Title: Feasibility, Impact, and Future of Expert Systems

Course:
Year: 1988
Author(s): R. Bhatia, M. Fenrick , M. G. I. yigun, B. Pathak, E. Salgado and A. Tor

Report No: P88006

Abstract: As Artificial Intelligence matures, it is likely to spawn major technological changes in society. Many authors believe that the technological changes brought forth by AI will be as profound as the change initiated by the introduction of the transistor in the 1940's and the microprocessor in the early 1970's. Some of the technological changes will come from the utilization of Expert Systems (E.S.). This report examines expert systems with emphasis on the organizational, economic and social issues related to the E.S. technology.

Feasibility, Impact and Future of Expert Systems

Rohit Bhatia, Michael Fenrick, M. Guven Iyigun
Bimal Pathak, Elizabeth Salgato, and
Abdulmennan Tor

P8806

FEASIBILITY, IMPACT and FUTURE of EXPERT SYSTEMS

EAS 541

Rohit Bhatia
Michael Fenrick
M. Guven Iyigun
Bimal Pathak
Elizabeth Salgato
Abdulmennan Tor

Feasibility, Impact and Future of Expert Systems

EAS 541

TABLE OF CONTENTS

1.0 INTRODUCTION

The field of artificial intelligence (AI), a subject of active research over the last 20 or so years, is growing rapidly and is likely to mature as a science in the near future. As AI matures, it will spawn major technological changes in our society. We believe that the technological changes brought forth by AI will be as profound as the changes initiated from the introduction of the transistor in the 1940's and the microprocessor in the early 1970's. Some of the early technological changes resulting from AI research will come from the utilization of Expert Systems (ES). The use of Expert Systems has some very important organizational, economic and social implications. It is the purpose of this paper to attempt to examine such issues and thus provide a better understanding of expert systems technology.

1.1 DEFINITION OF ES

An Expert System is an intelligent computer program with a tremendous amount of high quality specific knowledge in a particular problem area with the ability to make inferences and provide the reasoning behind its conclusions. Although ES are also computer programs, they differ from conventional computer programs in the following aspects [52]:

- ES programs mostly perform symbolic processing whereas conventional programs perform numeric processing.
- ES use heuristic search methods to arrive at solutions while conventional programs use an algorithmic approach.
- The control structure is usually separate from the knowledge base in ES. In conventional programs, the data and the control are usually integrated.
- In an ES approach, satisfactory and sometimes even incorrect answers are generally acceptable. In conventional programs correct answers are required and the best solution is sought almost always.

1.2 STRUCTURE OF ES

There are primarily two different ways of organizing ES : 1) Rule-based and 2) Frame-based. Rule-based ES are composed of five parts [14]:

1) Knowledge Base - This is the nucleus of an ES and accumulates during system building. It consists of heuristics about a particular problem domain. The knowledge base can serve as an institutional memory. This compilation of knowledge becomes a permanent record of the expertise of the contributing experts within a company. This can be a very desirable feature for business managers in an industry prone to rapid turnovers.

2) Data Base - All facts and numeric data associated with a problem domain are stored here. Some AI researchers consider this too to be a part of the knowledge base.

3) Inference Engine - The inference engine is a reasoning mechanism to interpret and apply the knowledge in the knowledge base. It monitors the data base and the knowledge base and executes rules as a user enters facts about a problem to arrive at a solution. There are two kinds of inference mechanisms :

i) Forward Chaining - In this scheme, the inference mechanism executes rules in a particular order in response to a stimulus from the environment. Inferences are actively drawn from available facts and rules.

ii) Backward Chaining - Also known as a goal driven approach, the backward chaining process fires rules in a certain order to achieve a specified goal. In other words, a goal is selected and the engine checks to see if the supporting facts exist or can be inferred to verify the goal.

4) Explanation Mechanism - The presence of this facility in ES is what really distinguishes them from conventional programs. The explanation tool allows an ES to trace back its reasoning and substantiate the solution it derived and thus allow the user to verify correctness of results. It is widely believed that the explanation facility of an ES will be a major factor in the acceptance of ES by end users.

5) Set of Control Strategies - The control strategy for an ES is described by what are called metarules. Metarules can simply be described as rules about rules. The control mechanism or metarules direct the inference engine in its rule selection and conflict resolution. The metarules help in organizing a search strategy, given a goal, in the huge knowledge base of an ES and thus improve its performance.

Due to their structure as stated above, ES have a number of desirable features. Two of these features, the institutional memory feature and the high level of expertise i.e. the knowledge base have already been mentioned above. The inference engine and the separate control mechanism allow ES to be used as predictive modeling tools. Different scenarios of the future can be created by modifying the meta rules and the goals sought by the inference engine. Also different facts can be furnished to the inference engine to simulate a desired scenario and thus analyze the consequences. The explanation facility allows an ES to be used as a training facility for new employees. With the addition of an user friendly interface, an expert system can be readily adapted as a cost effective training vehicle.

1.3 BENEFITS OF ES

Expert Systems can provide a lot of benefits to users. Potential benefits are [48]:

Cost reduction: Human expertise is very scarce, hence expensive. On the other hand, expert systems have high development cost but nominal operational cost. Hence once developed, they are inexpensive compare to human expertise.

Permanent expertise: Human expertise is prone to percentage of gross domestic product has remained relatively constant despite increasing use of machinery. From this we can infer that the share of labor in the monetary value of production has not declined as might be expected in the last 40 years [56]. The primary reason for this fact is that as production technology has advanced, so has the level of skill required of labor and thus resulted in higher employee compensation. As a consequence, we now have a different type of work force, namely one that is more professionally trained.

However, this trend cannot continue indefinitely, for a number of reasons :

- availability of skilled workers
- finite time lag in the training of skilled workers to the desired skill level
- limited life span of humans

Management science practitioners propose two ways of combating this trend. Both specify that technology be used itself to enhance the productivity of a skilled worker as production technology advances. The first approach, using the assumption that productivity increases with work experience, suggests use of such tools that allow an inexperienced worker to perform at the level of an expert. The second approach suggests development of new tools that allow the same skilled worker to increase his productivity without requiring additional training. In essence both approaches are advocating the development of computer tools that would increase the productivity of a processional worker. Several MS practitioners believe that this is a task for ES. However they emphasize that such tools should not be thought of as replacement for the skilled human workers but as assistants that enhance the worker's output by assisting in those domains where a machine has definite advantages over humans [56].

1.6 APPLICATIONS OF ES

In recent years, Expert Systems have been built to solve many different types of problems in various fields. Their basic functions can be grouped into the categories shown in Table 1.1 [52].

Table 1.1 Generic Categories of Expert System Applications.

Category	Problem Addressed
Interpretation	Inferring situation description from collected data.
Prediction	Inferring likely consequences of given situations
Diagnosis	Inferring system malfunctions from observations.
Design	Configuring objects under constraints.
Planning	Designing actions.
Monitoring	Comparing observations to expected outcomes.
Debugging/ Repair	Prescribing and administering remedies for malfunctions.
Control	Governing overall system behavior.

Expert systems usually perform some form of data interpretation. The data can be from meter reading or a sensor device. The system SPE[52], interprets waveforms from a scanning device to distinguish between different causes of inflammatory conditions in medical patients. Interpretation systems may process different kinds of data. For example, voice and pattern recognition systems use natural inputs to infer features and meaning.

Prediction based expert systems infer the likely consequences of given situations. Estimating global oil demand or predicting the damage to crops from some type of insects are some of the examples of these types of systems. These systems rely heavily on simulation models that mirror real-world activities to generate practical situations and scenarios.

Most of the systems in use today are probably diagnosis based expert systems that infer probable cause of system malfunctions. Diagnosis systems are used to diagnose problems and consult for possible remedies. The medical domains is a natural for diagnosis applications, and indeed, more diagnosis based systems have been developed for medicine than for any other problem area. The MYCIN[52] system is an example of quite successful diagnosis based expert system. It is used to diagnose bacterial infections in hospital patients.

Expert systems that perform design develop configurations of objects based on set of constraints. XCON[52], a rule based design systems that configures VAX computer systems according to user's specification is an example of designed based expert systems.

Planning systems perform planning or decides on an entire course of actions before acting. These systems usually require some form of backtracking, that is, reject a particular line of reasoning because it violates problem constraints, and restart reasoning from an earlier point in the solution of the problem. This backtracking effort can be expensive.

Monitoring systems compare actual system behavior to expected behavior. Examples are monitoring instruments reading in a nuclear reactor to detect accident conditions and assisting patients in an intensive care unit by analyzing his vital statistics. The system, called REACTOR[52] has been used in nuclear reactors for monitoring instruments readings for signs of and accident.

These are few of the existing expert systems in various problem domains that have been developed already, and there are many more not described here. As AI research continues, the scope of expert systems will broaden and will impact all aspects of socio-economic infrastructure.

OVERVIEW OF PAPER

The above sections have outlined a definition of ES, justified the need for ES and stated their benefits and limitations. In addtion applications of ES have also been briefly described. Having laid the basic groundwork here, the next section explores the issues in crafting ES. The building of a large ES is similar to any other engineering project and requires careful planning. The next section details the steps involved in the design of ES right from building the development team to delivery of the system. Section 3 presents five case studies of ES currently in use in industry. Each case study briefly describes the ES and its intended application. Then Section 4 discusses the social impact of ES and the issues arising from the use of ES at the organization level. The conclusion of Section 4 outlines the future of ES technology. Finally, the last section summarizes the findings of this project and draws conclusions of our investigation into the state of Expert Systems.

