
The chapter covers the expert system principles, elements, architecture and its characteristic, distinction between expert system and other algorithmic, representation of knowledge in expert system - rule based, object based and logical based, development of an expert system, the role of expert system shell and some of the available expert system shells.


5.1. Introduction

The chapter is focused to discuss about the fundamental principles, the architecture and advantages of an expert system. This chapter also covers the appropriate area of application of the expert system. The relationship of expert system to other methods of programming vis a vis the representation of knowledge in the expert system and expert system shells for their development is put forward in the chapter.

Expert Systems are computer programs which help to solve multifaceted setback decision in a specific domain that simulates the thinking process of a human expert. Several years are expected to continue its progression in development of expert systems with new exciting applications. It operates as an interactive system by responding; clarifying to questions being asked and makes a recommendation which generally aids the decision-making process. Expert Systems provide expert instruction and direction in extensive range of conduct from computer diagnosis. Several authors have assorted different definitions of expert system. Generally it represents that the intended functions of expert systems as: an
interactive computer-based decision tool that uses both facts and heuristics to solve difficult decision problems based on knowledge acquired from an expert [26].

Expert Systems are an emerging technology with numerous areas for impending applications such as: interpreting and identifying, predicting, diagnosing, designing, planning, monitoring, controlling, debugging and testing, Instructing and training [65]. In order to configure computer systems, MYCIN application of expert system was used to diagnose infectious blood diseases in the medical field and it has proof to be reasonably success.

Traditional decision support systems of solving problems such as spreadsheets are very mechanistic that operate executively under mathematical and Boolean operators and arrive at only one static solution for a given set of data. Traditional decision support tools or conventional programming are better handled by calculation intensive applications with very exacting requirements. Computational or deterministic natures of expert system are not good candidates for applications.

Those dealing with expert heuristics for solving problems are considered to be the best candidates. Humans work out problems on the basis of a mixture of factual and heuristic knowledge. Conventional computer programs are based on factual knowledge, an
indisputable strength of computers. The undeniable strength of humans is the heuristic knowledge which composed of intuition, judgment, and logical inferences. Successful expert systems are those that combine facts and heuristics and thus merge human knowledge with computer power in solving problems. Therefore effectiveness of the expert system mainly depends on the particular problem domain being focus.

5.2. Principles of Expert System

The main principle of expert system is operating of system by supplying the facts and discrete information to the system through user interface and in response to that the system gives the considered necessary advice as an expert. Expert System consists of two main components: knowledge base and the inference engine as illustrates below in Figure 5.1. The knowledge base contains the knowledge from which inference engine draws conclusions which are expert system’s responses to the user’s queries for expertise [66].
Some systems have been designed to act as an intelligent assistant to a human expert because of the development advantages with expert system's technology. In addition of more knowledge, it enhance further intelligent system which consequently act more like an expert and helps to speed up the expert's time in finding the solutions of problems. It is, therefore, a useful milestone in producing a complete expert system by developing an intelligent assistant.

The relationship between the problem domain and knowledge domain is an important factor to understand in an expert system. The expert’s knowledge area about solving specific problems is called the knowledge domain of the expert which is specific to one problem domain. Problem domain is the special problem area that an expert can solve correctly such as: medicine, finance, science, or engineering and so forth. Human experts are generally designed to
be expert in one problem domain. The expertise in a problem domain does not automatically carry over to another. By taking an example, a medical expert system designed to diagnose infectious diseases will have a great knowledge about symptoms caused by infectious diseases. In this case, the knowledge domain is medicine and consisting knowledge about diseases, symptoms and treatments. Thus, all knowledge domains are usually presented within the problem domain [52].

![Diagram showing the relationship between problem domain and knowledge domain](image)

**Figure 5.2: Problem domain and Knowledge domain relationship**

The diagrammatic representation shown in the above figure is the relationship between the problem domain and the knowledge domain. The knowledge acquired in most expert system are usually obtained from published materials and directly from the experts in the field that the system is being built.
5.3. Domain Specificity in the Expert System

Expert Systems are classically a domain specific knowledge base that combines with an inference engine which actually provides recommendations in a specific task domain. The data stored in the knowledge base is process to respond to the user’s request for advice. Expert System uses the domain specific knowledge of the human expert and the inference techniques, thereby simulating the problem solving behaviour of the expert in the required field. The inference engine combines the facts of a specific case with the knowledge contained in the knowledge base. In case of the rule based expert system, inference engine controls the order where production rules are applied. For example, a diagnostic expert system for troubleshooting computers must actually perform all the necessary data manipulation as a human expert. Developer of such system must limit the scope of the system to what is needed so as to solve the target problem. Special tools or programming languages are often needed to accomplish the specific objectives of the system [19].

