Artificial Intelligence, Expert Systems and the Domain

3.1. Introduction

Artificial Intelligence (AI) is a sub field of computer science concerned with the study and creation of computer systems that exhibit some form of intelligence. 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 programs, 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 programs to meet some formal specifications - we shall say that such systems possess some degree of artificial intelligence. 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.

One area of AI that can claim a large measure of responsibility for the current AI awareness in the world is expert systems technology which are computer software that embody human expertise. The scarcity of human expertise exists in almost all fields, such as diagnosis, planning, 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. 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 equipment that can act as intelligent assistance to human experts as well as assisting people who otherwise might not have access to expertise.

Expert Systems (ES) are knowledge intensive programs that solve problems in a domain that requires considerable amount of technical expertise. A desired additional characteristic which many would regard fundamental, is the capacity of the system on demand to justify its own line of reasoning in a manner directly intelligible to the inquirer. ES is a computer program that encodes the knowledge and reasoning of expert(s) in a given area and applies this knowledge and reasoning to derive problem-solving interfaces for the user of the system. To solve a given particular problem, the ES examines facts about the problem supplied by the user. The facts are interpreted in
terms of the system's search of its knowledge base and processed through its built-in reasoning procedures to find a solution.

The process of building an ES is inherently experimental. In order to have successful development of an ES in a domain, different potential issues have to be fixed up which demands a thorough analysis. This chapter is meant for fixing up 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.

In the next section, some potential issues such as categories and application areas, desirable features, life cycles, typical architecture, and types of expert systems have been discussed. In section 3.3, an attempt has been made to explain why the present problem domain may be considered suitable for an expert system domain. In section 3.4, we have tried to find out what requirements the domain lays on an expert system. A discussion has been provided at the end.

3.2. Expert systems technology

3.2.1. 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 thirteen functional categories [1,2] shown in table 3.1. In table 3.2, we indicate application areas for which some expert system has been developed [2,3].

In Aerospace, we may cite here REX as an example of expert system shell. REX [4] is an object-oriented, asynchronous real-time expert system shell to meet the challenges of the dynamic aerospace environment.

In Agriculture, different expert systems have been reported which have been discussed in the section 3.2.2.

In Business, we may cite here SUTA as an example of expert system. SUTA [5], 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.
Table 3.1. Generic categories of expert system applications.

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

In Chemistry, different expert systems have been reported such as DENDRAL, CONGEN, CRY SALIS, C-13, GA1, META-DENDRAL, MOLGEN, OCSS, SECS, SEQ, SPEX, SYNCH EM, SYNCH EM2, TQMSTUNE. DENDRAL [6], for example, infers the molecular structure of unknown compounds from mass spectral and nuclear magnetic response data. Knowledge in DENDRAL is represented as procedural code for the molecular structure generator and as rules for the data-driven component and evaluator.

In Communications, we may cite here COMPASS as an example of expert system. COMPASS [7] (Central Office Maintenance Printout Analysis and Suggestion System) analyzes maintenance printouts of telephone company control switching equipment and suggests maintenance actions to be performed.
Table 3. 2. Application areas of expert systems.

<table>
<thead>
<tr>
<th>Aerospace</th>
<th>Law</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agriculture</td>
<td>Manufacturing</td>
</tr>
<tr>
<td>Business</td>
<td>Mathematics</td>
</tr>
<tr>
<td>Chemistry</td>
<td>Medicine</td>
</tr>
<tr>
<td>Communications</td>
<td>Meteorology</td>
</tr>
<tr>
<td>Computer System</td>
<td>Military Science</td>
</tr>
<tr>
<td>Education</td>
<td>Mining</td>
</tr>
<tr>
<td>Electronics</td>
<td>Physics</td>
</tr>
<tr>
<td>Engineering</td>
<td>Process Control</td>
</tr>
<tr>
<td>Environment</td>
<td>Power Systems</td>
</tr>
<tr>
<td>Geology</td>
<td>Science</td>
</tr>
<tr>
<td>Image processing</td>
<td>Space Technology</td>
</tr>
<tr>
<td>Information Management</td>
<td>Transportation</td>
</tr>
</tbody>
</table>

In Computer Systems, different expert systems have been reported such as DART, CRIB, IDT, ISA, MIXER, R1-SOAR, TIMM / TUNER, XCON, XSEL, YES/MVS. DART [8], for example, assists in diagnosing faults in computer hardware systems using information about the design of the device being diagnosed.