CHAPTER 2 BUILDING EXPERT SYSTEMS

The purpose of this section is to discuss what goes into building a large expert system. By reviewing this process I hope to show some of the necessary issues an Engineering Manager should consider before plunging into the development of a large expert system.

2.0 Expert system development process

Dennis O'Connor in presenting Digital equipment's experience in developing expert systems presents the following 4 stage model in the development of an expert system [36].

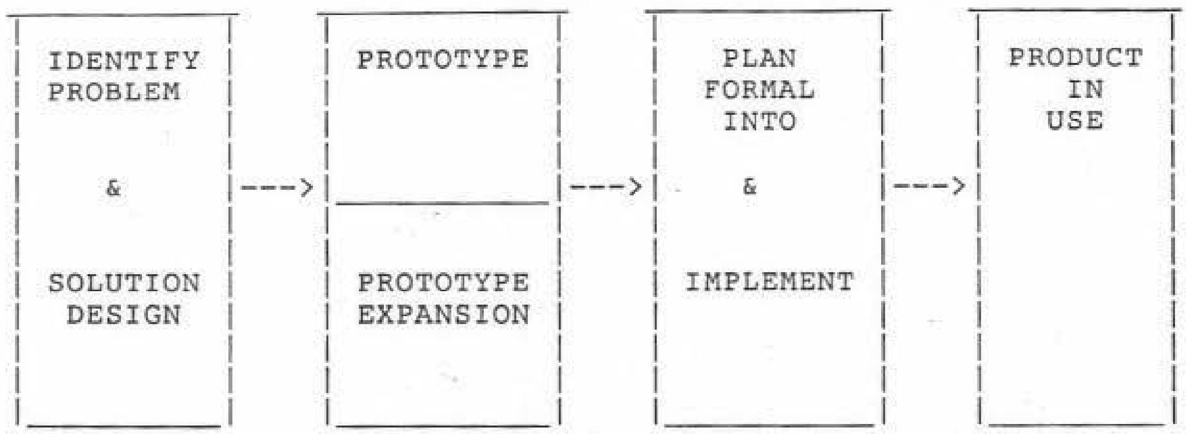


FIGURE 2-1 EXPERT SYSTEMS DEVELOPMENT PROCESS

The first phase includes the identification of the problem, determining the AI methodology and tools, the design document including testing and validation acceptance criteria.
The second phase includes the building of an initial shell prototype and demonstrating it to the hosting organization. It is at this point that the permission is given to go ahead or stop the project. If the go ahead has been given then the prototype is expanded adding more expertise to the system.
After the expert system has achieved a reasonable level of expertise it is inserted in parallel with the existing process and prepares the way for the initial technology transfer.
The third phase develops a formal introduction plan. The plan outlines all resources, training and documentation required for a successful transfer. A phase review process is implemented to measure progress and ensure the system meets its schedules milestones.
The fourth phase deals with the production environment and addresses maintenance and changes to the system. Large expert systems are never entirely complete because knowledge must constantly be added to the systems.

This paper covers the first two phases of problem identification, solution design, prototyping and prototype expansion.

2.1 Identification of the problem.

The first step in developing a large expert system is the identification of a problem. The previous section covers the types of problems which are appropriate for applying an expert system solution. Although the problem may be a suitable type of problem for applying an expert system there are many factors which make up the justification for building the system. To determine whether or not it's possible to develop the system it is often necessary to conduct a feasibility study.

The following picture indicates the elements that go into determining the feasibility of an Expert system.

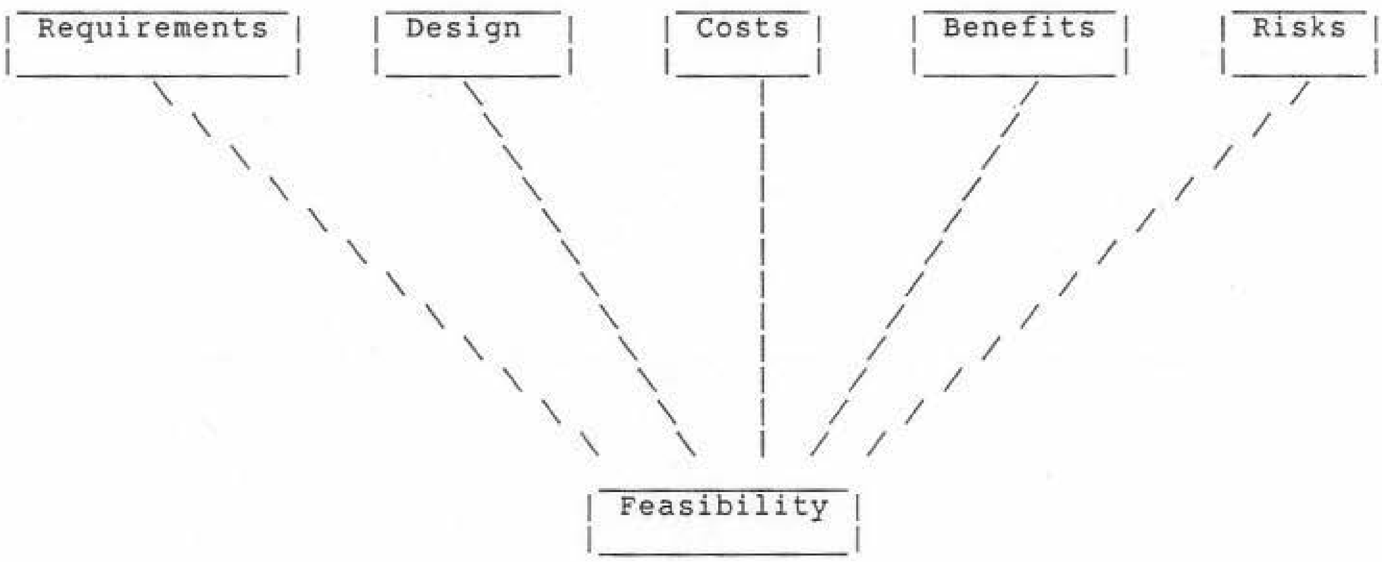


FIGURE 2-2 Deriving the Feasibility [50]

The following sections elaborate on the various aspects of determining the feasibility of an expert system. The feasibility study can be thought of as expanding the problem that's been identified and as well as proceeding on the way to a full solution design.

2.2 Developing Requirements

The development team.

The development team consists of Management, The Project Champion, Project Leader, and Knowledge engineers. This team will grow as the project passes the feasibility stage and proceeds into the development stage.

Management's role is to represent the organizational needs and requirements for the expert system. Management is also involved in providing the other resources, such as the expert's time, required for the development of the system. Management's requirements are usually focused on issues of how the application will effect the organization, and the cost of deploying the system verses the expected benefit of the system.

The project champion is a person with some stature within the organization and a belief in the purpose of the system. This person is a facilitator for the development effort and helps resolve the problems which arise during the development.

The project leader carries much of the responsibility of developing the technical portion of the expert system. This person needs to be familiar with knowledge based systems and be capable of dealing both with the developers and users of the expert system.

The knowledge engineers are the gatherers of information and the builders of the system. The design process requires them to be familiar with both the terminology of the experts they interview and the expert system methodology which captures that expertise.

Gathering information.

The knowledge used in the expert system can come from a variety of sources such as various publications, Textbooks, reports, video sources etc. These sources usually only contain part of the expertise necessary to build the system. The remaining knowledge must be extracted from experts within the organization. This knowledge is usually much more complex than that found in the documented sources. Since the expert knowledge is the heart of the system the process of gathering this information is critical to the success of the project. Factors to be looked at in selecting experts are availability, interest in the project, motivation and ability to express their expertise.

Interviewing experts.

The expert information is usually extracted from the experts by an interviewing process. The more an expert system relies upon first hand expert information the longer and more costly the development of the system is likely to be. [48]

One method for ensuring accurate information is to conduct interviews on a two knowledge engineers to one expert basis. One of the engineers serves as the interviewer and the other as a scribe. The engineers can then compare notes as to what was really said during the interview. [50]

If more than one expert is used to determine the rules to solve a particular problem a conflict of information may arise. Both experts may use perfectly valid but different methods to solve the same problem. The resolving of these differences adds time to the building of the system. An advantage of using more than one expert is to validate the information received.

The information gathered from these interviews will be used to build prototypes. At this time the experts can be re-involved in determining the validity of reasoning process they gave in solving the problem. This is discussed further in the section on prototype building.

Group Requirements

Each group involved with the building of the expert system may have their own requirements for the system. The perspectives of these different groups should be taken into account when developing the requirements document.

The expert's requirements usually pertain to the type of knowledge the system contains as well as the stability of the system. There is a tendency for experts to demand too broad of boundaries for the application requiring the system to solve 100% of all the cases rather than a more reasonable limit. [50]

The requirements of the user can vary depending on the computer literacy of the user, the amount of education they have had or the amount of training involved in the system. Careful attention needs to be paid to fitting the system to the users environment. This can be done by having the user participate in the design process of the expert system prototype.

The knowledge engineer's requirements can range from writing the application in machine language in order to make the application fast to using the latest state of the arts tools. The requirements must separate what is needed from what is wanted.

2.3 Evaluating the requirements

After the requirements have been gathered they need to be validated and prioritized. Some requirements put forth by one group may be overridden by another. For example a user may have issued a request that the system produce a graphics display of a result which would increase the cost and complexity of the project. Management may assign that a low priority while the user assigns it a high priority. After the requirements have been determined and the information has been gathered the technical, economic and cultural issues need to be evaluated. The issues of changing requirements or requirements which cannot be finalized needs to be evaluated also.