5.4. The Expert System Architecture

To understand the functions of expert system, it is very necessary to look the architecture of the expert system and to examine the different components that contribute to present the
expert's knowledge in such a system as shown below in figure 5.3 which mainly highlights the important components of an expert system such as: the knowledge base, the knowledge database, the inference engine and the user interface. Other components that can be part of the expert system architecture are the knowledge acquisition process and its components which are described in detail in the chapter 6.

![Architecture of the Rule Based Expert System](image)

**Figure 5.3: Architecture of the Rule Based Expert System**

**Knowledge Base**: The knowledge base is the component of an expert system that contains the facts about the subject which is being dealt with. The facts could be presented in the form of rules and sub rules. For example, the potential contents of a knowledge base for project JESS based Expert System for Rice Plant Disease Diagnosis: Ph.D. Thesis: Kh. Robindro Singh: Department of Computer Science, Gauhati University, 2013 110
tendering would include related facts about the project location, size and complexity of project, contingency plans available, etc. Since knowledge is continually changing and expanding it is considered to be important that the knowledge base is clearly structured and can easily be modified if required to do so.

Knowledge Database: It contains all the pieces of information the rule-based system is working with. It can hold both the premises and the conclusions of the rules. Typically, the rule engine maintains one or more indexes, similar to those used in relational databases, to make searching the knowledge database a very fast operation.

Inference Engine: An inference engine activates the rules from the knowledge base on the data, within the knowledge database, and for the solution of the problem. The inference engine usually employs a mechanism that matches information in the knowledge database with pertinent action rules in the knowledge base, and if several rules apply, it selects the most appropriate course. It then implements the selected action. There are two categories to arrive a conclusion in the expert system that matches the rules in the knowledge base with the facts presented in the knowledge database. The categories are:

i. Forward Chaining, where the inference engine starts with facts and matches them to the conditions of a rule. If the condition is satisfied, the rule's conclusions are used to prove additional
or further rules. This continues until sufficient rules and facts are established to make a conclusion.

ii. Backward Chaining, in which the inference engine processes rules by examining the rule's conclusion first and then its premise. The inference engine selects a rule with a conclusion that directly solves the problem. It then tries to determine whether the rule's premise is true or false. If the premise is false the engine selects another rule. If the premise is neither true nor false (not enough rules or facts have been examined to determine this), the engine selects another rule with a conclusion that could solve the premise.

The User Interface: The expert system user interface is the component which is responsible for the communication mechanism between the user and the system. In addition to being highly interactive, an expert system interface requires a transparency of dialogue, whereby some form of an explanation facility indicates the inference process that is being used.

5.5. Advantages of Expert System

The expert systems have many advantages [67]. Some of the advantages of the expert systems are:
i. Homogeneity: Because of the uniform syntax, the meaning and interpretation of each rule can be easily analyzed.

ii. Simplicity: Since the syntax is simple, it is easy to understand the meaning of rules. Domain experts can often understand the rules without an explicit translation. Rules therefore can be self-documenting to a good extent.

iii. Independence: While adding new knowledge one need not be worried about where in the rule base the rule is added, or what the interactions with other rules are. In theory, each rule is an independent piece of knowledge about the domain. However, in practice, this is not completely true, as we shall see in the next section.

iv. Modularity: The independence of rules leads to modularity in the expert system. We can create a prototype system fairly quickly by creating a few rules. This can be improved by modifying the rules based on performance and adding new rules.

v. Knowledge is separated from User and Control: The separation of the rule base from the inference engine separates the knowledge from how it is used to solve the problem. This means that the same inference engine can be used with different rule bases and a rule base can be used with different
inference engines. This is a big advantage over conventional programs where data and control are intermixed.

vi. Procedural Interpretations: Apart from declarative interpretation, rule based expert systems have procedural interpretations also, which enable them to be viewed as computational models.

