Machine Learning and Its Applications In Reliability Analysis Systems

Miss Hui-ling Hong

A thesis submitted for the degree of Master of Science

Computer Science Department
University of Durham
U.K.

September 1994

Machine Learning and Its Applications In Reliability Analysis Systems

Hui-ling Hong

University of Durham

Machine Learning and Its Applications In Reliability Analysis System

Presented by Miss Hui-ling Hong

A thesis submitted for the degree of
Master of Science in Application of Artificial Intelligence in Engineering

ABSTRACT

In this thesis, we are interested in exploring some aspects of Machine Learning (ML) and its application in the Reliability Analysis systems (RAs). We begin by investigating some ML paradigms and their techniques, go on to discuss the possible applications of ML in improving RAs performance, and lastly give guidelines of the architecture of learning RAs. Our survey of ML covers both levels of Neural Network learning and Symbolic learning. In symbolic process learning, five types of learning and their applications are discussed : rote learning, learning from instruction, learning from analogy, learning from examples, and learning from observation and discovery.

The Reliability Analysis systems (RAs) presented in this thesis are mainly designed for maintaining plant safety supported by two functions : risk analysis function, i.e., failure mode effect analysis (FMEA) ; and diagnosis function, i.e., real-time fault location (RTFL). Three approaches have been discussed in creating the RAs. According to the result of our survey, we suggest currently the best design of RAs is to embed model-based RAs, i.e., MORA (as software) in a neural network based computer system (as hardware). However, there are still some improvement which can be made through the applications of Machine Learning.

By implanting the 'learning element', the MORA will become learning MORA (La MORA) system, a learning Reliability Analysis system with the power of automatic knowledge acquisition and inconsistency checking, and more. To conclude our thesis, we propose an architecture of La MORA.

ACKNOWLEDGEMENT

There are so many people that I greatly owe my acknowledgment since they have been directly or indirectly assisting in producing this thesis. Most of them are names of authors listed in bibliography whereas many of the ideas present in this thesis are inspired by them, and others are people with whom I discuss about my research work. Among them are friends and lab-mates whose comments were so helpful which have resulted in considerable, sometimes spectacular, improvement.

Also, I would like to delegate my most sincere thanks to my supervisor Mr. Andrew Slade and Professor Keith Bennett, and to the University of Durham who provided abundant material essential to my research. In addition, I especially appreciate my supervisor for his free style supervision unrestraining my thinking and for his support in extending my study so I could plan my thesis in appropriate time.

Above all, thanks to those people who have been kind in offering me their support and encouragement during my study, who have provided me with the strength to face the most difficult time during the writing of my thesis. My gratitude towards them shall remain in my mind always.

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Artificial Intelligence

In the past, the design and maintain of an industrial system has mainly relied on human resources - the experts[1] - for performing the tasks in design verification and real time diagnosis of factory faults. Such skills have called upon the expert's knowledge and his/her abilities to plan complex actions, detect errors, and learn about the environment. Nevertheless, based on human intelligence, we have always enjoyed creating tools that take the muscle and repetition out of our working environment [Wright & Bourne 1988]. In this thesis, we intend to apply some Artificial Intelligence (AI) techniques in creating and in improving the performance of Reliability Analysis systems (RAs) used in an industrial aspect. In addition, we concentrate upon describing the current position of Machine Learning (ML) and we make a suggestion as to how this may be incorporated into existing RAs.

Reliability Analysis systems (RAs) are systems designed as a tool to assist engineers in verifying system design (i.e. failure mode effect analysis function) and in diagnosing of factory faults (i.e. real-time fault location function) :

1. The failure mode effect analysis function (FMEA) helps design engineers detect or simulate possible dangers caused by inappropriate plant design, therefore increasing plant safety.

2. The real-time fault location function (RTFL) aims at reducing the cognitive load of an on-line operator usually by helping to diagnose the cause of alarms and possibly by suggesting corrective action.

Based on the conventional approach, the expertise, i.e., knowledge and the problem solving techniques, used by human experts has to be elicited from humans and transferred into a computer-based system, e.g., RAs. Therefore, the RAs should, at best,

---

[1] The "experts" is a general term for representing many aspects of people required in industrial system, e.g., design engineer, on-line operator, diagnosis expert, etc.

possess equivalent knowledge as the experts mentioned above. In that sense, the establishing of RAs can prevent loss of expertise through their death, retirement, or their job transfer of human experts. Furthermore, RAs can be used by the experts themselves as an auxiliary tool, also the systems can be used for training in order to pass the expertise along to novices.

In Artificial Intelligence (AI), both symbolic processing (i.e. the conventional approach) and neural network approaches have been adopted to create the RAs. Though there are many successful applications, still there are several improvements that can be made by introducing Machine Learning (ML) techniques into these systems.

Our interest is focused on exploring some aspects of Machine Learning (ML) in Reliability Analysis systems (RAs). We begin by investigating some ML paradigms and their techniques, go on to discuss the applications and lastly give guidelines of the architecture of learning RAs.

1.2 Historical Remarks - An AI View

The following is a brief introduction to Machine Learning (ML) development through the history of Artificial Intelligence (AI). It has been divided into four periods, each centered around a different paradigm [Forsyth & Naylor 1986] :

1. **1950s - Neural networks.** In 1943, Warren McCulloch and Walter Pitts proposed a model of the neuron in the human and animal brain [Forsyth & Naylor 1986]. These abstract nerve cells provided the basis for a formal calculus of brain activity. Other workers, notably Norbert Wiener, elaborated these and similar ideas into the field that became known as Cybernetics; and it was from Cybernetics that AI emerged as a scientific discipline in the 1950s. The term "Artificial Intelligence" was first introduced by John McCarthy, in the Dartmouth Summer Conference (1956). Although both disciplines share the equal opportunities in creating intelligent systems, to distinguish them, we quote from Alain Bonnet [Bonnet 1985] :

> *The study of Cybernetics is concerned with the mathematical properties of feed-back systems and treats the human being as an automation, whereas AI is concerned with the cognitive processes brought into play by the human being in order to perform what we regard as intelligent tasks.*

During this period, the research interest was in building general purpose learning systems that start with little or no initial structure or task-oriented knowledge. The major thrust of research based on this *tabula rasa* approach involved constructing a variety of neural model-based machines, with random or partially random initial structure. These systems were generally referred to as neural nets or self-organizing systems. Learning in such systems consisted of incremental change in the probabilities that neuron-like elements (typically threshold logic units) would transmit a signal. Due to the primitive nature of computer technology at that time, most of the research under this paradigm was either theoretical or involved the construction of special purpose experimental hardware systems, such as Perceptrons, Pandemonium, and Adelaine. [Michalski, Carbonell & Mitchell 1984] Related research involved the simulation of evolutionary processes, that through random mutation and "natural" selection might create a system capable of some intelligent behaviour. Learning algorithms inspired by the evolution process are called genetic algorithms.

Rosenblatt's Perceptron [Michalski, Carbonell & Mitchell 1984] was an elementary visual system which could be taught to recognize a limited class of patterns. It consists of a finite grid of light-sensitive cells. This constitutes a miniature retina. In addition there are a number of feature-detecting elements - picturesquely called 'demons' - which monitor the state of groups of cells in the grid. They respond when characteristic subpatterns are presented by sending a signal to a higher-level decision-maker. The decision-maker multiplies each signal from a local demon by a positive or negative weighting factor and the resulting numbers are added. If the total exceeds a set threshold the Perceptron says 'Yes'; otherwise it says 'No'. Thus one perceptron can discriminate two classes of images. To recognize more patterns requires more Perceptrons - 26, say, for the letters of the alphabet. By adjusting the weightings attached to each demon, the Perceptron can be made to learn (in a sense).

Experience in the above areas spawned the new discipline of pattern recognition and led to the development of a decision-theory approach to machine learning. In this approach, learning is equated with the acquisition of linear, polynomial, or related forms of discriminate functions from a given set of training examples. One of the best known successful learning systems utilizing such techniques was Samuel's checkers program. [Michalski, Carbonell & Mitchell 1984] This program was able to acquire through learning a master level of performance. Somewhat different, but closely related, techniques utilized methods of statistical decision theory for learning pattern recognition rules.

In parallel to research on neural modeling and decision theoretic techniques, researchers in control theory developed adaptive control systems able to adjust automatically their parameters in order to maintain stable performance in the presence of various disturbances.

Practical results sought by the neural modeling and decision theoretic approaches met with limited success. High expectations articulated in various early works were not realized, and research under this paradigm began to decline. Theoretical studies have revealed strong limitations of the "knowledge-free" perceptron type learning systems (ibid.).