Several factors go into determining the suitability of the problem to an expert system solution. Some of these factors are:
Can the problem be solved by a traditional programming solution?
If this is the case its better to go with a traditional solution because its usually cheaper than capturing and testing the appropriate expertise.
Is the problem clearly defined and have limits been established?
Poorly defined problems can still be solved but there is a cost to generating a better definition. A better definition of the problem may still prove the problem to be unfeasible.
Does Expertise make a difference in solving the problem?
If current experts aren't solving the problem then the system can't be expected to make a difference.
Can the required expertise be acquired by a computer system?
If the expertise requires a specialized development of one of the senses such as touch, smell, sight then the system can't be developed using current technology. If the expertise is very specialized and acquired by years of training it may be very difficult to transfer that knowledge to an expert system.
What type of reasoning is involved in the expertise?
If the reasoning involved relies upon intuition it may be difficult to capture. The body of "common sense" reasoning that an individual develops over their lives is very large and has yet to be put into a computer.

The characteristics of the knowledge needs to be determined in designing the system. In evaluating the nature of the knowledge, it is important to determine what portion of the knowledge; is a permanent portion of the system, is stable but changeable by the system and, may change every time the system is used. Some data required to determine an expert solution may be too large to handle with todays technology. For example the case database a lawyer may search to defend his case may cover many thousands of books. This much data may not be feasible to store or retrieve efficiently with todays technology. The rules which govern an experts decision may be so complex that it may not be possible to capture the expertise in a reasonable amount of time because either the computer processing time may take days to evaluate the solution, or the knowledge expert cannot generate a set of rules in a reasonable time frame.

2.4 Evaluating the feasibility issues.

Determining the Benefits of an expert system starts with the assumption that the system has been built and then assesses what benefits will the system provide. A list of the some the advantages may include; improved reliability, reduced response time, increased availability, flexibility and improved product quality. These benefits need reviewed assigning a priority and a value to each one.

There are many elements to determining the costs of an expert system. Some of these costs are easy to assess like the cost of the hardware or software tools. Other costs like the expert's time and operational time are more difficult to assess. A list of costs associated with an expert system may include; hardware and software tools, expert's time for obtaining and validating information, operational time to test and validate system, training time, support time and ongoing maintenance of the knowledge database.

Many of the risks associated with building an expert system are technical in nature while others involve the environment where the expert system is being used. Some risks that need to be assessed when building an expert system are; The use of new technology, If the system is the first one being developed (Pioneering), Availability and cooperation of the experts, Technological level of the users, receptivity of the corporate culture and, management style of the organization.

Constraints for building the system need to be evaluated also. Some examples of constraints which could effect the feasibility of the project are; every application must be PC based using limited memory, the users must not know about the project, the application must be completed on two months or , the budget is only ten thousand dollars.

The feasibility study may uncover problems that indicate solution may not be feasible to develop. At this point it may be possible to find a way to work around to problems or it may be necessary to abandon the project.

2.5 Solution Design

After all the requirements have been determined and the application has been determined to be feasible, the appropriate representation for the problem and tools must be chosen. It is beyond the scope of this paper to discuss every possible knowledge representation and knowledge engine. What follows is a brief summary of a few techniques commonly in use.

Knowledge Representation

Several basic methods have arisen to describe the storage and retrieval of knowledge in an expert system. These are frames, rules and logic. Although each method has its advantage for a certain class of expert related problems, they are not mutually exclusive of one another. A frame based approach can be used by a logic or rule based system. Likewise a frame based system can allow both rules and logic statements to be used within frames.

Rule Based

A rule based system is best used where most of the knowledge can be expressed a a series of rules related by experts. Experts tend to relate their methods of problem solving in terms of situations and actions, such as IF this happens THEN I do that. The advantages of a rule based system are;

"1. They include practical knowledge in conditional if-then rules,
2. their skill increases at a rate proportional to the enlargements of their knowledge base,
3. they can solve a wide range of possibly complex problems by selecting relevant rules and the combining the results in appropriate ways,
4. they adaptively determine the best sequence of rules to execute, and
5. they explain their conclusions by retracing their actual lines of reasoning and translating the logic of each rule employed into natural language" [17]

```
                                               Knowledge base
                ___________                 ____________________
               | WORKING   |  Triggering    | RULE  |  FACT     |
     Inputs -->| MEMORY    |--------| data  | MEMORY|  MEMORY   |
               ---|----A---         |       --------|-----------
                  |    |            |               |
                  |    |            |               |  Rules
                  |    |            ------------ |  |   and
         Data     |    | Update                  |  |  Facts
                  |    |                         |  |
                  |    |                         |  |
               ---V----|-----                   -V------V-----------
               |   Rule     |  Selected rule   |  Rule and data    |
     Outputs <-|interpreter |<-----------------| element selection |
               -------------   Selected data   --------------------
```

FIGURE 2-3 Model of a simple rule bases system [17]

Rules - Rules are expressed symbolically in the computer as conditions and actions.
Rule Interpreter - Matches a rule to working memory data. Generally pattern matching is used to match the constants in working memory to the constants in the rule patterns.

There are several methods of organizing rules and interpreting sets of rules in a rule based system.

Forward Chaining

In a forward chaining system a rule is triggered when data changes which matches the rules conditions. If a system is accepting new knowledge and determining the implications of the new knowledge then it is a suitable application for forward chaining.

E.G.

If its Monday through Friday I'm at work
  If I'm at work then I go out for lunch.
    If I go out for lunch I don't have a sandwich to eat.

If its Saturday through Sunday I'm not at work
  If I'm not at work then I eat lunch at home.
    If I eat lunch at home I have a sandwich to eat.

If I tell the system that its Monday than it can infer that I a didn't have sandwich to eat for lunch.

Backward Chaining

If a system is being designed to analyze a problem and generally starts with data containing all the available facts then it's a candidate for backward chaining. In the preceding example if I tell the system I had a sandwich to eat for lunch it could infer that I had lunch at home or that its Saturday or Sunday.

Mixture

If a system is begin designed to analyze a situation that has many potential states then a combination of forward and backward chaining can be used. Forward chaining can be used to determine the subset of problem states that might exist. Backward chaining can be used to determine the sequence of rules which implies the consequence.

Rule based systems tend to get bogged down if there are a large number of rules. If the system can be decomposed into smaller independent rule set of about 50 rules then another method to structure the knowledge or rule base should be considered.[50]

Frame Based representation

Frame based representation is a method for structuring knowledge in a knowledge database. Frame based representation have been developed to meet the criteria of Expressibility, Understandability and Accessibility in representing the knowledge stored in knowledge based system. Frame based system can store not only the data but the taxonomic structure of the data. By structuring the data using frames the essential character of the object is defined once eliminating the need to derive the same properties for the same types of objects. Another advantage of frame based representations is the ability to allow to rules to be more generic and less dependent on the data. This reduces the overall complexity of the application.

```
                     /- buildings
physical.objects --< - computers
                    |\- things.owned.by.paul - - - - - - - - - - - - - |
                    |                                                   |
                    |                     /-  sedans - - - car2        |
                    |      /- automobiles -< -  coupes                  |
                    |     /                \-  stations.wagons          |
                    |    /          /- huge.grey.trucks - - - - - - truck1
products ----   vehicles <- trucks < - big.non.red.trucks
                          \- boats
```

FIGURE 2-4 A frame taxonomy

Frames provide a structure to represent objects or classes of objects. The automobiles frame above represents a class of all automobiles and the car2 frame is a member of the automobile class. Frames allow classes of objects to be specified as members of larger classes and specifications to be grouped into taxonomies. Taxonomies also allows the specification of subclass and member relationships. Vehicles is a subclass of both physical.objects and truck1 is a member of both huge.grey.trucks and things owned by Paul. [13]

Although Frame systems provide a powerful means for inferring knowledge about an object they provide no direct facilities facilities for describing how knowledge in the frames is to be used. Rule based reasoning can operate effectively on frame based systems. The term "object-oriented programming" has been used to describe style of programming used on a frame based system.

In the course of feasibility study if the characteristics of the application indicate that;
The objects and concepts of the domain bear a relationship to each other, there are a large number of related frames of knowledge and, the reasoning process involves multiple levels of detail about certain object then this is and appropriate situation for using a frame based solution.[50]

2.6 Prototyping the Application

The purpose of a prototype is to provide a model of the application which is as complete as possible. Prototyping is essential in developing an expert application because it is the only way for the developers to determine the structure of the database and the reasoning associated with it. Not only will prototyping show the user of the system what the system will look like, it also provides the more important function of validating that sufficient and appropriate expert knowledge has been gained to solve the problem.

The prototyping phase begins with an initial prototype created from the design information. This prototype is reviewed with the experts for completeness and validity. The prototype is then refined with information from the review and the processes repeats itself. The key to this method of refinement of rapid prototyping is advent of new tools coupling highly interactive workstation with a means of representing and reasoning with the knowledge. The number of times prototypes may have to be refined varies depending on the complexity of the application but experience has shown that anywhere from four to ten iteration of a prototype may be necessary. [50] When all the inputs from the review sessions have been included in the prototype the prototype is ready for evaluation.