In Education, we may cite here MIKE as an example of expert system. MIKE [9], 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.

In Electronics, different expert systems have been reported such as SYN, ACE, BDS, CADHELP, COMPASS, CRITTER, DAA, DFT, EL, EURISO, FG502-TASP, FOREST, INATE, MESSAGE TRACE ANALYZER, NDS, PALLADIO, PEACE, REDESIGN, SADD, SOPHIE, TALIB, TRANSISTOR SIZING SYSTEM. SYN [10], for example, assists engineers in synthesizing electrical circuits.
In Engineering, different expert systems have been reported such as SACON, CONPHYDE, DELTA, NPPC, REACTOR, SPERIL-I, SPERIL-II, STEAMER. SACON [11], for example, helps engineers determine analysis strategies for particular structural analysis problems. SACON is a backward chaining, rule-based system implemented in EMYCIN.

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 Geology, different expert systems have been reported such as PROSPECTOR, DIPMETER ADVISOR, DRILLING ADVISOR, ELAS, HYDRO, LITHO, MUD. PROSPECTOR [12], 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. PROSPECTOR uses a combination of 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 Image processing, we may cite here 3DPO [13] 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.

In Information Management, different expert systems have been reported such as GCA, CARGUIDE, CODES, EDAAS, FOLIO, IR-NLI, PROJCON, RABBIT, RESEDA. GCA [14], 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.

In Law, different expert systems have been reported such as LRS, AUDITOR, DSCAS, JUDITH, LDS, LEGAL ANALYSIS SYSTEM, LRS, SAL, SARA, TAXADVISOR, TAXMAN. LRS [15], for example, helps lawyers retrieve information about court decisions and legislation in the domain of negotiable instrument law, an area of commercial law that deals with checks and promissory notes.

In Manufacturing, different expert systems have been reported such as ISIS, IMACS, PTRANS. ISIS [16], 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. ISIS uses a frame-based knowledge representation scheme together with rules for resolving conflicting constraints.

In Mathematics, different expert systems have been reported such as MACSYMA, MATHLAB 68, ADVISOR. MACSYMA [17], 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 BABY, AI/COAG, AI/MM, AI/RHEUM, ABEL, ANGY, ANNA, ARAMIS, ATTENDING, 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. BABY [18], for example, aids clinicians by monitoring patients in a newborn intensive care unit (NICU). 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 [19] assists physicians in the selection of appropriate antimicrobial therapy for hospital patients with bacteremia, meningitis and cystitis infections. 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.

In Meteorology, we may cite here WILLARD as an example of expert system. WILLARD [20] helps meteorologists forecast the likelihood of severe thunderstorms occurring in the central United States. 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 [21], for example, helps aircraft technicians diagnose aircraft problems. MES is a forward chaining, rule-based system.
In Mining, we may cite here DustPro as an example of expert system. DustPro [22] 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. The system is now in operation in more than 200 mines. DustPro advises in three areas: control of methane gas emission, ventilation in continuous operations, and dust control for the mine's machines.

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.

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. Knowledge is represented in two ways - as a set of rules controlled by forward chaining and as a causal model in network form.

In Power systems, we may cite here ENERGY MANAGEMENT as an instance of expert system. ENERGY MANAGEMENT [25] provides the utility needed a real-time ES integrated with a signaling system as well as with existing databases in hydroelectric plant. 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 Science, different expert systems have been reported such as DENDRAL, MOLGEN. DENDRAL [6] 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 Space Technology, different expert systems have been reported such as ECESIS, FAITH, KNEECAP, LES, NAVEX, RBMS and RPMS. ECESIS [26], 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 hybrid architecture involving both rule-based and semantic net formalisms, and it uses the Bayesian scoring model developed for PROSPECTOR to handle uncertainty.