2. **1960s - Heuristic search**. AI workers abandoned the attempt to build artificial brains from the ground up. Instead they looked on human thinking as a complex coordination of essentially simple symbol-manipulating tasks, research mainly stemming from the work of psychologists and early AI researchers on models of human learning. The paradigm utilized logic or graph structure representations rather than numerical or statistical methods. Systems learned symbolic descriptions representing higher level knowledge and made strong structural assumptions about the concepts to be acquired. [Michalski, Carbonell & Mitchell 1984]

Examples of work in this paradigm include research on human concept acquisition, and various applied pattern recognition systems. Some researchers constructed task-oriented specialized systems that would acquire knowledge in the context of a practical problem. For instance, the META-DENDRAL program which generates rules explaining mass spectrometry data for use in the DENDRAL system. [Michalski, Carbonell & Mitchell 1984]

An influential development in this paradigm was Winston's structural learning system. In parallel with Winston's work, different approaches to learning structural concepts from examples emerged, including a family of logic-based inductive learning program (AQVAL), and related work by Hayes-Roth & McDermott. [Michalski, Carbonell & Mitchell 1984]

Here they were on firmer ground since computers can do things like searching, comparing symbols and so on, which they identified as the foundations of intelligent problem solving. The hard part was putting these simple activities together. The most influential workers at this time were Allen Newell and Herbert Simon of Carnegie-Mellon University, who worked on theorem-proving and computer chess, among other things. Their masterwork was a program called GPS, the General Problem Solver. GPS was general in so far as the user defined a 'task environment' in terms of the objects. However, its generality was restricted to puzzles with a relatively small set of states and well-defined rules. It could work on the Towers of Hanoi, cryptarithmetic and other problems of a similar nature. It functioned in formalized micro-worlds. What it could not do was solve what people would regard as real-life problems - e.g. Has this patient got cancer? Should I sell my shares now? Real problems are characterized by a lack of precise rules; but GPS and contemporary systems could only work in a very clearly defined environment. Another criticism of GPS was its reliance on what is now known as depth first search, which involves splitting larger problems into progressively smaller subproblems until one trivial enough to be solved directly is reached. It is an elegant idea, but it is not an optimal search strategy, since it can involve an unnecessarily thorough examination of unsuccessful pathways, requiring a lot of backtracking. The central idea behind GPS was that problem solving was a search through a space of potential solutions. To make the search efficient, it had to be guided by heuristic rules that directed it towards the desired destination. Thus, an automaton wandering around a maze would have to use an exhaustive search technique if it knew nothing about the structure of that maze; but if it had some way of telling when it was getting 'warm' it could normally reach its goal state sooner. (Not always, since heuristics are not guaranteed to work, and occasionally may lead it down a blind alley.) During this period, AI workers devised several heuristically guided search strategies, such as the A* algorithm, which are still valid. In addition, concepts such as list-processing were introduced into general computer practice. List-processing was once a specialist AI topic, motivated by the desire of AI programmers to handle diverse and flexible data structures, i.e., 'represent knowledge'. Now it is standard computing practice. This tendency for AI to 'export its successes' is remarkable, and continues to the present day. AI has always been a fertile breeding ground for new ideas, and if they are good ideas they soon cease to be regarded as belonging to AI. One of the best ideas that has spread outwards in this way is the notion of the expert system.

3. **1970s - Knowledge-base systems.** Researchers have broadened their interest beyond learning isolated concepts from examples, and have begun investigating a wide spectrum of learning methods, most based upon knowledge-rich systems. Specifically, this paradigm can be characterized by several new trends, including [Michalski, Carbonell & Mitchell 1984] :

(1) Knowledge-intensive approach : Researchers are strongly emphasizing the use of task-oriented knowledge and the constraints it provides in guiding the learning process. One lesson from the failures of earlier *tabula rasa* and knowledge-poor learning systems is that to acquire new knowledge a system must already possess a great deal of initial knowledge.

(2) Exploration of alternative methods of learning : In addition to the earlier research emphasis on learning from examples, researchers are now investigating a wider variety of learning methods such as learning from instruction, learning by analogy, and discovery of concepts and classifications.

(3) Incorporating abilities to generate and select learning tasks : In contrast to previous efforts, a number of current systems incorporate heuristics to control their focus of attention by generating learning tasks, proposing experiments to gather training data, and choosing concepts to acquire.

GPS, as we said, was not much good at real-life problems. In the 1970s a team led by Edward Feigenbaum at Stanford University began to remedy that defect. Rather than trying to computerize general intelligence, they focused on very narrow areas of expertise. Thus was the expert system born. The first expert system was DENDRAL, a mass-spectrogram interpreter built as early as 1967; the most influential has proved to be MYCIN which dates from 1974 (and was strongly influenced by DENDRAL). The problem addressed by DENDRAL is to take data about the fragmentation of an organic molecule provided by the mass-spectrogram and use it to infer the structure of that molecule. The knowledge used to guide the interpretation is of two kinds - knowledge about the chemical composition of the molecule, and knowledge about the way the chemical bonds break up within the instrument. Without the second kind of knowledge there could be literally millions of ways that the molecule might have been put together. MYCIN diagnoses bacterial infections of the blood, and prescribes drug therapy. It has spawned a whole family of medical diagnostic 'clones', some of which are in routine clinical use. For instance, PUFF, a lung-function diagnostic tool based on the MYCIN plan, is routinely employed at the Pacific Medical Center near San Francisco. MYCIN's importance lay in the introduction of various new features which have become the hallmarks of the expert system. Firstly, its 'knowledge' consists of hundreds of rules. Secondly, these rules are probabilistic. Shortliffe, the inventor of MYCIN who was also a doctor, devised a scheme based on certainty evidence. The significant point, however, is that MYCIN and systems like it can arrive at correct conclusions even with incomplete and partly incorrect information. They have some method of approximate reasoning - whether based on probabilities, Fuzzy Logic, certainty factors or some other likelihood calculus - for deriving a good estimate of the truth even from imperfect data. Thirdly, MYCIN can explain its own reasoning process. The physician using it can interrogate it in various ways, either to ask how it reached a particular conclusion or why it is requesting a certain item of information. The system answers by retracing and describing the deductive process that led to the current state. This degree of user-friendliness was essentially a by-product of the rule-based style of programming. Today hardly anyone doubts that the more important the task a computer system performs, the more necessary that it can explain and justify its own behaviour to the users. The fourth, and crucially, MYCIN works. It does what requires a human years of training. In fact, MYCIN is more used in teaching than diagnosis but the point is that large corporations, governments and the media are all becoming interested. One of MYCIN's successors, the PROSPECTOR geological exploration system, has been widely quoted as helping to discover a vast unknown molybdenum deposit in Washington state. It is early days yet but corporate America has scented the sweet smell of profits. AI has lost its innocence.

4. **1980s - now, Machine learning.** Expert systems are in fashion, and their magic ingredient is knowledge. For it is the scope and quality of its knowledge base that determines the success of an expert system. But knowledge is not something you can squeeze into a computer program like toothpaste from a tube. In fact it is often harder to quarry out of the unyielding rockface of ignorance than that famous molybdenum deposit! Codifying a human expert's skill can be a long and labour-intensive process. So while the world is marveling over expert systems, AI has moved on to concentrate on the problem of machine learning - which is one way of synthesizing knowledge automatically. AI always has been a moving target, and at the centre of that target right now is a program called EURISKO. EURISKO is a discovery program which extends and improves its own body of heuristic rules automatically, by induction. Apart from winning the 'Trillion Credit Squadron' naval wargame three years in succession (despite rule changes intended to stop it) EURISKO has also been applied to practical problems. One result was the invention of a novel three-dimensional AND/OR gate in the field of integrated-circuit design. Indeed, EURISKO is thought to be the first computer program holding a patent, though most of the credit rightly belongs to its author, Doug Lenat. There can be little doubt that systems like EURISKO represent the leading edge of AI research. And since AI itself can be viewed as a leading branch of computer science, this is the place to look for a peek at one future of computing. Ironically enough, by concentrating once again on learning AI has returned to its roots, because learning was seen as the key problem in the early cybernetic days. A lot of silicon has flowed under the bridge since then, however, and the present attempts to build systems that can improve their problem-solving abilities have a higher chance of success.

Apart from the resurgence of interest in Machine Learning through the influence of its application in knowledge acquisition of expert systems, other interesting researches evolving from this time are the development of second generation of expert systems and the rapid growth of neural network application.