One the most important parts of evaluating the prototype is providing the right case data. Case data needs to be selected to handle common situations as well special cases developed to test certain complex areas of the applications reasoning. Test cases can come from historical data, current data and generated data. The expert needs to be involved in analyzing the test situations and coming up with solutions to be compared with the solutions provided by the expert system.

In evaluating the prototype tests need to be developed to check; completeness, consistency and robustness. One method of testing for completeness is for the developer and the expert to review the system together with regard to the assumptions that have been made and determine whether other situations have arisen where there needs to be additional rules defines for the assumption to be valid. Consistency checking tests whether each component provides the data each other component needs in the expected fashion. Some systems provide mechanisms for checking the consistency of data. The consistency checking of rule sets can be done against the characteristics they have in common. In a frame based system such test can be included in the framed based taxonomy. Testing for Robustness checks how well the system can perform if its receiving poor inputs.

2.7 Conclusions on developing an expert system

The development of a large expert system is similar to the the development of any large engineering system. The successful implementation of an expert system requires project methodologies and management.

As with implementing any significant sized project an expert system needs;

Requirements document or Functional specification.
A feasibility study.
A design document.
Prototypes.
Evaluation and rework of prototypes.
Pilot Application.
Operational Period.

The development process should be managed like a project.

The more "state of the art" the expert system the more important each one of the preceding steps becomes. For example if expert system technology is being applied to a new problem prototyping becomes a very important method of refining and validating the design. However if an existing expert technology has been proven before prototyping may only serve a role in tuning the technology to the current environment. Rapid prototyping is an effective way to refine a design. Prototyping is essential to evaluate, test and validate the design. Prototyping is not a replacement for a design of a large knowledge based application. It is the design which provides the for direction the prototyping. Initially several small prototypes may be developed each representing only a portion of the knowledge of a large system. The design determines the means of integrating the prototypes into a larger system in a consistent manner.

3. CASES STUDIES

3.1 ACE: AUTOMATED CABLE ANALYZER

"ACE(automated cable expertise) is a system within the problem domain of telephone transmission equipment(55)." ACE reviews and analyses large amounts of CRAS* data and creates diagnostic messages for trouble found. ACE is used to assist the cable analyzers who are the human experts on outside telephone plants and are in charge of the maintenance. These persons use ACE to make their decisions about where and how to guide the maintenance efforts. It was decided to create such a system to help the cable analyzer because the analysis of the CRAS data consumes so much time and become "tedious." In addition the cable analyzers have to develop other tasks(55). Besides, this diagnostic requires so much knowledge and the amount of experts in this field is limited(35, 36).

This expert system(The ACE product) makes four analyses:

-Found trouble analysis(FTA)

-Trouble Report Improvement methodology(TRIM)

-Plus three analysis(P3)

-Pair transportation analysis(PTA)(55).

The analysis commences with the identification of the equipment that exceeds "a threshold number of the CRAS reports." This is done in order to identify the trouble and begin the analysis of the CRAS data. P3 guides the cable analyzers to the cables that are not giving good service to the clients. FTA and TRIM determine what the real problem is, the probable cause and what equipment is entailed. PTA analyzes what section of the cable is defective and locates the geographic position of the defective section. In addition PTA gives the distance a cable might run. ACE is running in an "UNIX" environment on an AT&T 3B2 microcomputer. At night, ACE uses a phone line to access the remote data base and return to the 3B2(without losing contact with the data base in order to extract more data when needed). After ACE gets the data, it begins its analysis which is done in one to four hours. At this time it has the results ready for display to the users the next day(55).

ACE's evolution:

-Definition of the problem domain

-Definition of a prototype system and knowledge base

  -Choosing a Franz LISP and OPS4 product system language for implementing the knowledge base.

-Work between the knowledge engineers and cable analyzer(expert) to define rules of the cable analyzer's outside plan analysis strategies for translation into OPS4.

-Evaluation of the prototype(55).

-Once the prototype is proved, it is decided to transform ACE from prototype to product. At this time the main task is the selection of the technology most appropriate for implementing each system function. The expert system technology is chosen for the ACE knowledge base and conventional technologies for the modules needed to support the ACE system operation(55).

The ACE developers found that a prototype expert system "may need extensive expansion" to give the expected results that a commercial product needs, that the use of an expert system alone would not give the results expected from ACE "ease of use and reliability" and that it was needed to work interactively with the experts to develop the knowledge base to make sure that the results provided by ACE would be of excellent quality(55).

* ACE is a data base and report system which contains records of past repair data for reviewing by the cable analyzer in the effort to identify and diagnose trouble in the outside plant(55).

3.2 XCON

XCON is an Expert System used in Digital Manufacturing operations to configure all of the VAX 11 system orders that the buyer has sent to the company. When a buyer sends his order it is entered into the XCON system which determines if it needs any correction in order to be a "consistent and complete system"(35, 36). XCON has 4000 rules and is implemented in OPS5 language and runs on VAX 11/780 and 11/750 machines running the VMS operating system. The use of XCON allows the company to get most of the things that were not obtained with a manual system used before, such as production of VAX orders with accurate configurations, better utilization of the human resource and cost saving(35, 36). The development process used by the XCON developers group was as follows(35, 36):

Stage I: Problem Identification and Solution Design Document.

1. Identification of the problem
2. Knowledge Engineer writes what they understand about the problem and its "interdependence"
3. Decomposition and Segmentation of the problem
4. Selection of the methodologies to be followed and the tools to be applied
5. "Outlines the problem space and solution proposed," define the job to be done, the job scheduling and the progress to be achieved, definition of how the project would be tested and validated and what the criteria were that the buyer would use to accept the product.
6. Definition of the human and non-human resources needed to meet the goal.
7. Definition of the support given by the organization(35, 36).

Stage II: Prototype and Prototype Extension

1. Presentation of the prototype to the organization to decide whether or not to continue with the project.
2. After the acceptance and expansion of the prototype insertion of the system in parallel with the existing process is needed.
3. Planning the use of XCON in the organization.
4. Preparing the user for implementing the system.(35, 36)

Stage III: Formal Introduction Plan and Implementation

1. Making an introduction plan which includes outlines of the resources, training and documentation needed to transfer the technology.
2. Measurement of progress(35, 36).

Stage IV: Production Environment

1. Maintenance
2. Addition of changes to the system(35, 36).

The developers' group found that to succeed in the implementation of the project it needed to have an adequate management program to support the process and to integrate these four stages(35, 36).

3.3 EMBEDDED AI. EXPERT SYSTEM TROUBLESHOOT AUTOMATED ASSEMBLY

The basic concept employed in solving the problem was to build an expert troubleshooting system into microcomputer software. The software, employing valid troubleshooting algorithms and working with inputs from the maintenance technician, would specify what actions the technicians should take and what information he should gather and input to the computer. The software would then deduce the source of the malfunction and give appropriate direction to the technician for repairing it(47).
This expert system was applied to use in the clutch assembly machine. In this specific application, the system provides assistance to the technician beyond doing his troubleshooting "thinking" for him. It also provides, from a computer-controlled videodisc, highlighted visuals that show what he should be doing, or what he should be looking at. This further reduces the training requirements and the incidence of errors in carrying out the instructions given by the computer to the technician(47).

The concept of this application:
1. The technician identifies improper operation of the assembly line(eg. the line stops) and identifies the malfunctioning station(s) through the use of fault lamps located above each station.
2. The technician wheels a troubleshooting cart, equipped with a monitor and push-button control, to the station, inserts a plug from the cart into a specially installed junction box at the station, and turns on the monitor. The computer and videodisc player, cabled to the junction box, are at the ready in a central protective enclosure.
3. A menu appears on the monitor from which the technician selects the assembly line and station. Using the push buttons described earlier, he can also select special procedures for the station such as calibration, mastering or semi-automatic operations.
4. The computer then specifies the test or actions the technicians should perform first. The computer also presents pictures from the videodisc, illustrating for the technician where he should be looking or what he should see.
5. The technician tells the computer the results of the first check using the two special push buttons.
6. Based upon these results, the troubleshooting rules and the characterization of the system, the computer specifies the next section for the technician. This interaction continues until the source of the trouble is narrowed down to a single floor-replaceable component that is being manufactured(47).

The benefits for using the expert system:

- Almost any member of the available work force can do this troubleshooting without extensive training. Practically no training is required to use the two button system and to follow the computer's directions(47).
- The technician doesn't even have to know the correct or nominal reading at the various test points; they can be included in the computer knowledge base(47).
- Within the constraints of the specific system, the most efficient course of troubleshooting is followed; errors in logic are greatly reduced or eliminated(47).
- The computer's logic and ability to "think" clearly are not affected by the stress caused by a key production line being down, as would be a factor with a human troubleshooter(47).
- If use of test equipment is required, the videodisc can provide specific, detailed and illustrated instructions in still or in motion pictures to show the technician exactly how to look it up, adjust it and read it(47).

3.4 WELD-ASSIST: AN EXPERT SYSTEM FOR ROBOTIC ARC WELDING

In the field of welding, this system may close the gap between a qualified robot operator who is inexperienced in welding and the skilled welder. With such a program, a robot operator is not only able to fulfill most of his welding task, he will also be able to react when welding problems like defects occur. He can use the expert system as if he were consulting with an expert(28).

Since many problems do not originate from the robot, but are caused by the welding process, an expert system was built that provides the user with a welding schedule for his current welding task, and in case of discontinuities, the system gives helpful hints to prevent and correct defects. The system also helps the robot operator to improve an unstable welding process or an imperfect shape of the weld(28).