5.6. Explanation Facility in Expert System

The most important characteristics of an expert system are the rationalization facility. It influences the user confidence and acceptance of ES-based decisions and recommendations. Expert System then imparts the user with a trace of the consultation process, pointing out the source of reasoning paths during the consultation. The user can pose different questions of dealing with what, how, and why aspects of a problem arise. With this capability, an expert system can explain how it arrives at its conclusions. Sometimes, expert systems are required to solve other problems which is not directly related to the specific problems but that solution bring a drastic impact to the entire problem solving processes. The explanation facility helps the expert system to clarify and justify why such digression might be needed.
5.7. Integrating Expert System with Other Related Fields

The expert system technology can be integrated with other related fields that include decision support systems (DSS), simulation, and data repositories. The objective of such integration is to expand and enhance the applicability and usability of ES. The scenarios for such integration are hereby discussed. Firstly, ES is used as an intelligent front-end to a DSS that assist the decision-maker in selecting an appropriate decision model. Secondly, it integrates ES and simulation which are two closely related fields, one front-end and another back-end approaches that increase the functionality of both disciplines. Finally, it describes the integration of ES to spreadsheets and databases. So, ES may import data and/or store data in these two types of data repositories [55].

5.7.1. Integration of an ES and a DB

Significantly, it is noted that despite of fact present historically, expert systems in artificial intelligence and DBs have been developed separately. Nevertheless, many application domains where, simultaneously both access to an industrial DB. Uses of an ES for decision making on the basis of experience or expertise are required. In association to this, problems of integrating static ESs and DBs have been investigated the best. Therefore, many publications are devoted to a specialized application of Integrated ESs that combines
the techniques of KBs and DBs. However, most authors have concentrated their efforts on the description of ways of implementing peculiarities of one or another application domain which are substantially paid less attention to the principles, methods, and models of integration in the framework of ESs and DBs. In this connection, the main emphasis place only on the papers is of interest from the standpoint of methods of joining ESs and DBs [68, 69].

There are two approaches to joining ESs and DBs - weak and strong coupling. Strong coupling is used when there is an opportunity to split the operation of an ES into a sequence of steps that require definite predictable and limited information from the DB. For example, in tasks of medical diagnosis, the information about a particular patient can be requested before the beginning of a consultation. As a result, the opportunity to efficiently arrange the operation of the ES is a merit of this approach, since in the course of consultations of the ES; there is no need to query the DB.

However, in other application domains, this approach may be impossible and a strong coupling of the ES and DB may be required. This coupling allows one to multiply read and modify the content of the DB. It is quite obvious that a strong coupling imposes more serious requirements on the system; in particular, the rate of transactions between the ES and DB supposes an efficient
implementation of this functional component. Among the first Russian systems of this class, the shell for the DI*GEN ES that has a mechanism of strong coupling of the ES and DB is worth mentioning. To implement this mechanism, DI*GEN contains a set of functional primitives that allow one to position the DB, to read and modify data from an arbitrary field of any record of the DB, to add/delete any records in the DB, retrieve data using a condition, etc.

A radically different approach to integrating ESs and DBs was implemented in one of the earliest Russian applied ESs, the integrated intelligent system of the molecular design of physiologically active substances, which was also hybrid, since it employed several reasoning strategies and models of knowledge representation. From the standpoint of the technique for joining an ES and a DB, the characteristic specific feature of this system is the use of an original relational DBMS for generating training samples and storing data, since the decision strategy in this IES is based on two complementary concepts of inductive learning by examples.

5.8. Expert System and Other Algorithmic Programs

The difference between expert systems and algorithmic program lies in the use of knowledge. Expert System separates the program into an explicit knowledge base, the problem solving strategy and a control program, an inference engine, which
manipulates the knowledge base. Traditionally, algorithmic application is organized into data and program. Data context describes the problems which are being solved and state the current status of the solution process. Such an approach is referred to as knowledge-based [70, 71].

Knowledge-based expert systems are especially designed based on the same types of premise-action rules as algorithmic programs design in order to overcome the shortcomings of algorithmic computer applications. Symbolically, the rules are represented and used by the knowledge processing component of the expert system. Prior to the problem solving, the exact sequence of selecting and applying those rules are not stipulated. It helps to invoke actions based on the rules that apply to a particular situation. Additionally, expert systems have the capabilities for selecting rules based on incomplete or uncertain data. It also helps in explaining why rules are selected and how they are being used.