Different from MYCIN-like systems, the second generation of expert system was first introduced by Luc Steels in 1985. Where as first generation systems rely purely on heuristic knowledge in the form of rules (shallow knowledge), second generation systems have an additional component in the form of a deeper model which gives them an understanding of the complete search space over a deeper model (deep knowledge). The introduction of a deeper model solves a number of fundamental problems of shallow knowledge-based expert systems [Steels & Velde 1989] :

1. Shallow systems can only solve a very narrow range of predefined problems because they possess no general knowledge concerning the domain, limiting their reasoning ability greatly. Second generation systems fall back on search (which is not knowledge-driven) and therefore potentially very inefficient. However, because these traditional search techniques can theoretically solve a wider class of problems , there is a graceful degradation of performance instead of an abrupt failure.

2. First generation systems base their explanations purely on a backtrack of the heuristic rules that were needed to find a solution. It is well known however that the path followed to find a solution usually differs from a convincing rational argument why the solution is valuable, particularly if a lot of heuristic knowledge entered into the reasoning process. Because second generation systems have access to a deeper understanding of the search space, they can formulate a deeper and more convincing explanation which goes beyond the mere recall of which rules fired.

3. The most important advantage lies however in knowledge acquisition. Finding heuristic rules has turned out to be extremely difficult. Experts typically take a long time to come up with solid rules, the rule-set never seems complete, is continuously changing, and shows inconsistencies across experts (and even within the same expert). These inconsistencies are apparently due to different experiences which are the source of heuristic rules discovery. Second generation expert systems constitute a major jump forward in current technology because they exhibit learning behaviour in the sense that they are capable of acquiring new heuristic rules.

1.3 Thesis Outline

The current chapter is an introduction to the development of intelligent systems.

The second chapter is an introduction to machine learning. It is subdivided into three sections : an overview of machine learning; the learning model; existing machine learning techniques and their applications.

The third chapter is a general introduction to Reliability Analysis systems (RAs) covering both symbolic and neural network approaches.

In chapter four, we discuss the applications of machine learning in the RAs in a more general fashion.

The fifth, and final, chapter is a conclusion of our survey and further works.

Apart from the first four main chapters and the conclusion, a references is appended at the end of the thesis.

CHAPTER 2

MACHINE LEARNING AND ITS APPLICATIONS

2.1 Introduction

Learning is an essential component of all intelligent systems. As we know, no matter how smart the system is, if it cannot learn (adapt to the changing environment) then it will rapidly become out of date. This applies particularly to computer systems. Basically, there are two main reasons to carry out machine learning research defined by Herbert A. Simon [Michalski, Carbonell & Mitchell 1984] :

1. Directly getting computers to be smart and learn things by themselves so that human beings do not have to reprogram them.

2. By using computer systems to simulate human learning, we might be able to find out how humans work and perhaps this can help us to improve our learning techniques. Furthermore, we might even find out better learning strategies than present human learning techniques.

In chapter one, we have briefly encountered some Machine Learning (ML) systems through the introduction of ML history (summarized in Fig. 2.1-1). One could find that ML can be viewed as many phase phenomena. Some people defined learning as : "learning is constructing or modifying representations of what is being experience," (Ryszard S. Michalski) while others say (Herbert A. Simon) [Rich & Knight 1991] :

> *Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more efficiently and more effectively the next time." As thus defined, learning denotes two forms : skill refinement, and knowledge acquisition.*

1. Knowledge acquisition (symbolic level) is defined as learning new symbolic information coupled with the ability to apply that information in an effective manner. For example, acquisition of new declarative knowledge, organization of

new knowledge into general, effective representations, and the discovery of new facts and theories through observation and experimentation.

2. Skill refinement (neural net level) is the gradual improvement of motor and cognitive skills through practice. For example, learning to ride a bicycle.

Differing in many ways from knowledge acquisition, skill refinement occurs at a subconscious level by virtue of repeated practice whereas the essence of knowledge acquisition may be a conscious process whose result is the creation of new symbolic knowledge structures and mental models. Most human learning appears to be a mixture of both activities. For example, learning to drive a car from a book or from an instructor is within "knowledge acquisition" domain whereas learning by practicing driving a car on the road is under the domain of "skill refinement", however, while we were practicing on the road, during the same time, we apply our knowledge to adjust our driving skill.

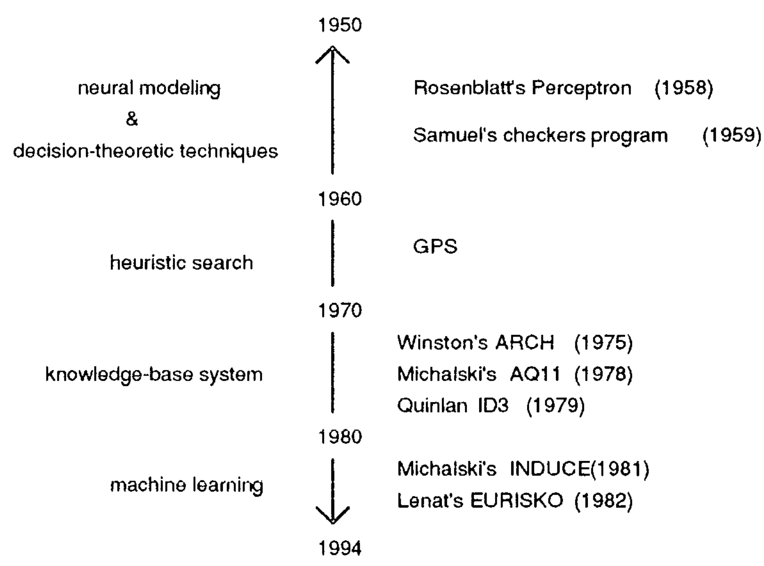

Fig. 2.1-1 Diagram of Machine Learning history

2.2 Learning Models

In the following, we are going to present two levels of learning model, i.e., model for conscious level and neural network model.

2.2.1 Learning Model of Conscious Level

There are six major elements composing the conscious learning model (see Fig. 2.2.1-1) : the environment, the learning element, the knowledge base, the effectors, the receptors, and the problem solver.

1. The environment : The input to the learning system is from the environment in which the learner currently finds himself. In machine learning this will be the problem domain.

2. The learning element : The learning element acts as a pattern recognizer. It is an interface between the problem solver and the knowledge base. The way that learning element consults the knowledge base is known as the learning skills. Many techniques adapted in Artificial Intelligence society are to simulate those learning skills, e.g., rote learning, learning by advice taking, etc.. This will be discussed later. Above all, there are a number of inference techniques applied by human beings, i.e. induction, deduction, abduction and creation.

(1) Deduction : The process of reasoning in which a conclusion follows necessarily from the stated premises; inference by reasoning from the general to the specific.

(2) Induction : A principle of reasoning to a conclusion about all members of a class from examination of only a few members of the class; broadly, reasoning from the particular to the general.

(3) Abduction : This is a form of deductive logic which provides only a 'plausible inference'. Using statistics and probability theory, abduction may yield the most probable inference among many possible inference.

3. The knowledge base : Here is where all knowledge has been kept, it captures the expertise of problem solving. There are several different ways of representing knowledge, we listed some out here : parameters in algebraic expressions, decision trees, formal grammars, production rules, formal logic-based expressions and related formalisms, graphs and networks, frames and schemes, computer programs and other procedural encodings, taxonomies, and multiple representations. [Michalski, Carbonell & Mitchell 1984]

4. The effectors : According to the solution generated from problem solver, the effectors produce system output. The interaction between system response and the environment will generate new stimula. So, the learning cycle will be initiated again.

5. The receptors : The receptors will decode received stimula and transmit the signal to the problem solver.

6. The problem solver. The problem solver collecting all the signals send from the sensors and integrates them into a pattern -- problem formed. This pattern then passed on to learning element for matching the solution from the knowledge base. If the solution is found, then it will be returned to the problem solver to generate the output pattern - solution found. The solution is then sent to effectors for producing system response. Otherwise, the learner will inform problem solver to request more information (e.g., input more knowledge) or suspend the problem and do something else.

Three of the components : the learning element, the problem solver, and the knowledge base are actually constructing within the CNS where the CNS stands for the central nervous system, details will be discuss in the next section.

The reason that the model is classified as conscious level is because the problem solving procedure usually involves the application of symbolic knowledge. For example, imagine that we are attending a mathematics examine. All the questions are written in a paper. Firstly, We perceive the question with our eyes (stimulus received by the receptors and coded into system signals, and the signals will be sent to the problem solver) then we will try to search our memory for the solutions (the problem solver sends the request to learning element - please find the solution for the question). If the learning element succeed in finding the answer, it will inform the problem solver to write it down on the paper. Finally, the problem solver will ask our hand to write down the answer on the answering paper, at the same time, it will also order our eyes to read. The answer that has been written down will then become a new stimulus for the system to check whether the writing is correct or not, and so on. However, if the learning element fails to find the solution for the question, the learning element might request the problem solver to go on trying the next question (depends on a particular situation), otherwise, the learning element could ask for more input, i.e., read the question again, or try hard to think of other way of answering the question, e.g., this could mean to modify the input patterns.