The program is designed for using with the gas metal and welding process(GMAW) and with mild and low-carbon steel. During the consultation, the user is guided through the program by questions. He can choose from three program sections(domains):welding schedule, discontinuities or improve process (28).

The program (Weld-Assist) is divided into three sections:

1. Welding schedule- provides the basic welding data for a given welding task.
2. Discontinuities- make recommendations for joint preparation, welding data, and electrode, to prevent and correct discontinuities or defects.
3. Improve process- suggests how to improve the process characteristics and weld appearance(28).

Weld-Assist can be used as a preprocessor for a robot controller. Recommended welding data can be fed automatically to the power source, wire feeder, and robot controller. Also, the welding cycle can be initiated by the expert system. The other feature is a user-edited database. In addition to the existing data base for welding data, a second data base can be generated(28). Thus user can store his personal welding data after performing an optimized weld(28).

4.1 SOCIAL IMPACT OF EXPERT SYSTEMS:

In the trend of human civilization, knowledge is taking a much more important and indisputable place in today's world. If we take a look at the history of early civilizations, they controlled other nations by using mere power, but later, mere power left its place to religion. After religion technological exploitation became dominant and very effective weapon on controlling other nations for developed countries until the end of WW.II.

After that, the most incredible era of human civilization started by producing huge amount of information. Naturally this era came along with lots of problems. Actually in the process of Knowledge development, people had leadership, but this era brought the idea of knowledge system that produced information; therefore, some part of the idea contains social impact of using this system.

In the well-known problem of this improvement is the possibility of using those Artificial Intelligence and its mature form Expert Systems instead of human expert.

4.1.1 Can Expert Systems Replace People:

It has been suggested in some articles and books by several author and scientists that expert system has a capability and potential to replace people. If we think radically this is not new, computer systems have been causing job description to change for thirty years or more. Probably we may think what the difference is between expert systems and traditional computer systems which we are familiar. The new prospect coming along with expert systems is replacement of people especially professionals.

Some professional level tasks requiring specialized intelligence can actually be done as well or better by an expert system than by a human expert. In some cases this is a desirable feature, while in others it is a subject for real concern. An insurance underwriting expert system could replace a large percentage of a company's underwriters, and this may be desirable both for the underwriters and the company, since those people could frequently be doing even more useful tasks elsewhere in the company.

For the time being, however, it is better to think of an expert system more as an apprentice, an intelligent assistant, than as a replacement for human experts. This approach is more realistic in the near term. Certainly no expert system is going to replace anybody until its knowledge base matures, which can be a matter of several years. And in many cases, we would like to have human present for their humanness as well as their expertise: physicians or receptionists are examples of this.

On the other hand application of expert systems may be considerable prospects for some experts only one in the company who can do a particular task or in knowing that their expertise will not be lost, if they retire, the advent of an expert system is a freeing experience for this kind of person.

Diametrically opposed to this attitude are those experts who feel that not only their job security but also their very worth as a human being, is based on their unique ability to do a particular task better than anyone else. To take away that task, or even suggest a computer might be able to do it, becomes a personal insult and is often met with overt hostility. If such an attitude can not be changed unequivocally, such a person should not be engaged to develop the knowledge base.

4.2 PROBLEMS IN THE PRODUCTION OF INFORMATION:

During the integration process, designer usually meets with different number of problems depending on the application. Some of these problems may be stated as follows:

. Defining the States:

A good identification of states is really important since these states really play a borderline effect for the domain of the problem, ex.for technical system, it is easy to distinguish between the states such as active or inactive. But think of crafting an ES for Human Diagnosis, what will be the states and the number of the states, perhaps infinite. And usually these are less well defined than technical systems. Therefore defining states is really important during the crafting and integration process of ES into organizations.

. Verification:

For some systems a diagnosis can usually be verified as an output of ES. Where on the other hand some systems may not allow this, ex. in technical systems, a diagnosis can be verified by exchanging a part or applying a certain test. But in medical situation this is very risky, that verification may not be applied to all running systems. Therefore the implementation and efficient use of produced (may either be extraction or acquisition which are discussed in the following chapters) information may be as important as how to achieve that information.

4.2.1 KNOWLEDGE EXTRACTION VERSUS KNOWLEDGE ACQUISITION FOR ES

Although the borderline between both processes is difficult to draw,their domains are very importantly different in principle. Acquisition system in AI is the one which is capable of automatically producing new concept structures or rules from a set of data the system receives as an input. The borderline, between Knowledge Extraction and Knowledge Acquisition is that acquisition system asks for specific inputs to build or to guide its construction task by itself.

Knowledge Extraction can be mentioned when a Knowledge engineer tries to discover the knowledge the experts are using for their decisions or this process may be putting together the structures which may explain or predict their decisions, thru interaction with experts. ie. Knowledge taken from experts must be manipulated to result in knowledge structures which can be used for implementing knowledge representation in actual ES.

There is great danger in mixing up the two. If it is mixed, the Knowledge Engineer can have the risk of neglecting the proper merit and heuristical value of the natural expert. The Knowledge Engineer may think that if he finds himself a structure for the task performance, his job is finished. By doing so, he neglects the mastery which is collective experience and intelligence of natural experts. On the other hand, also the intelligence of the natural experts is not in itself relevant but only a heuristic basis for the development of the Knowledge Base of ES.

Of course both knowledge extraction and knowledge acquisition are very important. Knowledge acquisition is important for the generation of new concepts and structures, and rules for the new task performances or for improving task performance. On the other hand knowledge extraction is important for knowledge acquisition to get structures to implement in the ES.

Extraction Techniques for ES Feeding:

> 'If you are building an expert system and are a domain expert, have a knowledge engineer [or epistemologist] help you understand and formalize your problem-solving methods. If you are a knowledge engineer who has studied the domain extensively (and think you are an expert), work with a real expert anyway. If you are, in fact, a bona fide domain expert and an experienced knowledge engineer (a rare combination), play the role of knowledge engineer and find someone else to act as the domain expert.'

D.A. Waterman, 1985
A Guide to Expert Systems, Addison-Wesley
(pp:154)

One possibility for the above reasoning may be that, one is tempted to give too much weight and importance to ones's own introspections, while at the same time it is practically impossible to see whether the introspection gives real information about the processes that goes on.

Today, methods used for the extraction of knowledge are the methods used by a system analyst. Phases of complete extraction process are listed below:

First Phase:

Before starting with interaction with the domain experts, the knowledge engineer needs to become acquainted with the public knowledge on the domain via existing documents, manuals, literature, publicity, etc.

Second Phase:

Preparatory contacts: One must make himself acceptable to the domain experts or expert teams. For achieving this goal:

- make informal contacts
- become familiar with the work place, environment and with involved personnel
- give the experts time to become accustomed to the presence of the observer and his tools of observation
- gather written texts, flowcharts, notes and diagrams used by the experts

Third Phase - The Non-intervening observation:

The knowledge engineer observes the focused domain experts in action during their natural expertise actions and communications. The observer follows a strategy which is mainly data driven.

Fourth Phase - The Participating Observation:

The degree of integration and cooperation is far greater with the method of participant observation. During this stage the knowledge engineer must prevent himself of thinking he's an expert. This participative observation is justified by the knowledge engineer presenting himself as a temporary member or as tool builder.

Fifth Phase - Intervening Observation:

The knowledge engineer intervenes actively in the observational situation, either by means of questioning the experts or expert teams or by open or directed interviews, asking the domain expert's vision about task performances. In the directed interview 'What if..?' type questions may be very relevant and helpful.

Sixth Phase - The Experimental Approach:

With the models the knowledge engineer has, the domain experts are confronted with a certain task to perform. Small changes to the task under control can be made. Eventually one can ask the domain expert to comment about his own task performance.

Seventh Phase:

Ask the experts to comment on the constructed or cognitive models the knowledge engineer has constructed.[49]

4.2.2 THE PLACE OF KNOWLEDGE EXTRACTION IN THE PROCESS OF ES. DEVELOPMENT:

(See figure 4.1) In the first run, target is usually called a Demonstration system. In the second run it is a Prototype, in the third an Operational System. And in the fourth run it can be called a Delivery System. However this process is somewhat idealized. Cycles per state may take more than one run.

4.2.3 LEARNING FLOWCHART:

(See figure 4.2) Any system designed to modify and improve its own performance must include the following major components:

- A set information structures which encode the system's present level of expertise (the rules)
- A task algorithm (the performer) which uses the rules to guide its activity
- A feedback module (the critic) which compares actual results with those desired
- A learning mechanism (the learner) which uses feedback from the critic to amend the rules

Most expert systems have a read only Knowledge Base. What we are talking about here is an erasable-programmable Knowledge Base.[6]

4.2.4 AUTOMATING KNOWLEDGE ACQUISITION:

When learning is viewed in its broad context as a part of intelligent behavior that characterizes many living systems, the role of influence and continuity in learning become apparent. Whether the mechanism of learning is as simple and crude as memorization or as broad and complex as induction, the role of selective growth in spreading influence can be seen. Whatever the learning method depends on, some parts of the learning structure must be augmented and some parts removed as learning proceeds. Thus in order that this augmenting and removing process proceeds successfully, the system should have the property of continuity.

A retrospective view suggests that the function of living things is to have influence. An organism can extend its influence by:

1. Staying alive
2. Begetting a family
3. Sharing or communicating with others
4. Creating a system or machine that has influence.

