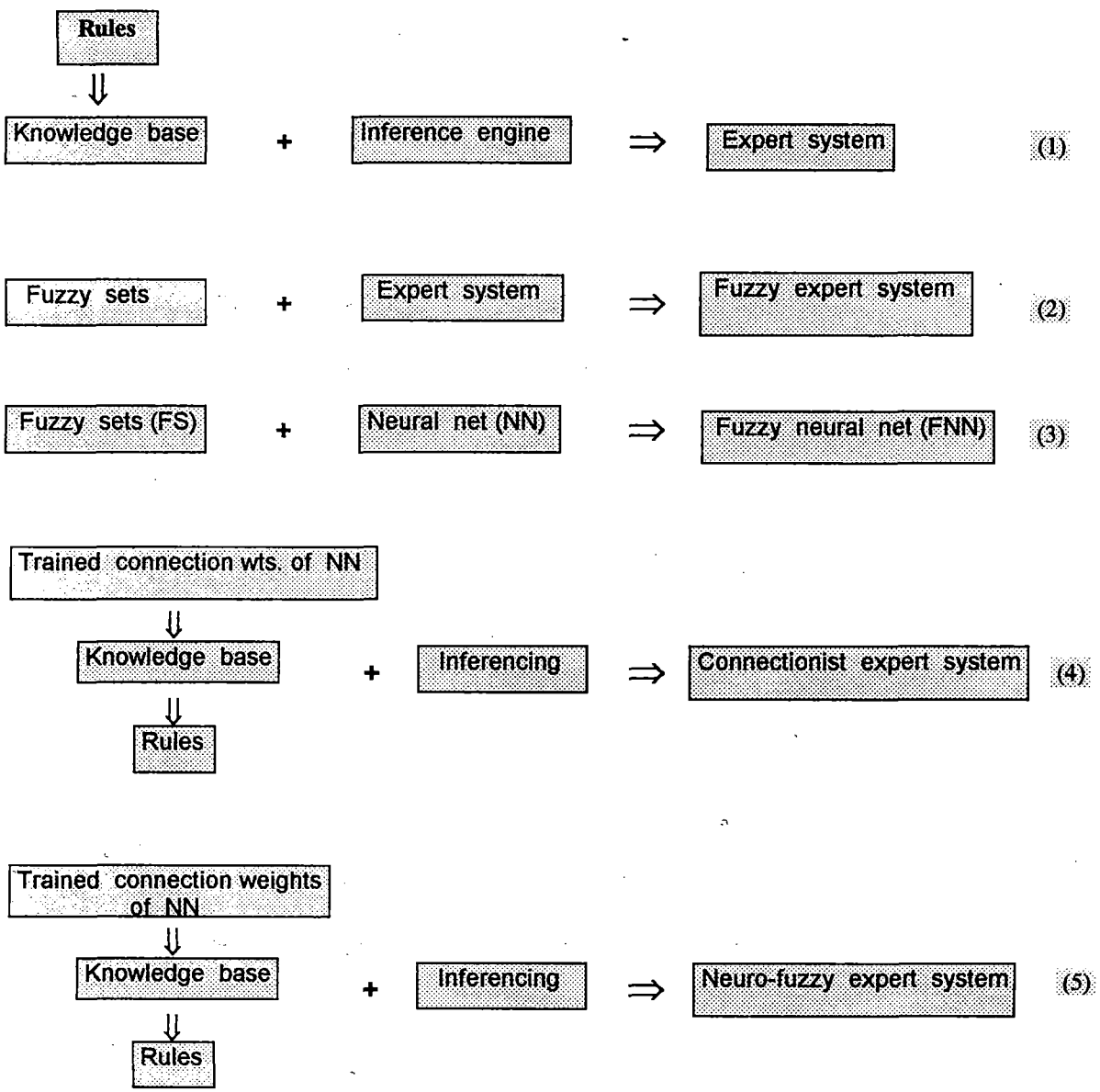
Model-based reasoning is based on knowledge of the structure and behavior of the devices the system is designed to understand. Model-based systems are especially useful in diagnosing equipment problems. The systems include a model of the device to be diagnosed that is then used to identify the cause(s) of the equipment's failure. Because they draw conclusions directly from knowledge of a device's structure and behavior, model-based expert systems are said to reason from "first principles".

Rule-based versus model-based expert system

Most of the well known expert systems, such as MYCIN, DENDRAL etc. are rule based expert system. But another promising approach to expert system design is to use model based approach instead of rule based one. A model based expert system is specially useful in diagonosing equipment problems. The hardware trouble shooting group in MIT's AI laboratory explored the use of model-based approach to diagnose malfunctioning to computers. Unlike rule-based expert systems which are based on human expertise, the model-based ones are based on knowledge of the structure and behaviour of the devices they are designed to understand.

5.2.6.2. **Based on other technological advancement**

It is worth mentioning two major problems in building expert systems : (i) constructing and debugging knowledge base, and (ii) management of uncertainties. In practice, the knowledge base construction can be said to be the only real task in building an expert system considering the proliferating presence of expert shells having their own uncertainty management schemes. In the recent years, different ideas, concepts, methodologies have been introduced in circumventing the above and allied problems in building knowledge-based expert systems and / or in improving the performance in decision making systems. The resulting basic modules of various expert systems [38] are shown in fig. 5.10.