In Transportation, we may cite here CARGEX as an example of expert system. CARGEX [27] is an expert system constructed to assist in making loading decisions. 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.

The predominant role of expert systems has been the diagnosis. One reason for the result is that this is the role most experts play. Fields such as medicine, engineering, agriculture, 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 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 [2].

3.2.2. Expert systems in agriculture

Advance in ES technology and its application to agriculture problems suggested that an ES approach might provide farm managers, extension workers and farmers with ready access for problems related to agriculture such as crop selection/management, weed management, irrigation and drainage management, soil conservation, insect pest and disease management, agricultural machineries, and more. A detailed review on ES in agriculture management can be seen in Peart et al. [28]; Lambert and Wood [29]; Jones [30]; Barret et al. [31]; Carrascal and Pau [32]; Edward-Jones [33]; Mohan and Arumugam [34] and recently in 1999, Kumar and Mohanti [35] have made a remarkable review.

Many ES and DSS found their application only in the later part of the last decade. The development of ES and DSS applications and their continuous improvement has resulted in their expansion and application in the different domains of agriculture.
- **Crop selection /management**

COMAX (COtton MAnagement eXpert), was developed to advise cotton growers on crop management at farm level by Lemmon [36]. It was developed to determine three factors: nitrogen requirements, irrigation schedules and the crop maturity date. This is the first integration of an expert system with a simulation model for daily use in farm management. Palmer [37] reported a soybean variety selection expert system for selection of crop varieties. The field characteristics, water availability and prevalence of disease were considered to recommend a variety of soybean to plant. Expert Systems for crop planning and farm management have also been developed.

CROPLOT developed by Nevo and Amir [38] is intended for determining the suitability of crop to given plot. It was made as a decision aid to plan the production of field crops such as cotton, corn and wheat on farm under uncertain conditions. Morgan et al. [39] also described an ES for crop variety selection. The developed system was designed to consider the soil characteristics. Halterman et al. [40] presented double cropping expert system. Plant [41] described an integrated expert decision support system for agricultural management. Stone and Toman [42] described an ES called COTFLEX to assist cotton producers in Texas, USA. This ES is linked to a large database and models for effective decision making for cotton production. Another ES, 'Soybean Management Alternatives using Real-time Simulation of Yield' (SMARTSOY) was developed by Batchelor et al. [43] in 1989 to cover all aspects of soybean crop management. McGregor and Thornton [44] developed a variety selection ES for winter wheat. King et al. [45] developed an ES for malting barley management (MKBS).

CROPS, a whole-farm crop rotation planning system to implement sustainable agriculture, has been developed by Buick et al. [46]. The ES developed by Nevo et al. [47] was based on the LP technique for optimal crop planning. This ES is strongly integrated with the LP algorithm in a modular format in such a way that the LP output is evaluated and refined by the ES.

Mohan and Arumugam [48] presented an ES for selecting among multiple crop types in a large region in South India. Availability of water and other resources, climate, soil characteristics and farmers-related factors were comprehensively entailed in representing the domain knowledge.

- **Weed management**

In weed control management, ES can assist the farmer/farm managers for better strategies. Edward-Jones et al. [49] developed an ES to provide information and assist
in giving recommendations to sugar beet advisors an appropriate herbicides, mixtures and sequences for the range of sugar beet weed problems in UK. The system was developed using Knowledge Pro (Knowledge Garden Inc., Nassau, NV). Another ES, "Weed Advisor" was designed by Pasqual [50] to identify and control common weeds in crops of wheat, triticale, barley and oat. It was developed using expert system shell Personal Consultant TM Plus (by Texas Instruments). It assists in identifying weeds, offers alternative control measures (if any), indicates what treatments should not be used and provides additional herbicide information. Weed Advisor also makes users more aware of the factors that need to be considered when selecting weed control strategies and can educate users by explaining the reasoning behind the conclusion or advice.