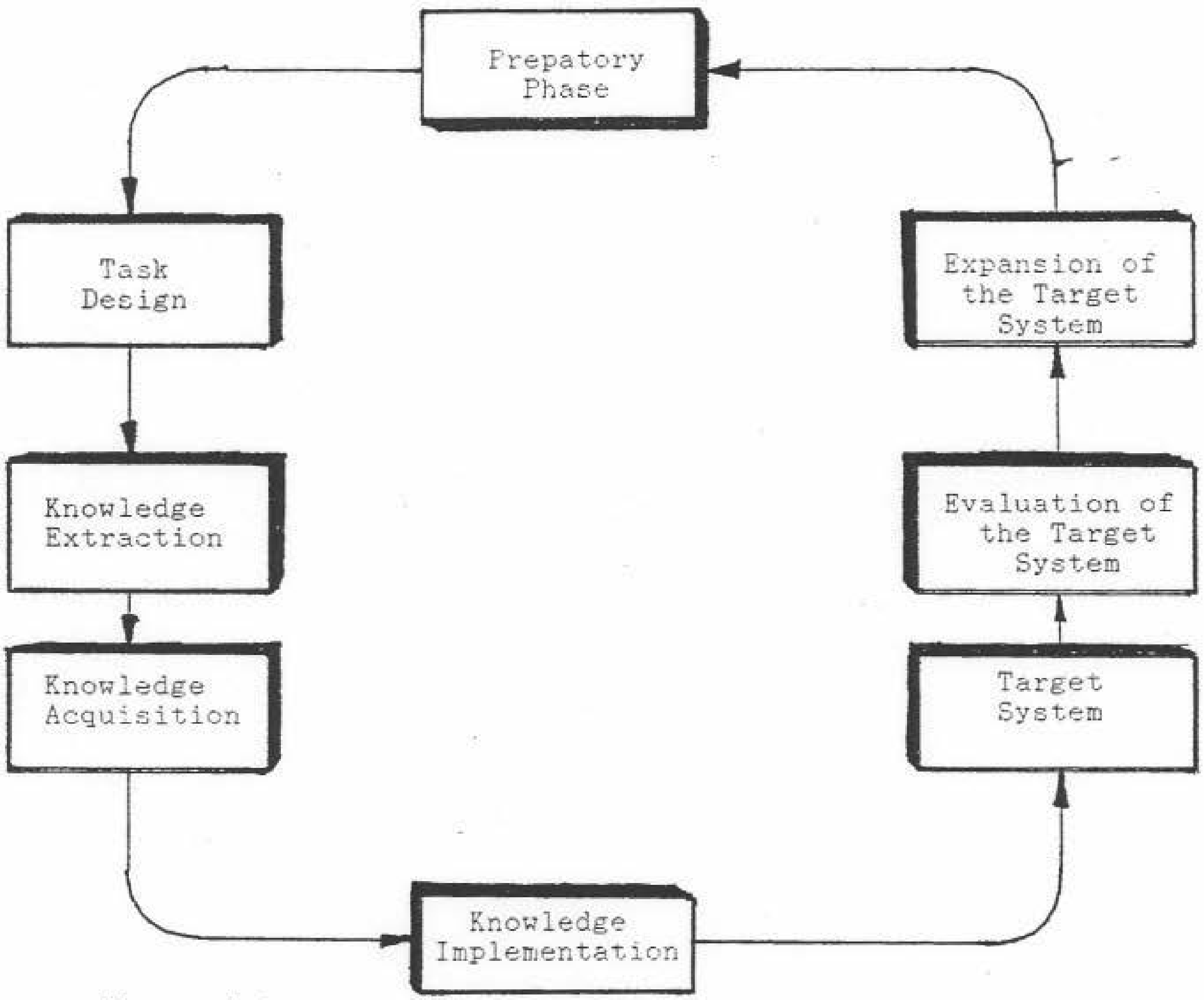

Prepatory Phase
Task Design
Knowledge Extraction
Knowledge Acquisition
Knowledge Implementation
Target System
Evaluation of the Target System
Expansion of the Target System

Figure 4.1

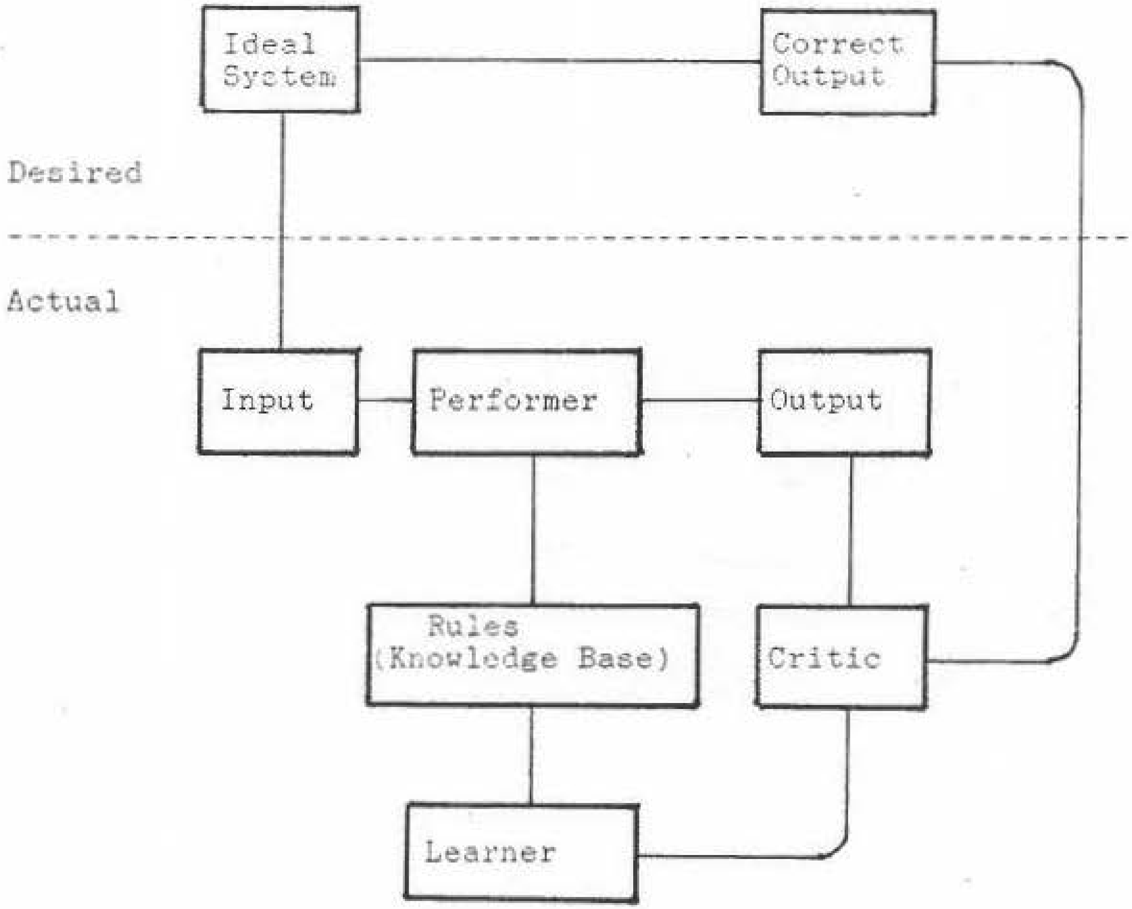

Ideal
System
Correct
Output
Desired
Actual
Input
Performer
Output
Rules
(Knowledge Base)
Critic
Learner

Figure 4.2

The first 2 are objectives that people have in common with all living systems. Scientist regularly engage in the third form of influence when they publish articles, lecture and confer with colleagues. The fourth is the challenging and least attainable.

For a system to spread its influence, it must generate new twists on the information that it transmits. Simply transmitting someone else's information doesn't constitute perpetuation. But also, exactly what twists will be successful in the environment can't be known in advance. Some amount of trial and error is needed. Intelligent organisms try to minimize the error by making trials whose probability of success is high and the best known criterion of success in a highly interactive world is that what is good now is probably likely to be similar to what'll be good in the near future.

In most learning experiments the environment is isolated. It gives (rules, examples) or inputs to the learning system, but the system doesn't compete with other systems in the environment. Any influence that occurs is at the micro level of the learning system's internal. At the macro level of the learning system's interaction with the environment, matters are so constrained that influence can only be indirectly seen to be occurring.

Best strategy of learning is decided to have the form of an evolutionary process where a population of trials is made, feedback is obtained, the trials which did well are repeated and the the trials which did poorly are unlikely to be reported.

Memorization may be viewed as a kind of learning but it alone seems to be very inadequate. To make it more powerful, it can be modified with a device called "selective forgetting". In this method, items of memory, whose value to the system is considered to below some threshold value are erased. New items that are given a chance to prove themselves to be useful are added to Knowledge Base (KB). New items come from the outside. Thus this method allows the environment to influence the learning system but doesn't allow the learning system to monitor influence within itself.

As a second method, Knowledge Debugging may be considered. Here the learning system is told what it should know but must convert this knowledge from its input form to a form which permits the system to perform the knowledge. In other words, input must be made operational.

Another method-Weight Adjustment-tries to set some systematic methods for the correction of thresholds and attenuations. One obvious strategy includes raising the attenuations (the number by which the certainty of the conclusion is multiplied) when a proposition is concluded without enough certainty and lowering the attenuations when a proposition is concluded with too much certainty. The amount by which a rule's weight is to be changed will depend on the past performance of that rule.