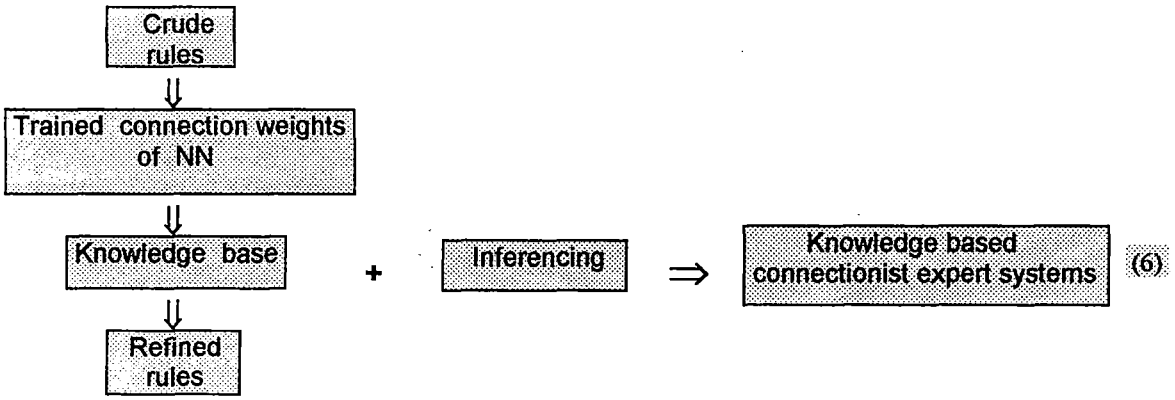


Fig. 5.10 Block diagram of the basic modules of various expert systems

Artificial neural networks [39-42] can be formally defined as massively parallel interconnections of processing elements that interact with objects of the real world in a manner similar to biological systems. All information is stored distributed among the various connection weights. The networks can be trained by examples and sometimes they generalize well for unknown test cases.

Fuzzy logic is based on the theory of fuzzy sets and, unlike classical logic, it aims at modeling the imprecise (or inexact) modes of reasoning and thought processes (with lingistic variables) that play an essential role in the remarkable human ability to make rational decisions in an environment of uncertainty and imprecision. This ability depends, in turn, on our ability to infer an approximate answers to a question based on a store of knowledge that is inexact, incomplete, or not totally reliable.

We see that fuzzy set theoretic models [43, 44] try to mimic human reasoning and the capability of handling uncertainty, whereas the neural network models attempt to emulate the architecture and information representation schemes of the human brain. Integration of the merits of fuzzy set theory and neural network theory therefore promises to provide more intelligent systems (in terms of parallelism, fault tolerance, adaptivity and uncertainty management) to handle real life recognition / decision making problems. For the last five to seven years, there have been several attempts [45-48] by researchers over the world in making a fusion of the merits of these theories under the heading 'neuro-fuzzy computing' for improving the performance in decision making systems.

As the knowledge base of an expert system is a repository of human knowledge and since some of these may be imprecise in nature, often, this may result in a collection of rules and facts which for the most part are neither totally certain nor totally consistent. The expert system is also likely to be required to infer from premises that are imprecise, incomplete or not totally reliable. The uncertainty of information in the knowledge base

of the question-answering system thus induces some uncertainty in the validity of its conclusions [49]. Hence a basic problem in the design of expert systems is the analysis of the transmitted uncertainty from the premises to the conclusion and the association of a certainty factor [50]. Fuzzy expert systems [50, 51], incorporating the concept of fuzzy sets at various stages, help to a reasonable extent in the management of uncertainty in such situations.

Neural networks are also used in designing expert systems. Such models are called connectionist expert systems [52], and they use the set of connection weights of a trained neural net for encoding the knowledge base for the problem under consideration. The use of Artificial neural networks helps in (A) incorporating parallelism, and (B) tackling optimisation problems in the knowledge base space. These models are usually suitable in data-rich environment and seem to be capable of overcoming the problem of the knowledge acquisition bottleneck of traditional expert systems. They help in minimizing human interaction and associated inherent bias during the phase of knowledge base formation (which is time-consuming in the case of traditional models) and also reduce the possibility of generating contradictory rules. Powerful learning techniques exist for generating connectionist networks from training samples. This enables us to automate the construction of knowledge bases for classification-type expert systems. When the connection weights of a trained fuzzy neural net are used as the knowledge base, we call the model a neuro-fuzzy expert system. This enables one to accommodate the merits of neuro-fuzzy computing in expert system design.