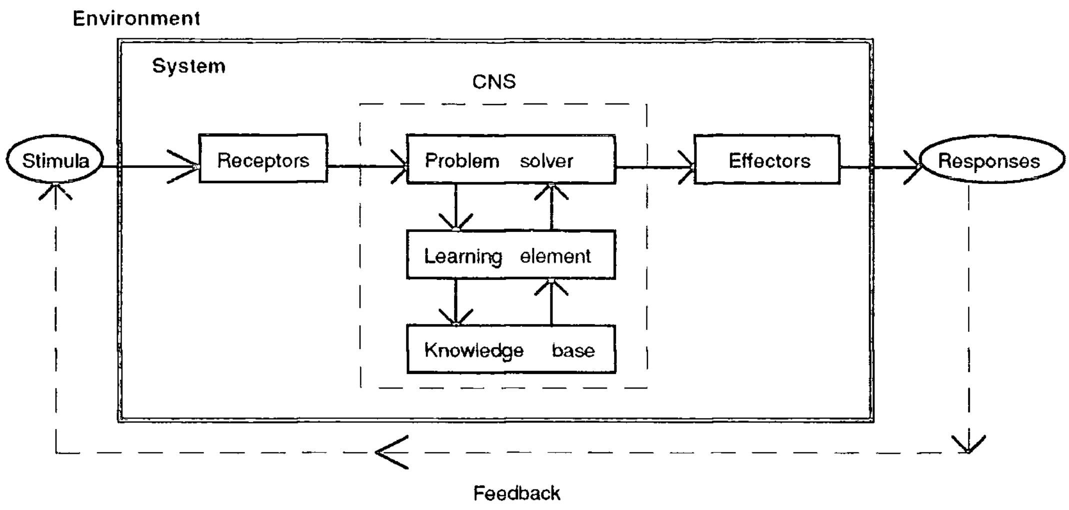

Fig. 2.2.1-1 Learning model of conscious level

2.2.2 Neural Network Model

As current Artificial Neural Networks (ANNs) are patterned after real neural networks, firstly we will briefly view the biological Neural Network model and then go on to describe ANNs.

2.2.2.1 Biologic Neural Network Model

Millions of receptors in our bodies continually monitor changes in our external and internal environment. Hundreds of thousands of cells called motoneurons (effectors) control the movement of our muscles and the secretion of our glands. In between, an intricate network of billions of cells called neurons (see Fig. 2.2.2.1-2) continually combine the signals from the receptors with signals encoding past experience to barrage the motoneurons with signals which will yield adaptive interactions with the environment. This network is called the Central Nervous System (CNS, see Fig. 2.2.2.1-1) [Arbib 1972] whereas the adaptivity is defined as learning.

Based on the theory of Token Physicalism[1], the conscious learning is actually an event of neural network activities. It can be proved by comparing the figures between Fig. 2.2.1-1 and 2.2.2.1-1. However, the neural network model are more comprehensible since it also deal with non-symbolic knowledge.

---

[1] Token physicalism is the thesis that mental events are physical events. In huamn beings, they are presumably neurological event.

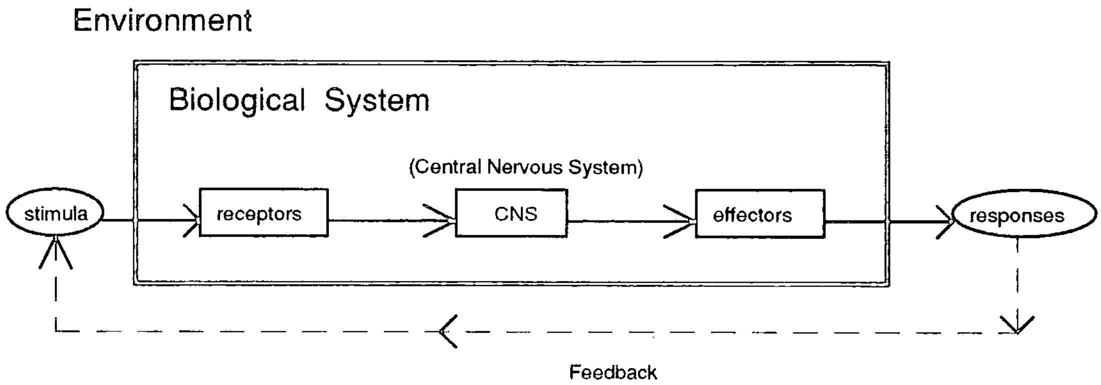

Fig. 2.2.2.1-1 Biological Neural Network model

There are billions neurons classified into thousands of different types. One basic scheme of biological neurons is shown in Fig. 2.2.2.1-2 which is composed of four major elements :

1. Synaptic buttons or synapses which serve as output devices.

2. The cell body which sums the membrane potentials provided by the synapses and fires at a rate which is a non-linear function of the total voltage.

3. The axon which carries the electrical signal from the cell body to subsequent synapses.

4. Dendrites, branch-like structures which provide sensory input to the cell body.

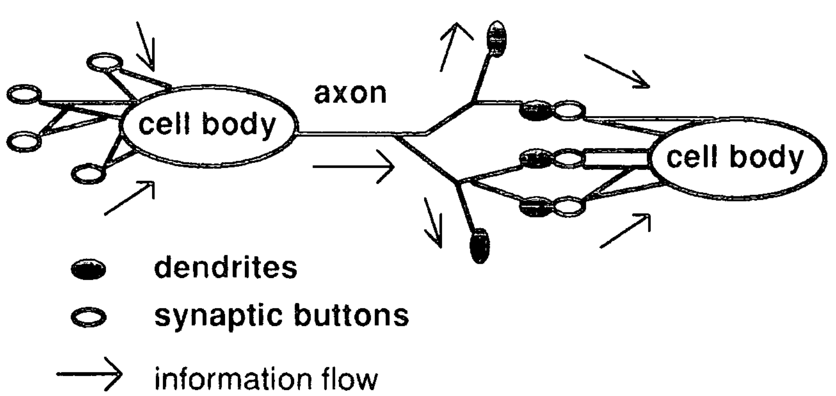

Fig. 2.2.2.1-2 Biological Neuron Model

We can best imagine the flow of information as shown by the arrows in the diagram, for although "conduction" can go either direction on the axon, most synapses tend to "communicate" activity to the dendrites or body of the cell they synapse upon, whence activity passes down the axon to the synaptic buttons. To understand more about this "communication," we must consider the cell as a living creature enclosed by a membrane across which there is a difference in electrical charge. Two forms of the information flow presented as followed [Arbib 1972] :

1. Passive flow - inhibitory : If we change this potential difference between the inside and outside, the change can propagate in much the same passive way that heat in conducted down a rod of metal. The change in temperature can propagate to other part of the rode, but as it moves further and further away from the point at which heat is applied, so does the temperature change decrease. In the same way, a normal change in potential difference across the cell membrane can propagate in a passive way so that the change occurs later, and is smaller, the further away we move from the site of the original change.

2. Activity flow - excitatory : If the change in potential difference is large enough (we say it exceeds a threshold), then in a cylindrical configuration such as the axon, a pulse can be generated which will actively propagate at full amplitude instead of fading passively. To understand this, think of a metal rod coated with gunpowder. If we heat the rod fairly gentle, the gunpowder will not ignite, and the propagation of the temperature difference will be passive and fading. However, if we exceed the ignition temperature of coating in hearing one end of our rod, that segment of coating will burn spontaneously and will be hot enough to ignite the neighboring segment of gunpowder coating, and so on, all the way down the bar. This is the case that we supply serves to trigger a "regenerative" process, a chain reaction which supplies its own energy so that once we have triggered the reaction at one place, it serves to unlock the energy stored in the next place, which then trigger the following place, and so on and so forth. So it is with cylinders of membrane.

Thus if the various potential differences on the dendrites and the body of neuron cell yield, usually by passive propagation, a potential difference across the membrane at the axon which exceeds a certain threshold, then a regenerative process is started - the electrical change at one place is then enough to trigger this process at the next place, to yield and undiminishing pulse of potential difference propagating down the axon.

Consider our last example, it is better to compare our axon with a recoatable metal bar - we are to imagine that after an impulse has propagated along the length of the axon, chemical processes take place which are the equivalent of recoating the fuse. This functional equivalence does not mean that the change in the membrane actually takes the form of a recoating. There is thus a short refractory period, a period during which a new impulse cannot be propagated along the axon, while the chemical restoration takes place.