- **Irrigation and drainage**

An ES that deal with irrigation scheduling primarily use soil moisture and climatic data. An irrigation strategy depends on the crop, soil and climate as well as the irrigation system, application rates and the scheduling mechanism.

In 1986, Thompson and Peart [51] developed an ES for irrigation scheduling based on only soil moisture data. McClendon et al. [52] presented an integrated simulation-expert system approach (SMARTSOYIRRIG) for center pivot irrigation management for soybean.

Doraisamy [53] presented an integrated expert decision support system for irrigation scheduling. ETES, an integrated front-end ES, was developed by Mohan and Arumugam [54] to select suitable evapo-transpiration estimation methods under South Indian climatic conditions. In 1996, Thomson [55] reported an irrigation scheduling decision tool.

An ES for real-time operation of reservoir system was first discussed by Floris et al. [56]. This ES utilizes real time hydro-meteorological inputs and applies heuristics to produce real time reservoir operational guidelines. Hershaeur et al. [57] developed an ES for irrigation water distribution through a canal system. This system incorporated devise operational rules for irrigation water distribution without optimization model.

Srinivasan et al. [58] made an attempt to link the reservoir operation with the operation of canals and distributors through an ES (ESIM) in a multiphase manner. The ES considered on farm and main system irrigation management together and addresses the issues related to water requirements and water availability for each system element (branch canal, distributory, etc.). Kumar et al. [59] developed an ES (KBS) which was
intended for the site suitability assessment of sprinkler irrigation type including economic evaluation.

Stone and Toman [60] described an ES called COTFLEX. This ES has an irrigation component among the farm management activities. Similar to COMAX system, it also combines both heuristic and algorithm components to make decision on irrigation application.

IRRIGATOR, developed by Clarke et al. [61] was designed to provide decisions on supplemental irrigation schedules for fruits and vegetables in Ontario, Canada. Plant et al. [62] described an ES called CALEX/Cotton designed for scheduling irrigation for crops in the San Joaquin Valley, California.

Bhatti [63] presented an ES for optimal operation of a reservoir system in Pakistan. This ES includes the cognitive and computational component involved in the reservoir operation. An important feature of this ES is that a dynamic programming (DP) algorithm was integrated with the ES to provide optimal operational knowledge in addition to judgmental and procedural knowledge base.

Goroth and Macvicer [64] proposed an ES, called OASIS (Operations Assistant and Simulated Intelligent System), designed to coordinate the operation of over 200 water control structures. It provides real-time display of important meteorological data and system status and generates various alternative water control policies.

In 1995, Arumugam [65] developed an ES called TANKES which covered optimal operation of a tank irrigation system using dynamic programming model to derive optimal weekly irrigation release to be provided to the optimal crop areas resulting from a LP model. Lilburne et al. [66] described a DSS to evaluate irrigation management plan (IMP). This integrated a simulation model SWIM, a decision tree and scientific soil hydraulic data. The simulation model is used to estimate the likely water requirement of the grower under IMP. The decision tree represents expert heuristics on the effect of the various irrigation strategies. The soil hydraulic data provides soil hydraulic properties to SWIM and to the decision tree. Local authorities and the likely environmental impact and water requirements of each grower.

There are application of the ES techniques in the selection, design and evaluation of irrigation system also. Bennett and Sneed [67] developed an ES for planning and design of irrigation systems. This ES does not evaluate the economic feasibility of irrigation systems. Hart et al. [68] addressed the problem of irrigation system selection through two expert systems. The first ES employs user input to rate irrigation system
from most to least favourable. The second ES recommends the most appropriate system. However, these expert systems do not provide any design or economic evaluation.