The block diagram of the basic modules of an expert system, fuzzy expert system, fuzzy neural net, connectionist expert system, neuro-fuzzy expert system and knowledge-based connectionist expert system have been provided in fig. 5.10. As stated above, a fuzzy neural net constitutes the knowledge base of a neuro-fuzzy expert system. (Note that this excludes other possible integrations, such as bringing the concept of Artificial neural networks (ANN) into the framework of fuzzy expert system). While the rules are collected by knowledge engineers for designing the knowledge base of a traditional expert system or fuzzy expert system, the connectionist models use the trained link weights of the neural net / fuzzy neural net to automatically generate the rules, either for later use in a traditional version or for providing justfication in the case of an inferred decision. This automates and also speeds up the knowledge acquisition process. The use of fuzzy neural nets helps in the handling of uncertainty at various levels (e.g. input, output, learning and neuronal) and generates fuzzy rules capable of more realistically representing real-life situations. The knowledge-based connectionist expert systems, on the other hand, initially encode crude domain knowledge among the connection weights of the neural net, thereby speeding up the training phase and generating better performance. Refined rules are later extracted from the less redundant trained network.

We now provide a few paragraphs in discussing the role, relevance, and need of the above more recent technological advancements in the building of more integrated and intelligent systems.

- **The role of fuzzy logic**

A fuzzy set A in a space of points R = {r} is a class of events with a continuum of grades of membership and is characterized by a membership function $\mu_A(r)$ which associates with each element in R a real number in the interval [0,1] with the value of $\mu_A(r)$ at r representing the grade of membership of r in A. Formally, a fuzzy set A with its finite number of supports r_1 , r_2 , ...r_t is defined as a collection of ordered pairs

$$A = \{(\mu_A(r_i), r_i), \quad i = 1, 2, ...t\}$$

where the support of A is an ordinary subset of R and is defined as

$$S(A) = \{r \mid r \in R \text{ and } \mu_A(r) > 0\}$$

Here, μ_i , the grade of membership of r_i in A, denotes the degree to which an event r_i may be a member of A or belong to A. Note that $\mu_i = 1$ indicates the strict containment of the event r_i in A. If, on the other hand, r_i does not belong to A then $\mu_i = 0$.

Fuzzy logic is based on the theory of fuzzy sets and unlike classical logic, it aims at modeling the imprecise or inexact modes of reasoning and thought processes (with linguistic variables) that play an essential role in the remarkable human ability to make rational decisions in an environment of uncertainty and imprecision. This ability depends, in turn, on our ability to infer an approximate answer to a question based on a store of knowledge that is inexact, incomplete or not totally reliable. In fuzzy logic everything, including truth, is a matter of degree [50]. Zadeh has developed a theory of approximate reasoning based on fuzzy set theory. By approximate reasoning we refer to a type of reasoning that is neither very exact nor very inexact. This theory aims at modeling the human reasoning and thinking process with linguistic variables [53] in order to handle both soft and hard data, as well as various types of uncertainties. Many aspects of the underlying concept have been incorporated in designing decision-making systems [54].

Because fuzzy sets are a generalization of the classical set theory, the embedding of conventional models into a larger setting endows fuzzy models with greater flexibility to capture various aspects of incompleteness or imperfection (i.e. deficiencies) in whatever information and data are available about a real process. Assignment of membership functions of a fuzzy subset is subjective in nature, and reflects the context in which the

problem is viewed. It cannot be assigned arbitrarily. In some cases, it is convenient to express the membership function of a fuzzy subset in terms of standard S and Π functions [55].

- **Relevance of connectionist models**

The various uncertainty management schemes of traditional expert systems share some common problems. For example, a willing human expert able to accurately quantify expertise is needed. The transfer of the knowledge takes place gradually through many interviews between the expert and the system, and is therefore very time consuming. Usually humans are prone to be easily biased and thus the quality of knowledge extracted from the experts depends greatly on the methods used for assessment. Moreover, large knowledge bases need to be searched quickly and it is also very important to check that this knowledge base remains consistent as more information is accumulated. It would therefore be welcome if knowledge assessment could be automated by freeing it from human intervention, thereby avoiding human bias and subjectivity.

It is worth mentioning that the most difficult, time-consuming and expensive task in building an expert system is constructing and debugging its knowledge base. In practice, the knowledge base construction can be said to be the only real task in building an expert system considering the proliferating presence of expert shells. Several approaches have been explored for easing this knowledge-acquisition bottleneck.

Connectionist expert systems [52] offer an alternative approach both to the knowledge base construction as well as the inferencing phase, providing interaction with the user accompanied by justification(s) of the conclusion(s) reached. Rules are not required to be supplied by humans. Instead, the connection weights of a trained neural network encode among themselves, in a distributed fashion, the information conveyed by the input-output combinations of the training set. The problems faced by traditional expert systems regarding the dificulties in normalizing across different experts' scales, conversion from human experessions to numerical terms, bias of the expert(s), generation of contradictory rules by the experts, etc, may be overcome here. The use of the learning technique of neural networks enables the model to extract the information inherent in the data (that is not utilised in the traditional models) and allows dynamical adjustments to changes in the environment. It also enables one to handle a complicated environment for which either no mathematical model exists, or, even if it exists is so strongly nonlinear that a design method does not exist. Besides, the various characteristics of neural nets, viz, generalization, tolerance to noise, graceful degradation at the border of the domain of expertise, ability to discover new relations between variables, etc. are in-built and hence can be exploited by the connectionist expert systems.