If we were to start an impulse at any one place on the axon, it would propagate in both directions. However, if we start the pulse at one end of the axon, it can only travel away from that end, since once a section has been triggered it becomes refractory until well after the impulse has passed out of range. An impulse traveling along the axon triggers off new impulses in each of its branches, which in turn trigger off impulses in their even finer branches. When an impulse arrives at one of the bottoms, after a slight delay it yields a change in potential difference across the membrane of the cell upon which it impinges. The membrane on the buttons is called the presynaptic membrane, and the membrane of the surface upon which the bottoms impinges is called the postsynaptic membrane.

Surprisingly, at most synapses the direct cause of the change in potential of the postsynaptic membrane is not electrical but chemical. However, the normal process is that the electrical pulse reaching the bottoms causes the release of a few little vesicles of a chemical called the transmitter substance, which then diffuses across the very small synaptic cleft (the gaps between presynaptic membrane and postsynaptic membrane) to the other side. It is the transmitter reaching the postsynaptic membrane that causes the change in polarization of these membrane. The transmitter substance may be of two basic kinds : either excitatory, that is, tending to move the potential difference across the postsynaptic membrane in the direction of the threshold, or conversely, inhibitory, that is, tending to move the polarity away from threshold.

We may think of the neuron's threshold as being normally constant, but after we have fired an impulse down the axon, the threshold increases enormously and then takes quite a while to return to normal. We thus introduce "absolutely" refractory period, when it is too improbable that the changes could exceed the raised threshold, and the "relatively" refractory period when an exceptionally strong level of input can trigger an axonal impulse. Clearly, though, there is no sharp border between the absolute and relative refractory periods.

2.2.2.2 Artificial Neural Network Model

In biological science, biological neural networks are composed of simple, tightly interconnected processing elements called neurons. The interconnections are made by outgoing branches, the axon, which form variable connections, synapses, with other neurons or with other tissues such as muscles or glands. Attempts to develop models of biological neural networks, called Artificial Neural Networks (ANNs) fall into two main categories :

1. In biological modeling the structure and function of real brains are studied in order to explain biological data on aspects such as behavior.

2. In technological modeling, the aim is to extract concepts from the biological networks with which the new computational methodologies can be developed.

To achieve the second goal, i.e., greater computational power, it is admissible to incorporate features in models belonging to the second approach, even if they are not neurobiologically established.

As formulated by Kohonen, ANNs are massively parallel interconnected networks of simple, usually adaptive elements and their hierarchical organizations which are intended to interact with the objects of the real world in the same way as biological nervous systems do. There are several names for ANNs, e.g., connectionist models, parallel distributed processing and neuromorphic systems.

Neural net models are specified by net topology, node characteristics and training or learning rules. The function and performance of neural networks are determined primarily by their pattern of connectivity. In this sense there are feed forward neural networks and networks also incorporating feedback loops. According to another classification, fully connected and sparsely connected neural networks can be distinguished. (In the former case processing elements or nodes are connected to all other elements of the network, in the later case they are linked only to a few others). A special case of sparsely connected networks is the networks where the nodes are locally connected, e.g., to their neighbors. Computational elements or nodes used in neural net models are usually characterized by an internal threshold or offset and by their transfer function type, which can be binary, linear or continuous-nonlinear. [Monostori & Barschdorff 1992]

Fig. 2.2.2.2-1 is an example of most common used feed forward network with connected nodes. Recall from section 2.2.2.1, we have make the comparison between neural network model and model of conscious level. We defined conscious level of learning is actually an event of neural network activities. Furthermore, if we compare three figures together : Fig. 2.2.1-1, Fig. 2.2.2.1-1 and Fig. 2.2.2.2-1, we will have a clear picture of how neural network conducting symbolic learning. The Internal Units in

Fig. 2.2.2.2-1 is equal to the learning element and the knowledge base in conscious model. Now we should have a clear picture of the relationship between model of conscious level and neural network model.

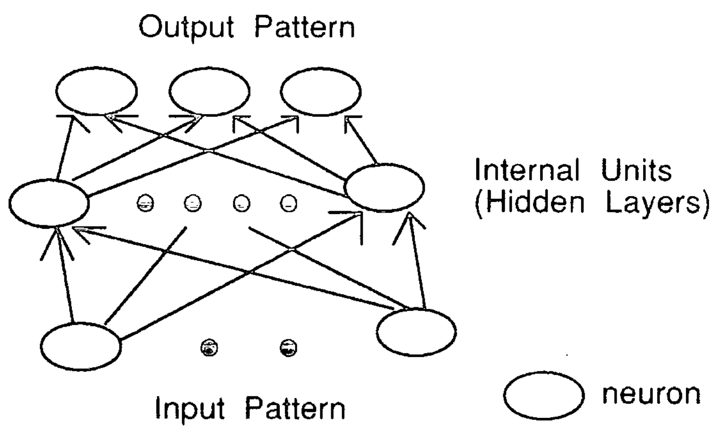

Fig. 2.2.2.2-1 Example of Artificial Neural Network structure.

ANNs have the following main characteristics :

1. Processing speed through massive parallelism.

2. Learning and adaptive ability by means of efficient knowledge acquisition and embedding.

3. Robustness with respect to fabrication defects and different failures.

4. Compact processors for space- and power-constrained applications.

One of the first abstract models of neurons was introduced by McCulloch and Pitts in 1943. Their model describes a neuron whose activity is the sum of inputs arriving via weighted pathways. The output signal is typically a nonlinear function of the neuron's activity. Many successors derived various neural network model based on their work. The following is a brief introduction of the history of ANNs [Monostori & Barschdorff 1992] :

1. The perception model of Rosenblatt which refers rather to a larger class of neural models. Rosenblatt expands the McCulloch and Pitts neuron with a learning process called back-coupled error correction, where the weights are adapted so that the actual output matches the target output imposed on the system.

2. Adaline (adaptive linear neuron) and madaline (many adaline) two level perceptrons of Widrow and Hoff with one and multiple outputs respectively, with the least mean squared or LMS error correction learning rule.

3. Multilevel perceptrons first described by Rosenblatt in 1962, who also introduced a probabilistic learning law, which anticipated the currently most frequently used back propagation learning algorithm for multilayered neural networks.

4. The "exclusive OR" problem described by Minsky and Papert, which proved that elementary perceptrons cannot distinguish between simple patterns, and to implement the exclusive OR function. Consequently a single McCulloch and Pitts neuron cannot act as a computationally universal element in the Turing sense.

5. The Hopfield net or crossbar associative network, which can be successfully used for optimization problems and as associative memory, gave a new impulse for the neural network research. This one layer, fully connected, binary net is adequate for supervised learning, which learning converges, when the initial weights are symmetric.

6. Back propagation learning algorithm for a class of multilayered neural networks, which through its simplicity made the neural network approach very popular and resolved doubts regarding the viability of the neural network approach completely. The algorithm is an extension of the LMS error correction learning rule for networks having hidden layers. It became welcome after the publication of Rumelhart et al. but was firstly developed by Werbos, and independently discovered by Parker.

7. Self organizing feature maps of Kohonen in which the input nodes are directly connected to the output nodes arranged in a two-dimensional grid and extensively interconnected with many local connections. After unsupervised, competitive learning the weights are organized such that topological close nodes are sensitive to inputs that are physically similar.

8. Adaptive resonance theory (ART) and the ANN models ART1 for binary input patterns and ART2 for binary and analog input sequences as well from Carpenter and Grossberg. These powerful networks were designed, in particular, to resolve the stability - plasticity problem, i.e., they are stable enough to preserve significant past learning, but remain adaptable enough to incorporate new information whenever it might appear. The minimal ART module incorporates a bottom-up competitive learning system resulting in quick recognition of learned patterns and preservation of adaptivity.

2.3 Learning Techniques and Their Applications

Our classification of learning is based on the underlying learning strategy, six types of learning are explored in this section with their applications [O'Shea, Self & Thomas 1987] : rote learning, learning by being told, learning from analogy, learning from examples (induction), learning from observation and discovery (unsupervised learning), and neural network learning. The first five learning techniques belongs to symbolic learning whereas the final one connectionist learning belongs to neural network learning. Following are the detail descriptions for the above learning techniques.

1. **Rote learning and direct implanting of new knowledge** : No inference or other transformation of the knowledge is required on the part of the learner. Variants of this knowledge acquisition method include : learning by been programmed, and learning by memorization of given facts and data, i.e., data caching. This technique has been applied in Samuel's checker program. Two characteristics involved in rote learning :

(1) Organized storage of information - in order for it to be faster to use a stored value that it would be to recompute it, there must be a way to access the appropriate stored value quickly.

(2) Generalisation - to keep the number of stored object down to a manageable level.