Haie and Irwin [69] developed an ES called EXSYS for use in land drainage decisions. It was designed to diagnose the causes of the drainage problem in the command area of an irrigation system. Factors such as water regime in the soil profile, presence of a cultivation pan or an impermeable layer below the top soil, etc. were considered.

- **Soil conservation**

The selection of appropriate practices for typical soil conservation plan is based on factors such as land use, topography and soil types, as well as economic and social factors. Experienced soil conservation engineers are, however, scarce and consequently there is a need for computer-based tools that permit engineers and planners to assess land use, perform erosion analysis and develop appropriate conservation plans with greater use and reliability.

The selection of appropriate practices in soil conservation is based on factors such as land use, topography and soil types, as well as economic and social factors. Expert systems, geographical information systems (GIS), hydrology and erosion process models can be integrated in the framework of an ES in such a way that it can provide combined capabilities [70]. Integrating expert systems with modern soil erosion, prediction simulation has been reported by Meyer [71]. Beckdash et al. [72] presented a DSS for agriculture and water quality management. Their approach involves a simulation model (Creams, a field-scale model for chemical runoff, and erosion from agricultural watersheds), databases, GIS and ES techniques. The databases includes climatic, geographic and soil data. The GIS linked to the DSS to provide topographic information.

Montas and Madramootoo [73] described a DSS for the planning of soil conservation system on a watershed scale and applied to an agricultural watershed in South Eastern Quebec, Canada. In 1994, an ES called ETCON by Amha et al. [74] was developed to make recommendation for land use planning and land utilization type requirements for a specified farm input level and management practice. ETCON makes an assessment of four land utilization types at three levels of measurable diagnostic parameters (land characteristics).
• Insect pest and disease management

There is no agricultural crop which becomes unaffected by insect pest and disease. Insect pest and disease are important factors to be controlled because of the loss they cause. If not controlled properly, they can make a considerable damage. ES and DSS can play a significant role in identification of insect pest and diagnosis of disease along with the control measures.

Roach et al. [75] developed an ES for insect identification in apple orchard. The system was developed using prolog. Knight and Cammell [76] described a DSS for forecasting *Aphis fabae* outbreaks in fields of spring-sown beans. This system was an integral part of FLYPLAST [77].

In between 1985 to 1994, a significant number of ES and DSS have been developed and used for insect pest management which include Stone et al. [78]; Coulson and Saunders [79]; Cervo et al. [80]; Pasqual and Mansfield [81]; Hoshi and Kozai [82]; Batchelor et al. [43]; Beck et al. [83]; Heong [84]; Foster et al. [85]; Kundu [86]; Olson and Wagner [87]; and Sing [88].

A prototype ES for diagnosis of potato disease based on various symptoms was developed by Boyd and Sun [89]. This ES was developed with the shell, PC Expert Professional (Software Artistry Inc.). Yialouris and Siderisdis [90] presented an expert system for tomato diseases identification. Many other ES and DSS for disease diagnosis have already been developed which include [80,86,88,91-94].

• Agricultural machineries

The agricultural sector is facing a severe shortage of labour and the available labour is becoming more expensive day by day because of the rapid industrial growth has caused a major shift of labour force from agriculture to industry. These changing circumstances have necessitated a speed-up in mechanization. However, in order for a farm to operate efficiently, the size or capacity and number of agricultural equipments should match the optimum power required by various sequences of cropping operations that must be performed within specific time periods. Larger machinery helps to reduce labour cost as well as timelines cost. For smooth running and perfect operation of these machines the owner of the machine should take proper care, should be familiar with the preventive maintenance and should have knowledge of common troubles, causes and remedies. So, usually, a qualified mechanic is required to solve the problem during breakdown which occurs repeatedly on the farm. Therefore, in such situation, ES/DSS approach can play a significant role.
Gaultney et al. [95] developed an ES for trouble-shooting tractor hydraulic systems. PC plus shell has been used to develop this ES. Bardale and Leong [96] developed an ES called, TRAPERT, which is able to diagnose most of the major failures in agricultural tractors and provide suggested remedy to the user.