Connectionist expert systems use the connection weights W of the trained neural network to form the knowledge base. The magnitudes of these connection weights are used to generate rules in order to justify any decision. The maximum weighted paths from the output layer to the input layer are used in the process [52, 56]. Note that in traditional expert systems, the knowledge base is formulated in terms of rules by interaction with the experts. On the other hand, here the rules may be automatically extracted from the trained connection weights, that form the knowledge base.

- **Need for neuro-fuzzy computing**

Both neural networks and fuzzy systems are trainable dynamic systems that estimate input-output functions. They estimate a function without any mathematical model and learn from experience with sample data. A fuzzy system adaptively infers and modifies its fuzzy associations from representative numerical samples. Neural networks, on the other hand, can blindly generate and refine fuzzy rules from traning data [57]. Fuzzy systems and neural networks also differ in how they estimate sampled functions, the kind of samples used and how they represent and store these samples. Fuzzy systems estimate functions with fuzzy set samples (A_i, B_i), while neural systems use numerical point samples (x_i , y_i), where both kinds of samples reside in the input-output product space $X \times Y$. Hence the input-output mapping corresponds to $f : X \rightarrow Y$ in both cases.

The fuzzy theory is considered to be advantageous in the logical field, and in handling higher order processing easily. The higher flexibility is a characteristic feature of neural nets produced by learning, and hence this suits data-driven processing better [58].

For the last few years, researchers all over the world [45-48, 59, 60] have been trying to combine the merits of fuzzy and neural approaches under the heading neuro-fuzzy computing for building more intelligent decision making systems. This enables one to incorporate the generic advantages of artificial neural networks like massive parallelism, robustness and learning in data-rich environments into the expert system model. The modelling of imprecise and qualitative knowledge as well as the transmission of uncertainty are possible through the use of fuzzy logic. Besides this generic advantage, the neuro-fuzzy approach provides some application specific merits in the following way. For example, in the case of classification-type connectionist expert systems one is typically interested in exploiting the capability of neural nets in generating the required (linearly nonseparable) decision regions. The uncertainties involved in the input description and output decision are also taken care of by the concept of fuzzy sets. It is observed that in certain cases a neuro-fuzzy model performs better than either a neural network or a fuzzy system considered individually [61, 62].

- **Knowledge based networks**

Recently, there have been some attempts in improving the performance of expert systems by using knowledge-based networks which use the domain knowledge to determine the initial structure of the network. Such a model has the capability of outperforming a standard MLP as well as other related algorithms including symbolic and numerical ones [63,64]. However, in the absence of knowledge one has to resort to a purely data-driven mode of learning as in the simple connectionist expert models. When the initial knowledge fails to explain many instances, additional hidden units and connections need to be added (often empirically). The initial encoded knowledge may be refined with experience by performing learning in the data environment. The resulting networks generally involve less redundancy in their topology.

A comparative analysis of the basic features of these models with those of the traditional and connectionist (non-fuzzy) versions is provided in Table 5.4.

Table 5.4

Comparative study of various expert systems

	Expert system	Connectionist expert system	Neuro-fuzzy expert system	Knowledge-based co nnectionist/ Neuro-Fuzzy expert system
Knowledge base	Knowledge acquisition and representation in the form of rules, frames, semantic nets or belief networks	Connection weights of trained neural net that were initialised with small random values	Connection weights of trained fuzzy neural net that were initialised with small random values	Connection weight of trained nonfuzzy/ fuzzy neural net that were initialised with crude domain knowledge in rule form with binary link weights apriori class information and distribution of pattern points
Knowledge refinement	Addition of new knowledge (say, as new rules)	Empirical addition of hidden nodes/links	Empirical addition of hidden nodes/links	Network optimization using growing and pruning of nodes /links, based on training data and additional knowledge
Iferencing	Matching facts with the existing knowledge base	Presentation of crisp input, forward pass and generation of crisp output	Presentation of fuzzy input, forward pass and generation of fuzzy output	Presentation of input , forward pass and generation of output
Rule generation		Crip rules obtained during backward pass using changes in levels of input and output units, magnitude of connection weights	Fuzzy rules obtained during backward pass using node activations and link weights	Rules obtained during backward pass; negative rules also possible

5.3. **Why it is an expert system domain**

There are two distinct parts under this aspect : (i) why does the domain demand ES-technology ? and (ii) why does the ES-technology suit the domain ? Let us consider the first issue, the foundation of which may be traced in **Chapter 1 (section 1.3)** : needs of the region. Constant monitoring of growth and development performance of children of this region is highly required. As an ideal case, atleast one paediatrician should be placed at each rural health centre. But, for a developing country like India, paediatrician is really a scarce commodity. To mitigate such needed expertise, an automated knowledge-based consultation system may be helpful. No other alternative is now really feasible. Let us now examine how the ES-technology suits the problem domain. The key attributes of a domain, to be a good candidate for expert system domain, are neither all absolute nor limited to the following [65-67] :

- The Knowledge associated with the domain must be bounded;
- Non-algorithmic approach is more useful than conventional algorithmic approach;
- Human experts or literature should be available or some prior case studies should be conducted to gather knowledge where human experts knowledge is neither adequate nor any literature is available;
- There should be some advantage to using computers with a significant payoff;
- The complete logic is not known in advance;
- Primarily it requires symbolic reasoning;
- There may be use of heuristics by the expert(s). Problems require multicriteria decision making (MCDM) [68] or use of incomplete or uncertain information;
- The domain is fairly stable or at least slowly changing;
- No alternative solution to the problem is being pursued or is expected to be pursued. The present solution under investigation for the domain problem will be used as long-term basis.