In the domain expert of a given program, the applicable conditions and corresponding actions can be defined by only knowledgeable person. It is practically true, where a very large proportion of the rules are not necessarily based on the causality of physical laws, as in any engineering program, but it represents heuristics - assumptions, limitations, rules of thumb or style of the
expert(s). Developing a complete set of rules is a major undertaking for any engineering application where the rules must satisfy the three criteria: *completeness, uniqueness* and *correctness*.

The applications may be easily and effectively developed as algorithmic programs based on mathematical models and those that require intense mathematical computations. Many problem solving situations, such as interpretation or design, are too ill-structured and less suited to the rigid algorithmic format [57]. Numerous engineering tasks fall into this category. The potentialities for applications of expert systems to such tasks are discussed in the following sections.

The important features of expert systems when compared to other mathematical models are pointed out by Jackson as [72]:

- Expert systems are not confined by rigid mathematical or analogue schemes and can handle factual or heuristic knowledge.
- The *knowledge base can be continuously augmented* as necessary with accumulating experience. As the knowledge advances the expert system can be easily upgraded to cope with the changes in technology;
- Its *ability to handle qualitative information* can be clearly experienced if the system to be built is expected to contain a
large amount of information as for example in many management areas.

- **Coping with uncertain, unreliable or even missing data** and with uncertainty in data and inference is a feature of expert systems. When pieces of the knowledge base and context are less than certain, then a new level of complexity is introduced into expert systems. Some expert system shells have adopted Bayesian probability in coping with uncertainties within the knowledge base.

- **The reflection of decision patterns of the users** is a function which is sometimes referred to as an explanation facility. It explains the reasons behind giving a certain conclusion that has been reached by the expert system.

### 5.9. Knowledge Representation

Knowledge representation involves finding the best method to use to present the domain knowledge as data structure. This is considered to be an important step so that the knowledge can be efficiently accessed for the problem solving task. The three basic requirements on a representation scheme in an expert system are extendibility, simplicity and explicitness [73].

- **Extendibility**: In an expert system, the knowledge base can be extended without forcing significant revisions. The data
structures and access programs must be flexible enough. Knowledge bases built out of experts’ experience will contain heuristics, thereby helping the experts to remember and in modifying the previous ones. The possibility of an expert to define a complete knowledge base is very rare but a subset and refining its consequence problems over weeks or months are possible. However, the most effective methods for building a knowledge base are by incremental improvement. All this argues are mainly for treating knowledge base of an expert system as an open-ended set of facts and relations, and keeping the items of knowledge as modular as possible.

Simplicity: The arrangements of data are so elaborate so as to be impenetrable, and unchangeable. The tractability maintained for above requirements conceptual simplicity and uniformity so that access routines can be written. The syntax of the knowledge base is once prepared, the approach routines can be prepared to a large extent. Knowledge acquisition takes place with the expert segregated from the data structures by access routines to make the knowledge base perform simple. New causes will in spite of this come in to view for retrieving the knowledge base as details of the contents of the knowledge base or analysis of the links among items, display. Simple data structures pay large benefits with these reasons. There are two
ways of maintaining conceptual simplicity from designer's point of view: keeping the form of knowledge as consistent as possible or making special access functions for non-uniform representations.

The simplicity that comes from using roughly the same terminology as the experts use is another sense of simplicity that needs to mention. Programmers often find alternative ways of representing and coding ingeniously what a specialist has requested. It is a fact that sometimes it makes in processing more "efficient" but also makes nightmare in modifying the knowledge base.

**Explicitness:** Representing an expert's knowledge is to give the system a sufficient knowledge base for high-performance of problem solving. A knowledge base must be built incrementally because it is necessary to provide means for inspecting and debugging it easily. With items of knowledge represented explicitly, in simple terms, the experts who are building knowledge bases can determine what items are present and absent.

There are three types of representation framework have been used in expert system to achieve extendibility, simplicity and explicitness goals. We concentrate on research & development based systems although, we also mention frame-based and logic-based systems by
way of contrast. One can express conditional expressions and causal dependencies in all of these languages. One sacrifices the ability to express many other general kinds of relations in favour of the homogeneity and simplicity of conditional rules in rule-based systems. These frameworks include inference and control routines making them even more like languages.