Expert systems have been developed for selection of agricultural equipments. The PLANTING model was developed by Morrison et al. [97] to select planting machines for given soil properties and field conditions. Another tillage practice system was developed for southwestern Ontario by Clark et al in 1990 [98]. Hasbini et al. [99] described an ES which covers the domains of pump selection and operation of centre pivot system. This ES was developed as a practical decision aid for operational procedures for unskilled operators.

The TESTOP model was developed by Meyer et al. [100] to select a tillage system primarily based on crop yields. A DSS was developed by Butani and Singh [101] for optimization of farm machinery system with the flexibility to incorporate regional variations in crops and cropping practices, farm characteristics, sizes of farm equipment and costs of the resources and outputs. This DSS was developed using MS Quick BASIC and presents the optimum power solution for mobile and stationary operations, and select the optimum power sources and the matching equipment.

Singh and Pathak [102] described a DSS for mechanical harvesting and transportation of sugarcane in Thailand. This DSS aids in decision making in management of a chopper harvesting system for sugarcane. It also simulates the harvesting and transportation operation and determines the optimum number of trucks for a given set of transportation conditions. A DSS called PARMS (Planting and Residue Management System) was developed by Smith et al. [103]. The primary use anticipated for PARMS are the evaluation of methods of managing crop residue and planters for conservation tillage. The system was developed in C ++ rather than an ES shell, because the complexity of the problem required more flexibility than shell allowed.

- **Storage, marketing and transportation**

To enable farmers to get higher returns from their produce, it is inevitable to know about marketing patterns, storage of farm produce and its transportation. Serious losses occur due to the infestation caused by the storage pests.

Denne [104] developed an ES for stored grain pest management. A DSS for the provision of planting plans has been developed by Hamer [105]. The major objectives for development of this DSS was to market a steady supply of produce over along
period. In this system, a model was used to simulate the yield and timing of development of marketable buttons. The effects of time of sowing and/or transplanting were simulated. The model makes use of the NIAB (National Institute of Botany) recommended lists to simulate the varietal differences in growth patterns. A method for eliminating varieties which have poor sprout or plant characteristics have been presented. An LP model selects varieties and provides a planting schedule which satisfies the market demand and maximizes profit.

- Other ES and DSS related to agriculture

Besides the above mentioned ESs and DSSs, there are other ES and DSS developed for crop research facility, financial analysis, agricultural extension etc. [106-110]. In 1998, ECOZONE II, a DSS for aiding environmental impact assessment in agriculture and rural development projects in developing countries has also been developed and implemented [111].

3.2.3. Typical features of an expert system

In spite of the fact that each ES is unique in some sense, certain features are desirable for any ES. As knowledge intensive programs, the features of an ES are as follows [112]:

- An ES should solve difficult programs in a domain as good as or better than human experts. This is a fundamental criterion.

- Such a system should process vast quantities of domain-specific knowledge to the minute details. These are pieces of knowledge a human expert acquires after long years of professional experience in a field. These private pieces of knowledge being termed as 'heuristics' need to be incorporated into the system along with the conventional knowledge acquired from various sources.

- An ES permits the use of heuristic search process. An ES provides facilities for incorporating these heuristic search procedures.

- The system explains why they ask a question and justifies its conclusions. Explanation facilities enhance the credibility of the system in the mind of humans.

- An ES accepts advice, modifies, updates and expands. These characteristics, in a nutshell, form the basis for learning.

- The system deals with uncertain and irrelevant data. Like human experts, ES also have to deal with a lot of uncertain and irrelevant data.
The system communicates with the users in their own natural language. This is a characteristic that everybody is looking for. However, a few primitive natural language front end systems have developed which one can hook up with the ES.

An ES possesses the capacity to cater to the individual's desire. By this we mean that an ES can be used in different modes of operation.

The system provides extensive facilities for symbolic processing rather than numeric processing. Symbolic processing is the core of any AI program and hence an ES should provide facilities for doing so.