The domain under consideration is child growth and development of a specific geographical region of India. The knowledge associated with the domain is large and varied. The boundaries of the region are somewhat defined but defining such boundaries of knowledge in paediatrics may not readily be possible. Reaching a conclusion with unbounded knowledge may not be possible. Therefore, experts

somehow confine their knowledge while handling any problem. We confine ourselves within such expertise knowledge during the knowledge acquisition process. It may be stated as a fact that the stimulation reactions or disease patterns on human body do not obey any particular algorithm or a particular set of algorithms. So, it is better to use here a non-algorithmic approach. Multiple experts on child growth and development are available in the region. We have no doubt to state that a significant pay off from the completed system will be achieved. This pay off may be in terms of tangible benefits or may be in terms of social values. For this kind of diagnostic problems in general, and in particular, for this region with typical characteristics, complete and sound knowledge may not be available in advance. This requires the use of expert systems technology where one may expect the ease of updating and maintainability of knowledge base. Primarily, the domain requires to deal with some symptoms (some symbols) like "AXIAL MUSCLE TONE" or "SPONTANEOUS GESTURES" or "RHYTHMS" (**Appendix A**). So, symbolic reasoning is the primary component here. It is such a domain where the use of heuristics by the experts gained in a number of years of practice will be useful. Multiple criteria decision making and incomplete or uncertain information processing are also the characteristics of the domain. The characteristics of the domain under consideration are of fairly stable nature. It is unlikely that the characteristics of this region will change abruptly in near future. It is also unlikely that the characteristic diagnostic parameters of child as well as the expected values will change drastically. From the socio-economic point of view, no better solution, seems feasible. It may not be possible to appoint even one human expert on child care per health centre of a developing country. The needs of the domain may be fulfilled by the present system under investigation. It is expected that the full system once achieved will be used as long-term basis, an important pay off.

5.4. What requirements the domain lays on an expert system

With the above justifications of using expert systems technology for the domain, let us now investigate the requirements the domain lays on an expert system.

- **Portability**

To have its increased usage an expert system is expected to be portable. This essentially means that the system can be run on different types of target machines which can be procured at low cost and can be transported easily to different remote health centres. Moreover, the recurring expenditure should be low in terms of power consumption, maintenance etc. During the system development, one has to select a software development tool to satisfy the said purpose. For example, one may suitably select PROLOG / LISP or an ES-shell or a tool-kit based on PC running under MS-DOS. Summarily, a low cost and easily manageable by the end users PC-based system is being proposed here. This portability feature should certainly encourage the usage

issue discussed in **chapter 1** (**section 1.3.2**). It should be easier then for doctors who already have hardwares with them to procure this system. This may require a small upgradation rather than procuring specialised LISP-based machines or AI workstations.

- **Modifiability**

The domain knowledge in knowledge base may have to be enhanced owing to different reasons. Three specific reasons may here be noted. First, when complete and sound knowledge may not be available in advance, a fact for the present domain, existing system should easily and quickly incorporate the required changes, specially bearing in mind the state-of-the-art knowledge of the domain. Secondly, the complete and sound knowledge may not be possible to acquire in the initial stage of the knowledge acquisition process. At the later stage of the development, further enhancement would be required. Third, for its survival, a system should be of open type. This essentially means that the system should cope with the changing environment, obviously small, suggesting the modifiability feature to incorporate in the system. A closed system should eventually die. In a system, the modifiability has to be taken care of at two levels : i) at the design level, and ii) at the implementation level.

- **Dealing with inexact information**

In real world, we have the experience that sometimes either we have no knowledge about an object or we have some incomplete, fuzzy or uncertain knowledge about the object. But, one has to reason in this situation and has to reach a decision. For a paediatric domain this is more critical. An expert system should be capable of handling these inexact situations. The importance of the topic demands an elaborate discussion which has been provided in a separate chapter (**Chapter 8**).

- **Non-monotonic reasoning**

The information supplied by the parents / guardians of an investigative child is subjective sometimes. To deal with this subjective reply nonmonotonic reasoning (NMR) will be useful. NMR proceeds with its reasoning as if the assumptions are true with their definite certainty values. With its reasoning it reaches a conclusion. If one finds the conclusion to be absurd, it is demanding to change an assumption and / or to change the uncertainty values.

Question : Does your baby take 4-5 meals / day?
Answer : Doctor. She doesn't want to take meals.