Knowledge Representation is an area of artificial intelligence research aimed at representing knowledge in symbols to facilitate inference from those knowledge elements, creating new elements of knowledge. Knowledge representation involves finding the best method to use to present the domain knowledge as data structure. This is considered to be an important step so the knowledge can be efficiently accessed for the problem solving task [18].

5.9.1. Rule Based Representation

Most rule-based systems can be classified as production systems. The main element in this system is that the domain knowledge is represented in modular rules. Production rules are the most comprehensive form of knowledge representations which are based on condition-dependent action. It has become apparent that experts can best formulate their knowledge in the form of 'if ..... then .....' rules. This is probably the reason why most expert systems
today, at least the most successful ones, are based on production rules. An example of production rule languages is OPS5.

5.9.2. **Object Oriented Representation**

Object oriented representation is about grouping objects by organising them into classes. One obvious advantage of classification is that it eases memory load as only the characteristics would be considered and not each individual object. Also it defines relationship between those classes. The fundamental organising principle in such systems is the packaging of both data and procedures into structures related by some form of inheritance mechanisms. The process of expanding a knowledge-based system is one of incremental programming; simply tell the system about more objects, and as long as they are placed at the right point in the hierarchy. Inheritance would always do the right thing. One of the object-centred languages is FLAVORS which is embedded in ZETALISP [72].

5.9.3. **Logical Based Representation**

A logic-based representation is one in which knowledge about the world is represented as assertions in logic, usually first order predicate logic or a variant thereof. Logic based languages allow quantified statements and other well defined formulas as assertions [73].
A logic database for representing knowledge about some domain is typically a set of clauses, ordered so that special cases occur first. An example for logic programming languages is PROLOG. Logic programming techniques for expert systems are by no means as tried and tested as techniques based on rules and frames but it can be observed from the attempts to implement it that it does not automatically abolish all pitfalls associated with frames and rules. It needs an experienced programmer to structure rules and data according to a logic programming language.

5.10. Expert system Shells

There are several approaches to building expert systems in terms of what kind of tools to use. These tools are normally classified as: languages, environments or shells [74]. Languages can be either special purpose languages for symbolic programming, such as LISP or PROLOG, or a conventional one, such as PASCAL or C. Environments contain various types of knowledge representation, inference mechanisms, user interface and development aids. These tools also give access to the underlying language the environment is written in. This enables the developer to incorporate special tasks. KEE, ART and EDSS are all environments. Shells provide a more specific set of knowledge representation languages and inference mechanisms, geared to handle a particular class of problems. An expert system
shell is a software development environment containing the basic components of expert system. Both shells and environments differ from the languages in the fact that they already contain control mechanisms that determine how they reach conclusions.

5.10.1. Types and Features of Expert System Shells

According to Terry, the types and feature of the Expert System Shells are listed and discuss as follows [75]:

- **Built-in inference engine with choice of chaining methodology**: Some shells mixed chaining is considered to be a useful feature, as it helps the knowledge engineer to either put constrains at the start of the rules or at the end depending on what is required to solve a problem.

- **Menus and prompts for development and testing**: Here, menus must be easy, clear and contain all the functions needed to build the required expert system to use. It is important that the knowledge engineer must become familiar with the shell menus and prompts in order to find out whether it will be able to perform the job in hand.

- **Debugging and value-checking aids**: The function of this approach will deal with the ability of the shell in the debugging of any input errors, which could be contained in the information and values.
**Consistency checking:** Here the function will check that the rules do not contradict each other, as this could lead to false assumptions or wrong conclusions.

**Rule - prioritise capabilities:** It is the solving of a particular problem, a group of rules which relate more closely to the problem will be considered first. By matching the rules to the problem in hand the correct rule to solve the problem will be chosen.

**Interface with other operating systems and software:** The shell must have all the required interfaces to assist in building the expert system. These interface can either be supplied as part of the shell or it must have the ability to communicate with other software, whether by direct access or by exporting and importing functions. The shell should also have the ability to create and display graphics or interface with graphics software programs.

**Mechanism for handling input and calculation of uncertainties:** This function will allow the shell to deal with missing or uncompleted data which could be inserted by the user in the form of input.

The explanation facilities function will allow the user to find out why and how the system arrives at its decisions, with the sequence of rules involved.
5.11. Suitability and Feasibility of Expert System in Agriculture

The problems raised in the field of agriculture can be solved up to a large extent with the help of expert system to generate the information for the growers by using the knowledge base and reasoning mechanism acquired from human experts and other resources.