Above mentioned methods are related to influence where influence is a kind of selective growth. A necessary condition for influence is continuity. If the definition of continuity is really precisely circumscribed, quantifiable results that facilitate comparison across the experiments can be achieved. [29]

4.2.5 ADAPTIVE LEARNING:

Application of ES technology is still restricted to fairly narrow,self contained domains and performance decreases rather sharply as the system approaches the boundaries of its knowledge. There is little ability to adapt or reorganize knowledge as performance requirements change over time. Although preceding learning systems are good enough, most potential for providing long-term solutions to these problems has in the area of adaptive learning which offer several opportunities for acquiring, refining and reorganizing system knowledge in response to observed performance. Although the learning process can be formulated as a search, the manner in which this search is conducted will ultimately determine the general effectiveness of the learning mechanism due to following 2 criterias:

1.The size of underlying search space prohibits a systematic consideration of alternatives. The learning mechanism must posses heuristic methods for reducing the scope of the search.

2.This reduction must not affect the reachability of good points in the space.

Second point argues against excessive reliance of the learning mechanism on domain specific criteria and assumptions. Given the inherent limitations of relying on domain specific strategies for exploration in the representation space, the problem tries to find a domain independent search technique. One example of such a mechanism can be found in nature where the evolutionary process rapidly yields structures (=organisms) that are highly adapted to the specific environmental niches or locations in which they exist.

Genetic Adaptive Algorithms:

These algorithms are motivated by standards models of heredity and evolution in the field of population genetics and embody abstractions of the mechanisms of adaptation present in natural systems. A Genetic Algorithm simulates the dynamics of population genetics by maintaining a knowledge base (=population) of structures (=individuals) that evolves over time in response to the observed performance (=fitness) of its structures in their operational environment. The search proceeds by repeatedly selecting structures from the current knowledge base on the basis of the associated utility measures derived via interpretation and applying idealized genetic operations to these structures (=offsprings) for evaluation.

Explanation on step.3:

If m structures are selected to participate in the generation of new structures at time t, expected number of structures to be derived from any given structure in the current knowledge base is:

$$m \cdot \frac{u[s(i,t)]}{\sum_{j=1}^{m} u[s(j,t)]} = \frac{u[s(i,t)]}{\hat{u}[S(t)]}$$

where $\hat{u}$ denotes average utility of its argument.

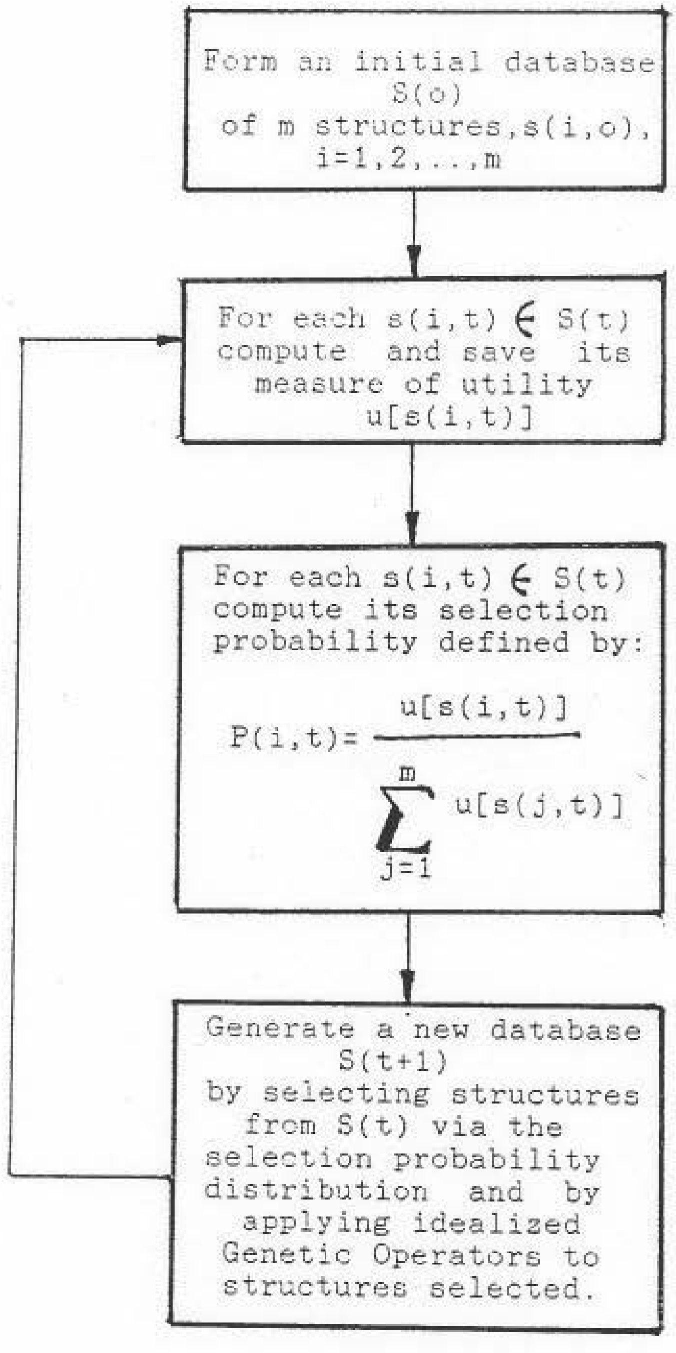
Therefore the probability impose selective bias toward above average performing structures relative to the rest of the structures in Knowledge-Base. Such a selective pressure will cause the best performing structure in the initial Knowledge-Base to occupy a larger and larger proportion of the Knowledge-Base over time. Propagation of the best structures thru Knowledge-Base does nothing to further the search for better performing structures. They are performed by Genetic Search operators, transforming the structures selected from the current Knowledge-Base into new, untested ones.

4.2.6 ISSUES AT THE ORGANIZATIONAL LEVEL:

As the number of the knowledge sources increases incompleteness and inconsistency comes into stage. Moreover, the nature of inconsistencies in the sum of the knowledge may allow judgments of relative reliability.

In a situation when multiple agents are problem-solving in parallel, each may generate new knowledge, each agent may have to communicate all it knows, or make its knowledge available in a global knowledge base. The first one may quickly result in the saturation of the broadcast medium, while the latter may saturate the storage device and become an access bottleneck. Alternatively, an agent may not want all of the knowledge he generates to be made available to the entire organization. In any of these cases, it is clear that one agent's model of the world may rapidly become incomplete. Techniques are required to decide:

- What knowledge to store locally
- What knowledge to communicate
- To whom to communicate this knowledge
- How to restrict access by other agents


Form an initial database S(o) of m structures, s(i,o), i=1,2,..,m
For each s(i,t) ∈ S(t) compute and save its measure of utility u[s(i,t)]
For each s(i,t) ∈ S(t) compute its selection probability defined by:
$P(i,t)=\frac{u[s(i,t)]}{\sum_{j=1}^{m} u[s(j,t)]}$
Generate a new database S(t+1) by selecting structures from S(t) via the selection probability distribution and by applying idealized Genetic Operators to structures selected.

. Where to look for knowledge if you don't have it

In the case where knowledge is shared among multiple agents and stored locally, an agent may extend or alter the knowledge in a manner that makes it consistent with another agent's understanding of the same knowledge. For example, an agent may share a description of a person with other agents,but during it's problem solving it may alter its description (ex. guessing of the person's age) so that it is inconsistent. At some time during problem solving, another agent may request that person's description from other agents and receive differing descriptions. How is this inconsistency to be solved? It is not enough to maintain separate models; this only maintains a separation. [15]

4.3 ES APPROACH TO MANAGEMENT:

What we need to have for making quick and consistent decisions, are easy accessible knowledge base systems, because today's rapidly growing and changing business world making qualified decisions is indisputably important. How do we provide this easy accessible data base or knowledge base and what does it bring to the organization. The first question was discussed in the previous chapter. In the second question advantages coming along with ES will be discussed in the next parts of this chapter. Besides advantages, if we try to see disadvantages of ES we can see highly complicated information system which requires very high-skill, well oriented workers; therefore, running ES requires much more effort rather than building ES. But every rose has its thorn. We can earn lots of things with ES such flexibility, consistency and certainty.

4.3.1 Flexibility and Stability:

The most important basic elements of any viable systems are flexibility and stability. The organizations in this instantly changing business world can not be viable in this environment without flexibility and stability. These two terms come along with some questions such as, how organizations can reach (or achieve) flexibility and stability at the same time. What are flexibility and stability?

Flexibility is defined as the ability of the system to modify its informational (or knowledgeable) base[ ], and stability is defined as the ability of the system to maintain its essential informational (or Knowledgeable) content[ ]. According to this definition, any principle of systematic organization assuming such dominance and pervasiveness in the natural world provides an appealing model from which to design those social systems over which man as a rational being has some control. In its most generalized form it should be expected that any social system which embodies analogue mechanism for stable and flexible (conservative and liberal) control will fare well in the process of social evolution. By establishing governing instructions which explicitly acknowledge the need for both stability and conservation of the traditional principles of social organization, on the one hand, and a willingness to change the system's information as is demanded by a changing world we will be incorporating one of the most fundamental principles of the systematic control as found in the major systems of nature into our social systems.

In the question of the method of achieving flexibility and stability, existence of any method or tool can be discussed. In this point the existence of balance between flexibility and stability is obviously indisputable and not only the existence of balance but also establishing and controlling this balance are important for viable organizations.