With this reply (CF = 0.9) the doctor proceeds to other examinations. After overall examinations, the doctor finds that the baby is disease-free and the growth of the baby is normal. This finding, obviously, contradicts the reply from the mother whose CF may be 0.3 or 0.2. The associated facts and rules may have to be changed with this changing CF. This is how the nonmonotonic reasoning works to offer a safeguard to the subjective reply during a consultation session. NMR is also relevant in connection with modifiability. One can also have the view that uncertain reasoning itself has a nonmonotonic aspect.

- **Transparency**

For a firm diagnosis as well as for further course of action(s), a doctor may not be satisfied with the decision only offered by an expert system. He / she may demand the total reasoning path traversed by the system. It may, sometimes, also be demanded by the parents / guardians of the child under investigation, may be, for their mental satisfaction. Generally, 'HOW' and 'WHAT IF' types of transparency are expected. So, an explanation tracing procedure should be there, as a module, with the system. The most of the users of the system would be the general medical professionals and medical students who are not experts in paediatrics. This particular feature should assist them to view the chain of reasoning leading to a conclusion. This chain of reasoning should certainly assist a doctor for further analysis and treatment planning. This chain of reasoning should also be useful to nonmonotonic reasoning issue.

- **Learning facility with a dynamic knowledge base**

It may be useful to remember the results or facts of at least one previous consultation session for better comparison, especially for the paediatric field. It is true that the deficiency in growth and development should be estimated in comparison with a set standard i.e. milestones what are stored in static part of knowledge base. But, however, it should also be useful to estimate the increment / decrement of growth parameters in comparison with the previous consultation session. This should give us an idea about the parameters which need more attention. It may also be useful to avoid any repetitive questionnaire during interrogation with the child and / or with the parents / guardians. This is essentially a learning facility with the system. This facility should be useful with this type of application on the argument that there are a number of cases where paediatric patients are resistant to such interrogation. They may become hostile with any repetitive process. It is necessary to complete the process as quickly as possible. This facility may be achieved with a dynamic knowledge base. We call this dynamic portion of knowledge base as short-term knowledge base (STKB). This STKB may also help to achieve 'improved backtracking' compared to 'blind or chronological backtracking'. This STKB, we observe, may also play an active role on nonmonotonic reasoning.

- **Structured and modular data structure**

Let us now identify some key requirements of the domain in connection with its knowledge representation where structuredness and modularity are demanded for :

Managing a large and varied knowledge base

The domain knowledge size of child growth and development is significantly large and varied. In this situation, the knowledge can become unmanageable. To make it manageable, it will be worthwhile to use structured and modular data structure for knowledge representation.

Avoiding redundancy and thereby removing inconsistency

Any component of knowledge is expected not to be duplicated in a knowledge base either in the design phase or in the implementation phase. This redundant information requires more space and also leads to inconsistency problem during upgradation of knowledge. Using a structured and modular data structure one can avoid this redundancy problem.

High level of abstraction

An abstraction is a way of representing a group of related things by a single thing which expresses their similarities and suppresses their differences. For the present domain of child growth and development, the level of abstraction is expected to be high for the ease of proper diagnosis from a large and varied knowledge base. A high level of abstraction may be achieved using an equally highly structured and modular data structure for knowledge representation.

In **chapter 6**, a detail discussion has been provided on the knowledge representations schemes along with their relative merits and demerits.

5.5. **Conclusions** and **Discussion**

After a brief introduction to AI and expert systems technology, categories and application areas of expert systems with some examples have been provided. Some pages have been devoted for the discussion on the trends of applications of expert systems. Components of a typical expert system, typical features of an expert system, major stages of expert system development have been provided in brief. Then we have devoted some pages on the discussion on the types of expert systems with a note on the recent trends of the technology.

One can observe that recently case-based and / or model-based reasoning are preferred by some researchers in some domains. For the generation of more intelligent decision making systems some researchers propose fuzzy systems, some propose neuro-fuzzy models, some propose knowledge-base networks model and some propose connectionist model. They have their relative merits and demerits. A comparative study of the various methodologies has been provided in tabular form.

However, in our present study we are confined to the development of a rule based fuzzy object-oriented knowledge based system for the domain.

References

1. T. Moto-Oka et al. Challange for knowledge information processing systems. Proc. of Int. Conf. on FGCS; Oct. 19-22,1981, 1-85.

2. D. Dutta Majumder. Pattern recognition, image processing, artificial intelligence and computer vision in FGCS. Sadhana. The Indian Academy of Sciences; Bangalore, 1986.

3. D. Dutta Majumder. Functional and technical objectives of FGCS research. Journal of institute of engineers (India); vol. 67, Sept. 1986, 1-10.

4. D. Dutta Majumder. A unified approach to artificial intelligence, pattern recognition, image processing, and computer vision in FGCS. Int. J. Inf. Science; 1988, 1-41.

5. D. Dutta Majumder. Computer vision and knowledge based computer systems. JIETE ; vol. 34, no. 3, May-June, 1988, 230-245.

6. Frederick Hayes-Roth, Donald A. Waterman and Douglas B. Lenat eds. Building Expert Systems, Reading, MA., Addison-Wesley; 1983, 13-16.