Expert Systems are computer applications, which encode expert knowledge that assists the users in making decisions. Expert knowledge is collected from many sources, including human experts in the domain, research results, and government policies. The expert knowledge is then embodied in simulation models, expert systems, hybrid systems, databases, or spreadsheets. Unfortunately, many of these expert systems do not adequately explain and justify their recommendations. Agricultural expert systems must be able to make recommendations that the user can easily understand, trust, and apply to his or her particular situation.

The agricultural industry today is complex, dynamic, and competitive. Many factors, such as biology, weather, economics, marketing, legal constraints, and social responsibilities interactively influence this business. Agricultural technology is constantly changing.
the way farm managers operate their farms. Today, farmers are swamped with many new cultivars, pesticides, and farming techniques. Coping with this ever-growing complexity is overwhelming. In order to make efficient and accurate decisions, farmers must be agronomist, entomologist, plant pathologist, soil scientist, horticulturist, and economist rolled into one.

Agricultural expert systems, which embodied such expert knowledge, are a promising means of distilling information down to only that which the farmer needs. Although expert systems are an obvious solution, farmers are reluctant to adopt these systems because they are too complex and do not use the terminology and logic that is familiar to the users.

Expert Systems are knowledge-based systems that use the factual knowledge, procedural rules, assumptions, and heuristics of an expert to perform a specific task. Reasoning and domain knowledge are explicitly represented in expert systems, enabling conclusions to be readily explained. However, rule-based expert systems tend to lack quantitative precision since they are not designed to efficiently carry out numeric simulations. The inference mechanism involving rule-firing does not lend itself to computation intensive applications. Despite recent advances in uncertain reasoning, expert systems typically fail to reason effectively with
noisy and uncertain data, especially in domains with large quantities of data [76].

Expert System helps to generate information to the farmers by using its knowledge base and reasoning mechanism from human experts and other sources. There is not as such limitation on the number of generated recommendation. When a user enters the data of his plantation to the system, appropriate advices are generated based on its knowledge base and reasoning mechanism. It can solve the above said problem to a large extent. Therefore, it overcomes the problem of static information provided in extension documents.

The integration of knowledge and experiences of different specialists are facilitated on the expert system. When problems occur related to the nutrition, plant pathology, entomology, breeding and production, the expert system can help the users in identifying the cause of the problem in an efficient way, thereby consulting a document that handles the specific problem.

Expert System makes easier for user to identify the describe symptoms like image bases or textual bases information as it is very difficult to describe in words. It can also be integrated with textual database which can be used for explanation purposes of basic terms and operations to confirm and to reach conclusion in some situations.
Confirming the diagnosis of the rice plant and the cause of certain disorder can also be done through image suitability.

The knowledge base can be maintained more efficiently than maintaining manual documents. Updating problem is also done in expert systems. However, the problem of updating the versions in the field can be eliminated in case that the expert systems are stored on a central computer and accessed through a computer network.

Expert System can also help in overcoming the problem of the relatively few numbers of experts relative to the demand from the farmers. Expert System technology can help to transfer the information of experts, and experienced farmers to farmers through the extension system [35].

The applications of expert system in agriculture are mainly found in the area of rice plant diseases diagnosis and pest controls. In agriculture, many domain specific expert systems are being used at different levels. Shikhar Kr. Sarma, Kh. Robindro Singh & Abhijeet Singh developed an expert system in agriculture with rule-based expert system, using ESTA for diagnosis of diseases in Rice Plant and its application [77]. Another similar study on developing an expert system for plant disease diagnosis is also an application of expert system in agriculture. The system is for the purpose of plant disease control using the shell CLIPS by S.S. Abu-Naser, K.A.Kashkash and
M. Fayyad [78]. Ahmed Rafea conducted a study on Expert Systems Applications: Agriculture. It is also the application of expert system in the agriculture domain development [35]. “Web based Expert System for Diagnosis of Micro Nutrients Deficiencies in Crops”, by S.S. Patil, B.V. Dhandra, U.B. Angadi, A.G. Shankar, and Neena Joshi also describes application of expert system in agriculture particularly in the area of nutrient deficiencies in crops. The system is a web based system using the ServCLIPS tool [79].