2. **Learning from instruction (or, learning by being told)** : Acquiring knowledge from a teacher or other organized source, such as a textbook, requiring that the learner transform the knowledge from the input language to an internally-usable representation, and that the new information be integrated with prior knowledge for effective use. Mostow describes a program called FOO, which accepts advice for playing a card game.

3. **Learning from analogy** : Acquiring new facts or skills by transforming and augmenting existing knowledge that bears strong similarity to the desired new concept or skill into a form effectively useful in the new situation.

4. **Learning from examples (a special case of inductive learning)** : Given a set of examples and counterexamples of a concept, the learner induces a general concept description that describes all of the positive examples and none of the counterexamples. Learning from examples can be subcategorized according to following sources of the examples : teacher, learner itself, and external environment. One can also classify learning from examples by the type of examples available to the learner : only positive

examples available, positive and negative examples available. According to the way examples have been presented to the learner, learning could further been subdivided into : one-trial or incremental. In the formal case, all examples are presented at once whereas in the later case, the system must form one or more hypotheses of the concept (or range of concepts) consistent with the available data, and subsequently refine the hypotheses after considering additional examples. Two methods of analogical problem solving that have been studied in AI are transformational and derivational analogy.

Carbonell describes one method for transforming old solutions into new solutions. Whole solutions are viewed as states in a problem space called T-space. T-operators prescribe the methods of transforming solutions (states) into other solutions. Reasoning by analogy become search in T-space; starting with an old solution, we use mean-ends analysis or some other method to find a solution to the current problem. Notice that transformational analogy does not look at 'how' the old problem was solved: it only looks at the final solution. The detailed history of a problem-solving episode is called derivational analogy. Carbonell claims that derivation analogy is a necessary component in the transfer of skills in complex domains. One way to model this behaviour is to have a problem-solver "replay" the previous derivation and modify it when necessary. If the original reasons and assumption for a step's existence still hold in the new problem, the step is copies over. If some assumption is no longer valid, another assumption must be found. If one cannot be found, then we can try to find justification for some alternative stored in the derivation of the original problem. Or perhaps we can try some step marked as leading to search failure in the original derivation, if the reasons for failure conditions are not valid in the current derivation.

Winston's ARCH describes an early structural concept learning program. This program operated in a simple block world domain. Its goal was to construct representations of the definitions of concepts in the domain from provided examples. The examples were given in a near miss fashion. A 'near miss' is an object that is not an instance of the concept in question but that is very similar to such instance.

Michell describes another approach to concept learning called version space. Version spaces work by maintaining a set of possible descriptions and evolving that set as new examples and near misses are presented.

Quinlan's ID3 is a third approach to concept learning by the induction of decision trees. ID3 uses a tree representation for concepts. To classify a particular input, we start at the top of the tree and answer questions until we reach a leaf, where the classification is stored.

5. **Learning from observation and discovery (also called unsupervised learning)** : This is a very general form of inductive learning that includes discovery systems, theory-formation tasks, the creation of classification criteria to form taxonomic hierarchies, and similar tasks without benefit of an external teacher. One may subclassify learning from observation according to the degree of interaction with an external environment : Passive observation - where the learner classifies and taxonomies observations of multiple aspects of the environment, active experimentation - where the learner perturbs the environment to observe the results of its perturbation. Experiment may be random, dynamically focused according to the general criteria of interestingness, or strongly guided by theoretical constraints.

Lenat's AM exploited a variety of general purpose AI techniques : Firstly, it used a frame system to represent mathematical concept. One of the major activities of AM is to create new concepts and fill in their slots. Second, AM also uses heuristic search, guide by a set of 250 heuristic rules representing hints about activities that are likely to lead to "interesting" discoveries. Third, generate-and-test is used to form hypotheses on the basis of a small number of examples and then to test the hypotheses on a larger set to see if they still appear to hold. Finally, an agenda controls the entire discovery process. When the heuristics suggest a task, it is placed on a central agenda, along with the reason that it was suggest and the strength with which it was suggested. AM operates in cycles, each time choosing the most promising task from the agenda and performing it.

Langley's *et al.* present a model of data-driven scientific discovery that has been implemented as a program called BACON, named after Sir Francis Bacon, an early philosopher of science. BACON has been used to discover a wide variety of scientific laws, it begins with a set of variables for a problem, and go on to try out all the possibility of their mathematics or physical relationship, i.e., an equal for indicating variables' relationship, go on adding more variables until there is no more then it stop.

Cheeseman's *et al.* AUTOCLASS is one program that accepts a number of training cases and hypothesizes a set of classes. For any given case, the program provides a set of probabilities that predict into which class(es) the case is likely to fall. This types of discovery is called clustering. Clustering is very similar to induction except no class labeling are provided, the program must discover for itself the natural classes that exist for the objects, in addition to a method for classifying instances.

6. **Neural Network learning** : The most general neural network models assume a complete interconnection between all neurons and resolve the cases of non-connected neurons (i, j) by setting the connection strength $T_{i,j} = 0$. There are, however, a number

of system design parameters which must be specified for any neural network model. These include :

(1) The structure of the system, i.e., the number of "layers".

(2) The synchrony of the system.

(3) The symmetry of the interconnections.

(4) The feedback structure employed.

(5) The transfer of activation function relating input to output.

(6) The formulation of the learning strategy.

The formulation of learning strategies for neural networks continues to be one of the most active and productive areas of research in the field. Several strategies were proposed and one of the first strategies was the perceptron learning rule for adjusting the weight, $W_{i,\ j}$, between input unit j and output unit i when presented with the "true value", t_i , for unit i. This learning rule may be stated in terms of the learning rate, η , and activation a_i and a_j as : $\Delta W_{ij} = \eta(t_i - a_i)a_i$. This rule is equivalent to the set of rules :

(1) Change weights only on those connections to neurons with activation $a_j = 1$.

(2) If the present value, a_j , of neuron j is the true value, t_j , make no change in the weight connecting it to neuron j .

(3) If unit i has an activation $a_i = 0$ when the true value $t_i = 1$, then increase the weights on all active connections by amount η .

(4) If unit i has an activation $a_i = 1$ when the true value $t_i = 0$, then decrease the weights on all active connections by amount η .

These simple learning strategies produce excellent results in simple systems. But as system complexity grows, the effectiveness of many learning strategies decreases. Stephen Grossberg has written extensively on the theory of learning in neural networks, i.e., Adaptive Neural Network model, ANN, and his theory of cooperative/competitive learning appears to offer great promise. Research in learning in neural networks offers many of the same challenges as research in learning in symbolic learning.

The above classification of learning strategies helps one to compare various learning systems in terms of their underlying mechanisms, in terms of the available external source of information, and in terms of the degree to which they rely on pre-organized knowledge. Furthermore, two class of learning strategies ranging out from the above classification are : learning in problem solving and explanation-based learning.

1. Learning in problem solving subclassify into : learning by parameter adjustment, learning with macro-operators, and learning by chunking.

(1) Learning by parameter adjustment : the learning programs rely on an evaluation procedure that combines information from several sources into a single summary statistic. For example, Samuel's checker player.

(2) Learning with macro-operators : Macro-operators were used in the early problem-solving system STRIPS. After each problem-solving episode, the learning component takes the computed plan and stores in a way as a macro-operators.

(3) Learning by chunking : Chunking is a process similar in flavor to macro-operators. The idea of chunking comes from the psychological literature on memory and problem solving. Its computational basis is in production systems. SOAR exploits chunking so that its performance can increase with experience. In fact, the designers of SOAR hypotheses that chunking is a universal learning method, i.e., it can account for all types of learning in intelligent system. PRODIGY is an automatically knowledge acquisition system, employs several learning mechanisms. PRODIGY can examine a trace of its own problem-solving behaviour and try to explain why certain paths failed. The program uses those explanations to formulate control rules that help the problem solver avoid those paths in the future. So, while SOAR learns primarily from examples of successful problem solving, PRODIGY also learns from its failure.

2. Explanation-based learning (EBL) : An EBL system attempts to learn from a single example x by explaining why x is an example of the target concept . The explanation is then generalized, and the system's performance is improved through the availability of this knowledge. Mitchell *et al.* and DeJong and Mooney both describe general frame works for EBL programs and give general learning algorithms. We can think of EBL programs as accepting the following as input : a training example, a goal concept (a high level description of what the program is supposed to learn), an operationality criterion (a description of which concepts are usable), and a domain theory (a set of rules that describe relationships between objects and actions in a domain). From this, EBL computes a generalization of the generalization of the training example that is

sufficient to describe the goal concept, and also satisfies the operationality criterion. Explanation-based generalization (EBG) has two steps : explain and generalize. During the first step, the domain theory is used to prune away all the unimportant aspects of the training example with respect to the goal concept. What is left is an explanation of why the training example is an instance of the goal concept. The explanation is expressed in terms that satisfy the operationality criterion. The next step is to generalize the explanation as far as possible while still describing the goal concept. Next, the explanation is generalized.