7. Donald A. Waterman. A Guide to Expert Systems. Addison-Wesley Publishing Company; 1985, 136-140.

8. John Durkin. Expert Systems : A view of the field. IEEE Expert; April 1996, 56-64.

9. A. G. Boulanger. The expert system Plant/cd : A case study in applying the general purpose inference system ADVISE to predicting black cutworm damage in corn. M. S. thesis; Computer Science Deptt., University of Illinois at Champaign-Urbana, July 1983.

10. B. G. Buchanan and E. A. Feigenbaum. Dendral and Meta-dendral : their applications dimension. Artificial Intelligence; vol.11,1978.

11. J. S. Bennett and C. R. Hollander. DART : An expert system for computer fault diagnosis. Proceedings IJCAI-81; 1981, 843-845.

12. J. Dekleer and G. J. Sussman. Propagation of constraints applied to circuit synthesis. Circuit Theory and Applications. Vol.8, John Wiley and Sons, Ltd.;1980, 127-144.

13. J. Bennett, L. Creary, R. Engelmore and R. Melosh. A Knowledge-based consultant for structural analysis. Computer Science Dept., Stanford University, Stanford, California, September, 1978.

14. J. Gaschnig. Prospector : an expert system for mineral exploration. In Machine Intelligence. Infotech State of the Art Report 9; no.3,1981.

15. M. G. Valtorta, B. T.Smith and D. W. Loveland. The graduate course advisor : A multi-phase rule-based expert system. Proc. of the IEEE workshop on principles of knowledge-based systems, IEEE Computer Society, IEEE Computer Society Press; 1109 Spring Street, Silver Spring, Md., 1984.

16. Carole D. Hafner. Representation of knowledge in a legal information retrieval system. In R. Oddy, S. Robertson, C. Van Rijsbergen, and P. Williams (eds.) Information Retrieval Research, London : Butterworths and Co.; 1981.

17. M. S. Fox, B. Allen and G. Strohm. Job-shop scheduling : an investigation in constraint-directed reasoning. Proceedings AAAI-82; 1982, 155-158.

18. W. A. Martin and R. J. Fateman. The MACSYMA system. Proc. of the Second Symposium on symbolic and algebraic manipulation; March 1971, 59-75.

19. L. E. Rodewald. Baby : an expert system for patient monitoring in a newborn intensive care unit. M. S. thesis; Computer Science Dept., University of Illinois, Champaign-Urbana, 1984.

20. E. H. Shortliffe, S. G. Axline, B. G. Buchanan, T. C. Merigan and S. N. Cohen. An artificial intelligence program to advise physicians regarding antimicrobial therapy. Computers and Bio-medical Research; vol.6,1973, 544-560.

21. D. Michie, S.Muggleton,C. Riese and S. Zubrick. Rulemaster : a second-generation knowledge-engineering facility. Proc. of the first Conf. on artificial intelligence applications, IEEE Computer Society; December 1984.

22. G. R. Ferguson. Aircraft maintenance expert systems. Master's thesis; Air Force Institute of Technology, Wright-Patterson AFB, Ohio, November 1983.

23. D. R. Barstow. Knowledge engineering in Nuclear Physics. Proc. IJCAI-79; vol.1, 1979, 34-36.

24. D. Chester, D. Lamb and P. Dhurjati. Rule-based computer alarm analysis in chemical process plants. Proc. of the Seventh Annual Conf. on Computer Technology, MICRO-DELCON 84, IEEE; March 1984, 22-29.

25. F. J. Dickey and A. L. Toussaint. ECESIS : an application of expert systems to manned space stations. Proc. of the first Conf. on Artificial Intelligence applications, IEEE Computer Society; December 1984.

26. E. Turban. Expert Systems and Applied Artificial Intelligence. Macmillan Publishing Company; Int. ed.,1992, 725-727.

27. Artificial Intelligence Letter V, no. 10, October 1989, Texas Instruments, Data systems group, Austin, Texas.

28. E. Turban. Expert Systems and Applied Artificial Intelligence. Macmillan Publishing Company; Int. ed.,1992, 720-723.

29. E. Turban. Expert Systems and Applied Artificial Intelligence. Macmillan Publishing Company; Int. ed.,1992, 703-704.

30. S. I. Goyal et al. COMPASS : An Expert System for Telephone switch maintenance. Expert Systems; July, 1985.

31. E. Turban. Expert Systems and Applied Artificial Intelligence. Macmillan Publishing Company; Int. ed.,1992, 732.

32. P. Suetens and A. Oosterlinck. Using expert systems for image understanding. Associated with the National Fund for Scientific Research (NFWO); Belgium, 1987, 61-73.

33. B. E. Prasad, T. S. Perraju, G. Uma and P. Umarani. An expert system shell for Aerospace Applications. IEEE Expert; August 1994, 56-64.

34. R. Simon. The Morning after. Forbes; vol.140, no.8, October 19, 1987,164-168.

35. D. Dutta Majumder. Artificial intelligence and expert systems : Its role in Industrial System Development. Nat. Symp. on Elec. and Appl.; Sept.,1989.

36. J. L. Kolodnor. Extending problem solver capabilities through case-based inference. Proc. of the fourth Int. workshop on machine learning; Los Altos, CA, Morgan Kaufmann, 1987.