Earlier, in the definitions of the flexibility and stability, informational base and essential informations were discussed as a basic structure which should have a dynamic flexible shape. It means informational or knowledgeable base has the main step to reach (or achieve) the goal of the flexibility or stability in the organization.

4.3.2 Consistency and General Approach to Uncertainty Mgmt.

It's not always possible to provide the consistency of the knowledge that is produced by the organizations; therefore, we have to define the uncertainty in the management structure.What we need to confirm, when we make a decisions, is definition of goals and knowledge sources. The claim of our work is that the paradigm of DPS [ ](Distributed Problem Solving) is well suited to object recognition using multiple KS's. Various KSs communicate with each other in a team-like fashion. KSs mutually influence each other in the process of disambiguation of their results. A final decision based on the mean value of their Confidence Factor Value [ ](CFV) and degree of organization in the CFV is made

In the process of uncertainty management disbelief a considerable importance is also given. In some research a certainty factor is computed by subtracting disbelief value from belief value for a hypothesis and is used by in later calculations. This implies that it emphasizes the net difference between supporting and opposing bodies of evidence

This net differences may mislead by hiding much necessary information. this situation of uncertainty management comes along with some problems described as below.

i-Modyfying belief value or disbelief value of an individual expert for each alternative based on the disbelief value or belief values respectively of the same expert for other alternatives in the process of competition between hypothesis.

ii-Computing the CFV of each expert for each alternative based on the modified belief value and the modified disbelief value of the alternatives

iii-Updating the CFV of an expert for an alternative based on the CFVs of other experts for same alternative, process called cooperation and

iv-Combining the CFVs of all the experts to give a final value for each alternative and deciding the best alternative based on the associated CFV.

In the study of LES (Lockheed Expert System) a new approach to find and define uncertainty was evaluated. This approach assumes that same problems and gaps in the knowledge base and it tries to find those problems and gaps.

The Lockheed Expert System is a Generic Rule-Based expert system tool that has been used as a framework to construct expert systems in many areas such as electronic equipment diagnosis, design checking, photo interpretation and hazard analysis. LES employs a combination of goal-driven and data-driven rules with the latter being attached to the factual database. One of the aids that LES uses is the knowledge base completeness and consistency verification program called CHECK.

According to the system, potential problems are as defined. By statistically analyzing the logical semantics of the rules represented in LES's case grammar format, CHECK can detect redundant rules conflicting rules, rules that are subsumed by other rules and circular-rule chains. The following definitions for these fair potential problems are used in CHECK.

Redundant Rules:

Two rules succeed in the same situation and have the same results. In LES, this means that the IF parts of the two rules are equivalent and one or more THEN clauses are also equivalent. Because LES allows variables in rules, equivalent means that the same specific object names can match their corresponding rule "p(x)-->q(x)" is equivalent to the rule "p(y)-->q(y)", where x and y are variables

Conflicting Rules:

Two rules succeed in the same situation but with conflicting results. In LES, this means that the IF parts of the two rules are equivalent, but one or more THEN clauses are contradictory, or one pair of IF clauses is contradictory while they have equivalent THEN clauses. For example, the rule "p(x)-->not(q(x))" is contradictory to the rule "p(x)-->q(x)".

Subsumed Rules:

Two rules have the same results, but one contains additional constraints on the situations on the situations on which it will succeed. In LES this means one or more THEN clauses are equivalent, but the IF part of one rule contains fewer constraints and/or clauses than the IF part of the other rule. For example, the rule "(p(x) and q(y))-->r(z)" is subsumed by the rule "p(x)-->r(z)".

Circular Rules:

A set of rules is a circular-rule set if the chaining of those rules in the set forms a cycle. For example, if we had a set of rules as follows: (1) "p(x)-->q(x)", (2) "q(x)-->r(x)", (3) "r(x)-->p(x)" and the goal is r(A), where A is a constant, then the system will enter an infinite loop at run time, unless the system has a special way of handling circular rules.

Potential gaps in a knowledge base:

The development of a knowledge based system is an iterative process in which knowledge is encoded, tested, added, changed and refined. This iterative process often leaves gaps in the knowledge engineer and the expert may have overlooked during the knowledge acquisition process. In the LES there is found three situations indicative of gaps in the knowledge base. These situations called (1) missing rules, (2) unreachable clauses, and (3) deadend clauses are described below:

Missing Rules: A situation in which some values in the set of possible values (called legal values) of an object's attribute are not covered by any rule's IF clauses (ie. the legal values in the set are covered only partially or not at all). A partially covered attribute can prohibit the system from attaining a conclusion or cause it to make a wrong conclusion when an uncovered attribute value is encountered during its run time.

Unreachable Clauses:In a goal driven system, a THEN clause of a rule should either match a goal clause or match an IF clause of another rule (in the same rule set). Otherwise, the THEN clause is unreachable.

Deadend Rules:To achieve a goal (or subgoal) in LES, it is required that either: (1) the attributes of the goal clauses are askable (user provides needed information) or (2) that the goal clause is matched by a THEN clause of one of the rules in the rule sets applying to that goal goal. If neither of these conditions is satisfied then the goal clause can not be achieved, ie. it is a "deadend clause". Similarly, the IF clauses of a rule also must meet one of these two conditions, or they are "deadend clauses".

By evaluating these problems and gaps CHECK generates a dependency chart which shows the interactions among the rules and between the rules and the goal clauses. An example of a dependency chart for a small problem is shown in figure[ ] A "*" indicates that one or more clauses in the IF part of a rule or a goal clause matches one or more clauses in the THEN part of a rule. The dependency chart is very useful when the knowledge engineer deletes, modifies, or adds rules to the rule base.

Finally, as a by-product of the rule checking processing, CHECK generates a dependency chart which shows how to rules couple and interact with each other and with the goals; this chart should help the knowledge engineer to identify immediately the effects of deleting, adding, or modifying rules.(See figure 4.3)[ ]

4.4 LOOKING FORWARD IN EXPERT SYSTEM TECHNOLOGY:

Expert Systems hold great promise for technical application areas such as medical diagnosis or engineering design. They are, we argue, less promising for management applications in today's world. The reason is that managers are not experts[ ] in the sense of possessing a formal body of knowledge which they apply. The limitations of Artificial Intelligence approaches in managerial domains is explained in terms of semantic change,motivating attention toward management (decision) support system.

AI is getting market appeal. Expert Systems, robotics and 5th generation technology are getting serious recognition in actuality.The attempt here to assess the potential impact of AI future technology on commercial organization and other social institutions. Technology assessment suffers the lack of a convincing methodology. Hence the strategy here is not to try to predict the actual course of AI innovations, but rather consider what would be the theoretical limits to the technology.

The concern is mainly with AI technology in organizations, i.e., with groups of people working on cooperation. These remarks are not intended to apply to industrial robots, nor to single user expert systems, but rather to what might be called a "Knowledge Based Information System". Such applications would seem to be the eventual result of a convergence of data base management with AI knowledge representation. In this point we may think what such a KBIS could do. if we think radically, we can easily see, super-powerful KBIS may eventually eliminate the need for management in the future of unlimited applications of Expert systems.

Bibliography

[1] Archibald, I. G., Craig, J. A., and Redfearn, J. J., "Bridging the generation gap: Expert systems as part of large scale computer systems," in R&D Management, Vol. 15, No. 2, 1985.

[2] Badiru, Adedeji B., "Successful Initiation of Expert Systems Projects," in IEEE Transactions of Engineering Management, Vol. 35, No.3, August 1988, pp. 186-190.

[3] Basden, Andrew, "On the Application of Expert Systems," in Developments in Expert Systems. New York, Academic Press Inc., 1984, pp.59-75

[4] Bernold, T., "Aspects of technology assessment - the future role of artificial intelligence," in R&D Management, Vol. 15, No. 2, 1985, pp. 179-182.

[5] Brodie, Michael, "Large-Scale Knowledge-Based Systems: Concluding Remarks and Technological Challenges," in On Knowledge Based Management Systems. New York, Springer-Verlag, 1986, pp. 579-558.

[6] Carey, Michael J. and Dewitt, David J., "Extensible Databases Systems," in On Knowledge Based Management Systems. New York, Springer-Verlag, 1986, pp. 315-336.

[7] D'Agapayeff, A., "A short survey of expert systems in UK business," in R&D Management, Vol. 15, No. 2, 1985, pp. 89-99.

[8] Davis, D., "Artificial Intelligence Goes to Work," in High Technology, April 1987, Vol. 7, No. 4, pp. 16-25.

[9] Dhar, Vasant "On the Plausibility and Scope of Expert Systems in Management," in Journal of Management Information Systems, Vol. 4, No. 1, Summer 1987, pp. 25-41.

[10] Ensor, Robert J., Grabbe, John D., "Transactional Blackboard," Proceedings of Annual Expert System Conference, 1987, pp. 340-347.

[11] Ferrante, Richard D., "The Characteristic Approach to Conflict Resolution," in Proceedings of Annual Expert System Conference, 1987, pp. 331.

[12] Fersko-Weiss, Henry, "Expert Systems Decision-Making Power," in Personal Computing, November 1985, pp. 97-105.

[13] Fikes, R. and Keller, T., "The Role of Frame-Based Representation in Reasoning," in Communications of the ACM, Vol. 28, No. 9, 1985, pp. 904-920.

[14] Fordyce, Kenneth et al., "Review of Expert Systems for the Management Science Practioner," in Interfaces, Vol. 17, No. 2, March-April 1987, pp. 64-77.