Some examples of machine learning and its applications [O'Shea, Self & Thomas 1987] are presented below :

1. ARCH, by Pat Winston at MIT, 1970s. He was interested in how one might learn concepts, and in particular how one might acquire structural descriptions of concepts. This work is very important in artificial intelligence because it is the first clear example of a symbolic representation approach to learning. Winston represented all his learning symbolically; he organizing systems. His approach depends on three things : a training sequence; a set of exemplars of the concept being learned; and the idea of 'near misses'. His system learn to distinguish arch from positive and negative examples.

2. LEX, by Tom Mitchell at Rutgers. LEX learns to do integration and it learns to solve the problem by heuristic search. It learns essentially by reflecting on its own way of doing integration, and so, in some senses, it is an extremely novel system. LEX has four main components : problem solver, critic, problem generator, and generalizer.

The system starts off with an impoverished set of heuristics, with the result that it will eventually solve integrals but not necessarily by the shortest or most economic route. These heuristics about what the system should have done are used by a generzlizer to create a 'version space'. This has examples of the most general things that have been used successfully, and examples of the most specific things that have been used successfully. Critic which compares the system's actual solution, which may have lots of blind alleys (such as applying an inappropriate trigonometric transformation), with an ideal solution which it is able to extract from the actual solution. It uses that comparison as a basis for saying there is an example of something the system should not have done because it took the system away from the ideal solution. Once we have worked with the generalizer we go back to the problem generator to produce a problem that lies in the space of things that we are not sure that we can solve. We do not want problems that we are guaranteed to solve or not to solved : we want integrals that we will solve, but not in the ideal way.

3. ID3, by Ross Quinlan. ID3 which is derived from a piece of psychology work, that of Hunt on concept learning. Quinlan regards learning as constructing classification rules. He says that learning is essentially discrimination : you need to discriminate between one thing and another. He represents classification as a decision tree. The decision tree branches on attributes in the domain, e.g., colours, shapes, etc.

Where Quinlan's work is superior to all the past work in this area is that he has found a way of constructing a minimum tree. For a given set of attributes, his system looks and says, 'I want to classify things in the most economical way : what is the smallest tree I can build?'

4. ANALOGY, Pat Winston at MIT, 1980s. He used an artificial intelligence representation called frames, which are large structures with slots that allow you to look at expectancies. Winston's system attempts to generalize these slots in the context of a system that is trying to understand the plots of plays or stories. His frames structure gives you, as it were, a causal structure of how the plot works, and the system tries to generalize some of the slots and transfer any constraints they have. One of the alarming things about a representation like this is that there are probably many other ways of using the same arrows and the same notations to express the plot of content.

5. POKER, by Don Waterman, 1960s. The system used ordered production rules, and what it learns to do is very appealing : it learns to improve its bet decision at poker. Poker is an inherently more interesting game then chess or checkers because it is a game of imperfect information. Waterman's system had a set of production rules that represented a poker-playing strategy. The system would play poker and it would then, as a good player would after playing a game, analyze how it played it and ask from the point of view of its theory what it should have done otherwise.

6. AM, EURISKO, by Doug Lenat. Lenat was interested in the process of mathematical discovery, and he started out with a clear commitment to the notion that if your system is to learn anything serious then it is going to need to start with background knowledge. Lenat said that the best way to learn knowledge is to have quite a lot of knowledge already. So he created a system (AM) with a lot of heuristic rules and a whole pile of concepts. In his case he started with 115 mathematical concepts represented much like frames with about 24 slots. His system has a control mechanism based on an agenda of tasks, and it prefers to do tasks that are heuristically designated as being the most interesting. The system conjectures that this or that might be an interesting thing to do. It has a notion of 'focus of attention', which is roughly that if something has been done recently keep doing it. And he has, in some sense, some

innovative heuristics. Lenat's four heuristics are : parsimony, regularities, extremes, inverses.

Parsimony is that if you can find a simple way of combining concepts or a way of reducing your descriptions then do it. Parsimony can be thought of as a form of generalization, one of our learning mechanisms. Regularities is simply means that to look for regularities. Extremes is to look at extreme cases, which is a heuristic that is well known in mathematical problem solving and is the type of heuristic described by Polya. Another Polya's heuristics that Lenat uses is that of 'inverses'; that is, if you have a mathematical function or transformation that is interesting to look at its inverse.

AM started off with a description of some fundamental mathematical concepts and then started building up other notions. It built the idea of 'plus', 'partition', 'Cartesian product', 'times', 'exponentiation', and 'divisors', etc. Lenat's system uses a very familiar artificial intelligence method, going by the name known from the General Problem Solver, namely 'generate and test'. It generates a new concept by combining old concepts, or a concept and its inverse, and then tests it.

7. **Alan Bundy**. The architecture which Alan Bundy and his colleagues proposes for learning is basically very simple. He tries to consider everything pretty much as an example of rules learning. There are just two components : a 'critic', which looks at the body of rules and says which is the faulty rules; and a 'modifier' which corrects the faulty rule. There are two standard ways to correct it : one way is to add a new condition, the other way to change the consequence.

The representation that Bundy uses is called a 'description space', and is actually a generalization of Winston's work.

8. **ART1 and ART2**, by Carpenter and Stephen Grossberg. The ART1 and ART2 networks based on adaptive resonance theory (ART) are the result of about 20 years of research. ART family is an unsupervised neural network architecture which can self-organize stable recognition categories in real-time in response to arbitrary sequences of input patters, This network implements a clustering algorithm which is very similar to the simple sequential leader clustering algorithm. It starts by selecting the first input pattern as the exemplar for the first category. The next pattern is then compared to the selected exemplar. If the distance to the selected exemplar is less than a threshold, it will be considered as in the same category. Otherwise, it will be seen as a new category. The differences between ART1 and ART2 is that ART1 is capable of processing arbitrary sequences of binary input patterns whereas ART2 is capable of handling either binary of analogy input patterns.

2.4 Summary

To summarise the learning techniques, we compare some existing learning techniques and their application as followed :

Supervised concept learning, ID3, AQ. This is the most mature machine-learning paradigm. A type of inductive learning, supervised concept learning constructs a concept description in some predefined description language based on a collection or training set of examples. Elements of the training set are marked as positive or negative examples. The resulting concept description can then be used to predict concept membership or future examples. Such algorithms differ in several ways. The first is the language for expressing the target concept. A second important difference is the inductive bias applied in constructing a concept description. A third difference is whether the algorithm operates incrementally or in batch mode. Supervised concept learning has inspired most of the formal work on the theoretical foundations of learning (that is, the "probably approximately correct" learning theory). Most surprisingly, it is also the area that has produced the most application to date. Nevertheless, many important problems remain : dealing effectively with noise in the input examples, worrying about concept drift (when the target concept changes over time), selecting the appropriate inductive bias, handling non-discrete-valued features, and so on.

Conceptual clustering. Conceptual clustering systems differ from supervised clustering systems in that the training examples are not marked as positive or negative by an outside agent or teacher. These systems must recognize the similarities between examples and group them according to some preestablished notion of similarity. Application for these systems are readily drawn from the same problems usually addressed by traditional statistical clustering systems, with one important difference : Unlike statistical clustering algorithms where the number of outcome clusters is predetermined, conceptual clustering algorithms determining the most appropriate number of clusters and then allocate examples to those clusters. In general, concept clustering shares many open problems with supervised concept learning (this is hardly surprising given the close relation between them). Like supervised concept learning, conceptual clustering is a relatively mature paradigms. An officiate success story for such methods is the discovery of a new categorization of stellar spectra that differed from the generally accepted clustering. The new classification was discovered from spectral data by the Autoclass Bayesian clustering system.