37. E. Turban. Expert Systems and Applied Artificial Intelligence. Macmillan Publishing Company; Int. ed.,1992, 225-226.

38. S. Mitra and S. K. Pal. Neuro-fuzzy Expert Systems : relavance, features and methodologies. JIETE; vol.42, nos.4 and 5, July - Oct 1996, 335-347.

39. D. E. Rumelhart and J. L. McClelland. Parallel distributed processing : Explorations in the microstructures of Cognition. eds., vol.1, Cambridge, MA : MIT Press; 1986.

40. R. P. Lippmann. An introduction to computing with neural nets. IEEE Acoustics, Speech and Signal Processing Magazine; vol.4,1987, 4-22.

41. T. Kohonen. Self-organization and associative memory. Berlin, Springer-Verlag; 1989.

42. J. Hertz, A. Krogh and R. G.Palmer. Introduction to the theory of Neural computation. Reading, MA , Addison-Wesley; 1994.

43. L. A. Zadeh. Fuzzy Sets. Information and Control; vol.8, 1965, 338-353.

44. G. J. Klir and T. Folger. Fuzzy Sets, Uncertainty and Information. Reading, MA, Addison-Wesley; 1989.

45. J. C. Bezdek and S. K. Pal. Fuzzy models for pattern recognition : methods that search for structures in data, (eds.), New York, IEEE Press; 1992.

46. Y. H. Pao. Adaptive pattern recognition and neural networks. Reading, MA, Addison - Wesley; 1989.

47. Proceedings of third International Conference on fuzzy logic and neural networks (IIZUKA 94); Iizuka, Fukuoka, Japan, July 1994.

48. Proceedings of Third International Conference on Fuzzy Systems (FUZZ-IEEE); Florida, USA, June 1994.

49. H. J. Zimmermann. Fuzzy sets, decision making and expert systems. Boston, MA, Kluwer, Academic Publishers; 1987.

50. L. A. Zadeh. The role of fuzzy logic in the management of uncertainty in expert systems. Fuzzy Sets and Systems; vol.11, 1983, 199-227.

51. A. Kandel. Fuzzy Expert Systems. (ed.), Boca Raton, CRC Press, Inc.,1991.

52. S. I. Gallant. Connectionist Expert Systems. Communications of the Association for Computing Machinery; vol.31, 1988, 152-169.

53. L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning : part 1, 2, and 3. Information Sciences; vol.8, 8, 9, 1975, 199-249, 301- 357, 43-80.

54. M. M. Gupta, A. Kandel, W. Bandler and J. B. Kiszka. Approximate reasoning in Expert Systems. (eds.), Amsterdam, North Holland, 1985.

55. S. K. Pal and D. Dutta Majumder. Fuzzy mathematical approach to pattern recognition. New York, John Wiley (Halsted Press); 1986.

56. S. Mitra and S. K. Pal. Fuzzy multi-layer perception,inferencing and rule generation. IEEE Transactions on Neural Networks; vol.6, 1995, 51-63.

57. B. Kosko. Neural Networks and Fuzzy Systems. New Jersey, Prentice Hall; 1991.

58. H. Takagi. Fusion technology of fuzzy theory and neural network-survey and future directions. Proceedings of the International Conference on fuzzy logic and neural networks; Iizuka, Japan, 1990, 13-26.

59. M. M. Gupta and D. H. Rao. On the principles of fuzzy neural networks. Fuzzy Sets and Systems; vol. 61, 1994, 1-18.

60. J. J. Buckley and Y Hayashi. Fuzzy Neural Networks : A survey. Fuzzy Sets and systems; vol. 66, 1994, 1-13.

61. S. K. Pal and S. Mitra. Multi-layer perception, fuzzy sets and clasiification. IEEE Transactions on Neural Networks; vol.3, 1992, 683-697.

62. S. K. Pal and D. P. Mandal. Linguistic recognition system based on approximate reasoning. Information Sciences; vol.61, 1992, 135-16.

63. L. M. Fu. knowledge-based connection for revising domain theories. IEEE Transactions on Systems, Man and Cybernatics; vol.23, 1993, 173-182.

64. G. G. Towell and J. W. Shavlik. Knowledge-based artificial neural networks. Artificial Intelligence; vol.70, 1994, 119-165.

65. B. G. Buchanan, D. Barstow, R. Bechtal, J. Bennett, W. Clancey, C. Kuiilowski, T. Mitchell and D. A. Waterman. Constructing an expert system in F. Hayes-Roth, D. A. Waterman, D. B. Lenat, (eds.), 1983, Building Expert Systems. Addison-Wesley, Reading, Mass.; 1983, 127-167.

66. R. H. Anderson. Four criteria for choosing expert system applications. Asian Computer Monthly; April, 1985, 21-22.

67. D. S. Prerau. Choosing an expert system domain, in G. Guida and C. Tasso, eds., 1989. Topics in expert system design; North-Holland, Amsterdam, 27-43.

68. D. Dutta Majumder. Fuzzy mathematics and uncertainty management for decision making in science and society, Computer Science and Informatics; vol.23, no.3, 1993,1-31.