A final characteristic is from the point of economists and financial people. ES need heavy investment and there should be considerable Return-on-Investment (ROI).

3.2.4. Life cycle of an expert system

There are five major stages in the development of an ES [112]. Each stage has its own unique features and a correlation with other stages shown in fig. 3.1.

Stage 1: Identification of the problem. In this stage, the expert and the knowledge engineer interact to identify the problem. The major points discussed before for the characteristics of the problem are studied. The scope and the extent are analysed. The resources and finance are identified and estimated. The return-on-investment analysis is done.

Stage 2: Decision about the mode of development. Once the problem is identified, the immediate step would be to decide about the vehicle for development. The knowledge engineer can develop the system from scratch using a AI language like PROLOG or LISP or any conventional language or adopt a shell for development. In this stage, various shells and tools are identified and analysed for their suitability. Those tools whose features fit the characteristics of the problem domain are analysed in details.

Stage 3: Development of prototype. Decision on concepts needed to produce the solution is very important. One important factor to be decided here is the level of knowledge. Starting with coarse granularity, the system development proceeds towards high granularity. In this stage, the task of knowledge acquisition begins. The knowledge engineer and the domain expert interact frequently and the domain-specific knowledge is extracted. Once the knowledge is acquired, the knowledge engineer decides on the method of knowledge representation. When the knowledge representation scheme and the knowledge is available, a prototype is constructed. This prototype undergoes the process of testing for various problems and revision of the prototype takes place.
Stage 4: Planning for a full-scale system. The success of the prototype provides the needed impetus for the full-scale system. In prototype construction, the area in the problem that can be implemented with relative ease is first chosen. In the full-scale implementation, interactions with additional experts take place. Extensive planning is done.

Stage 5: Final implementation, maintenance and evaluation. This is the final life cycle stage of an ES. The full scale system developed is implemented at the site. The basic resource requirements at the site are fulfilled and parallel conversion and testing
techniques are adopted. The final system undergoes rigorous testing and later it is handed over to the user.

Maintenance of the system implies tuning of the knowledge base because knowledge, the environment and types of problems that arrive are never static.

Evaluation is a difficult task for any AI program. Solutions for AI problems are only satisfactory. Since the yardstick for evaluation is not available, evaluation becomes difficult. However, what one can do utmost is to supply a set of problems to the system and a human expert and compare the results.

3.2.5. Components of an expert system

Although at present there is no such thing as a standard expert system, but, however, 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. 3.2 and fig. 3.3) which are more or less self explanatory. 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 Logic, Semantic Networks, Frames, Rules etc. which will be discussed in chapter 4.

![Fig. 3.2. An architecture of a typical expert system.](image-url)
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 rules 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 intertwined 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 eliminating 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. 3.3 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.

3.2.6. Classifications of expert systems

3.2.6.1. 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 [114] 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. A comparison of rule-based reasoning vs. case-based reasoning has been shown in table 3.3. In the recent years case-based approaches are gaining momentum in different domains.
Table 3.3. Comparison of Rule-based and Case-based reasoning [115].

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

- **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. unconfirmed 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 hierarchical 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”.

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.

### 3.2.6.2. Based on other technical issues

Mentioning two major problems in building expert systems: (i) constructing and debugging knowledge base, and (ii) management of uncertainties might be relevant here. 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 [116] are shown in fig. 3.4.

Artificial neural networks [117-120] can be formally defined as massively parallel interconnections of processing elements that interact with objects of the real world in a
Fig. 3.4. Block diagram of the basic modules of various expert systems.
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 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 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 [121,122] 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 to handle real life recognition / decision making problems. For the last five to seven years, there have been several attempts [123-126] 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 [127]. 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 [128]. Fuzzy expert systems [128,129], 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 [130], and they use the set of connection weights of a trained neural net for encoding the knowledge base for the problem under consideration.

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. 3.4. As stated above, a fuzzy neural net constitutes the knowledge base of a neuro-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 justification 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.

A comparative analysis of the basic features of these models with those of the traditional and connectionist (non-fuzzy) versions is provided in table 3.4.

Table 3.4. Comparative study of various expert systems.

<table>
<thead>
<tr>
<th>Knowledge base</th>
<th>Knowledge acquisition and representation in the form of rules, frames, semantic nets or belief networks</th>
<th>Connection weights of trained neural net that were initialised with small random values</th>
<th>Connection weights of trained fuzzy neural net that were initialised with small random values</th>
<th>Connection weight of trained nonfuzzy / fuzzy neural net that were initialised with crude domain knowledge in rule form with binary link weights a priori class information and distribution of pattern points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge refinement</td>
<td>Addition of new knowledge (say, as new rules)</td>
<td>Empirical addition of hidden nodes/links</td>
<td>Empirical addition of hidden nodes/links</td>
<td>Network optimization using growing and pruning of nodes/links, based on training data and additional knowledge</td>
</tr>
<tr>
<td>Inferencing</td>
<td>Matching facts with the existing knowledge base</td>
<td>Presentation of crisp input, forward pass and generation of crisp output</td>
<td>Presentation of fuzzy input, forward pass and generation of fuzzy output</td>
<td>Presentation of input, forward pass and generation of output</td>
</tr>
<tr>
<td>Rule generation</td>
<td>Crisp rules obtained during backward pass using changes in levels of input and output units, magnitude of connection weights</td>
<td>Fuzzy rules obtained during backward pass using node activation and link weights</td>
<td>Rules obtained during backward pass; negative rules also possible</td>
<td></td>
</tr>
</tbody>
</table>
3.3. 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 can be found in chapter 1 and chapter 2. Proper and prompt identification and control of tea insect pests and diseases are highly required. As an ideal case, at least one entomologist/plant pathologist should be placed at each tea garden. But, as the tea cultivation is spread over a wide area, human experts are really a scarce commodity. To mitigate such needed expertise, an automated knowledge-based consultation system would be helpful. 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 [131-133]:

- 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 multi-criteria decision making (MCDM) [134] 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 for quite sometime.

The knowledge associated with the domain is large and varied. Reaching a conclusion with such 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 plant insect and plant disease interactions 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 of the domain are available here. 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. Knowledge of this problem domain is vast and varied and hence 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 "deformed" or
"stunted" or "defoliated" (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. From the economic point of view, no better solution, seems feasible. It may not be possible to appoint even one human expert on insect pest and disease per tea garden. The needs of the domain may be fulfilled, at least partially, by the present system under investigation. It is expected that the full system once achieved will partially; be used as long-term basis, an important pay off.

3.4. 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 tea gardens. 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/Windows. 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. It should be easier then for tea garden workers who already have hardware 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 agricultural domain this is more critical. An expert system should be capable of handling these inexact situations.

- **Transparency**

For firm identification and diagnosis as well as for further course of action(s), an expert 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, 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 tea garden workers and management trainees who are not experts in entomology or plant pathology. This particular feature should assist them to view the chain of reasoning leading to a conclusion. This chain of reasoning should certainly assist a worker for further analysis and control planning. This chain of reasoning should also be useful to non-monotonic 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 agricultural 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. This is essentially a learning facility with the system. 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 non-monotonic 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 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, the level of abstraction is expected to be high for the ease of proper identification and 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 4, a detail discussion has been provided on the knowledge representations schemes along with their relative merits and demerits.

3.5. Discussions

After a brief introduction to AI and expert systems technology, categories and application areas of expert systems with some examples have been provided. 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 discussed 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 (e.g. medical, agricultural etc.). For the
generation of more intelligent decision making systems some researchers propose fuzzy systems, some propose neuro-fuzzy models, some propose knowledge-based 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.

In our present study we have explored the development of a rule based object-oriented knowledge based system for the insect pest and disease domain and as well as case-based reasoning for the development of our models for case-based learning and case-based classifier approach.

References


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