Analytic learning. Analytic learning is more recent. A chief example of the paradigm is explanation-based learning algorithms, which are intended to improve the efficiency of a problem-solving system. While they generally do not change the problems that are in principle solvable by the problem solver (that is, the problem solvers deductive closure), they do bias the problem solver's search space. For this reason, EBL has sometimes been described as speed-up learning. Naturally, given unlimited resources, a problem solver would eventually find a solution to any problem within its deductive closure; thus, EBL only makes sense when used to alter the future performance of a resource-limited problem solver. For some EBL systems, this bias takes the form of acquired problem-space macro-operators, which alter the search space by compressing generalizations of previously useful solutions into more efficiently applicable idioms. Essentially, EBL integrates redundant problem-space operators with existing operators to bias the exploration of the search space. Acquired macro-operators can lead to quick solutions, but in other circumstances they can delay the discover of a goal. Other EBL systems represent acquired bias as explicit search-control heuristics for existing problem-space operators. These heuristics typically alter the ordering of alternative choices by promoting heuristically more promising operators so that they are tried first. Some heuristic reject certain operators outright, while others select a particular operator as especially suitable to the current situation (to the detriment of all other operators). As in the macro operator systems, while heuristics should contribute to a quicker solution, the time spent evaluating these heuristics can slow down the search. Several problems within this paradigm remain to be addressed. Speeding up real applications requires controlling performance degradation. No doubt, this problem can be alleviated or avoided altogether through clever indexing techniques coupled with heuristics for managing learned information in some semiprincipled fashion. Perhaps the biggest remaining problem is that, unlike inductive learning systems, EBL systems are domain-knowledge intensive. Thus most EBL systems require complete and correct problem-space descriptions (or domain theories). Recently, the analytic-learning community has begun to address the problem of revising inaccurate or incomplete domain theories on the basis of classified examples. This involves repairing inaccuracies in the domain knowledge that are exposed when examples are handled incorrectly by the original domain theory. Thus domain theory revision is a hybrid problem that shares elements with incremental, supervised, inductive learning problems. Starting from an initial theory (a concept description) that might contain some errors, we patch the theory to account for training examples that were misclassified by the current theory. Given the relative youth of the paradigm, it is not surprising that most analytic learning systems are either proof-of-concept systems or research vehicles to study the performance characteristics of different learning algorithms. Direct applications of this technology to real problems are just beginning to emerge.

Genetic algorithms. These adaptive search systems are inspired by the Darwinian notion of natural selection. First introduced by Holland, these algorithms are ideally suited to solving combinatorial optimization problems, since they efficiently search solution spaces for quasi-optimal solutions. Genetic algorithms can be incorporated in a performance system in a variety of ways. The development of genetic algorithms has followed a path largely independent of the mainstream machine-learning community, spawning a specialized conference, the International Conference on Genetic Algorithms. Unlike some of the other learning paradigms, work in this area has been largely application driven.

Connectionist learning. The ground-breaking work on Perceptrons in the late 1950s represents some of the earliest work on learning systems. After a hiatus of some 25 years, neurally inspired, fine-grained, massively parallel systems are once more attracting attention. Learning is an integral part of any neurally inspired system; indeed, the development of the backpropagation learning algorithm has largely spurred the recent activity in this area. Unlike the early perceptron work, this algorithm supports the training of networks with internal layers of units separating input and output units. Such networks avoid many of the pitfalls of earlier systems. As with genetic algorithms, much of the connectionist work is performed within a specialized community. Nevertheless, the basic problem is exactly the same as that addressed by supervised concept learning . Thus some researchers have evaluated the strengths and weaknesses of connectionist learning schemes and compared them with supervised concept-learning systems.

CHAPTER 3

RELIBALITY ANALYSIS SYSTEMS

3.1 Introduction

Machine monitoring and diagnostics has been considered to be an integral part of the manufacturing process in recent years. It has played an important role in increasing productivity and reducing costs.

In this chapter, we consider three ways of constructing an intelligent system for industrial fault detection and reliability analysis. For neural network and production approaches we indicate how a system might be constructed using examples from the literature. For the model based techniques we suggest an approach adapted from models of electronic components. This approach has not been applied to large scale industrial system and our study is a step in that direction.

3.2 Diagnosis and Risk Analysis in RAs

Diagnosis systems infer malfunctions or faults from observable information. Most diagnosis systems have knowledge of possible fault conditions with means to infer whether the fault exists from information on the system observable behaviour. For example, locating malfunctions in a production line (i.e., real-time fault location, RTFL). Conventionally, production expert system has been applied heavily in this area. Recently, neural network and model-based system also have been applied in this area.

When we diagnose a process problem, we try to determine what fault or disturbance caused the observed deviation. In other words, we try to explain the difference between observed and expected behaviors. When observation and expectations differ in a plant, incorrect assumptions have been made about the plants physical structure or parameter values. [Vinson, Grantham & Ungar 1992] Similarly, in system design stage, if we simulate the plant operation and apply the same technique then this procedure become risk analysis. Risk analysis is a technique apply is system design variation, it allows design engineers simulate of failure caused by inappropriate design (failure mode effect analysis, FMEA).

There are many reasons for causing process malfunction : equipment degradation or failure, external disturbances, operator error, and inappropriate process-control settings [McDowell, Kramer & Davis 1991]. Since operating a process safely and economically requires considering of these situations, risk analysis and real-time diagnosis became a key segment of successfully process operations. Diagnosis for process engineering is complex and consists of many steps : monitoring, detecting faults, diagnosing malfunctions, and planning corrective actions.

1. Monitoring activities track process variables intelligently and provide knowledge-based explanations of normal process behavior, but their task is complicated by the overwhelming number of possible alarms and by the problem escalation. Fixed-threshold alarms might not be sensitive to process trends that can lead to an alarm state.

2. Fault detection, which is closely related to monitoring, involves differentiating between normal and abnormal conditions. Managing this kind of problem solving requires reasoning about physical relationships in a way that explains the current process state and predicts trajectories that the process is likely to follow. Current efforts use causal models to explain alarm states and apply connectionist architecture's to classify current process states from on-line data.

3. Malfunction diagnosis isolates and identifies process malfunctions. This can be especially difficult in process systems, given dynamic behaviors caused by control systems and by mass and energy feedback. To isolate malfunctions under dynamic conditions, we must represent and reason about feedback and its effects on symptomatic information that is used to evaluate hypotheses. Diagnosis is also made difficult by large malfunction hypothesis spaces. Chemical process plants often have hundreds or thousands of pieces of equipment and process settings. Navigating these hypothesis spaces requires that we structure diagnostic knowledge in forms that are efficient for problem solving. The problem-solving architecture should also let us easily integrate traditional numerical techniques that can contribute to diagnostic problem solving.

4. Corrective-action planning takes a diagnostic conclusion and provides a plan of action to deal with the problem safely and economically. Choosing the proper corrective action can involve mapping malfunction states to specific plans. If a plan is not preenumerated, it must be constructed at runtime. Once a plan has been executed, its effectiveness must be monitored. Additional corrective actions might be required if the plan fails to return the process to a safe and economical state.

Our RAs is focus in performing item 3. However, diagnosis process is made difficult by its the large amount of knowledge and experience it requires [Rauch-Hindin 1988] :

1. It requires knowledge of the equipment and how it operates normally.

2. It requires gathering some information about the failed equipment and its fault symptoms.

3. It requires knowledge of what type of equipment information it is necessary to gather that is relevant to the fault.

4. It requires the ability to form a hypothesis and perform some tests to get back more information that either confirms or denies the hypothesis. Knowledge systems offer a way to preserve and protect a troubleshooter's expertise and to make that expert troubleshooter a consultant to many people.

Not surprisingly, RAs have become an important area of application for diagnosis systems. ESCORT (an expert system for complex operations in real time) and REALM (a reactor emergency action level monitor) [Dvorak & Kuipers 1991] are two of many expert systems developed for process industries. These systems aim to reduce the cognitive load on operators, usually by helping to diagnose the cause of alarms and possibly by suggesting corrective actions. Most of these conventional rule-based expert systems get their knowledge of symptoms, faults and corrective actions through the usual process of codifying human expertise in rules or decision trees. The problem as Denning observes (ibid.) :

> *The trial-and-error process by which knowledge is elicited, programmed, and tested is likely to produce inconsistent and incomplete databases; hence, an expert system may exhibit important gaps in knowledge at unexpected times.*

An alternate approach is to use a model of the process to predict system's behaviour or to check consistency among observed variables. Actually, we can regard models as a specific type of knowledge representation, i.e., a way of representing deep knowledge. When observations disagree with the model's predictions, some diagnostic technique is initiated to identify the fault candidates. Alternatively, unsupervised learning neural network system provided a very promising future for constructing RAs as well.

The potential applicability of RAs suits many industries applciations, such as power and process industry. The advantages of RAs to industrial sectors mentioned above are :

1. Acceleration in the procedures of risk assessment and reliability analysis allowing much faster feedback during the system design phase. Design changes or alternatives usually require only minor modifications of the component and of reliability and safety measures. Obviously, this will benefit the final system reliability/safety.

2. The highly structured system and component modeling procedures reduce the possibility of overlooking vital events. This will improve the quality of reliability analyses.

3. Providing more efficient and faster (corrective) operator actions through RTFL. When deviation in process parameters is reported, the RTFL software is able to generate a sorted list of faults and fault combinations which are most probably responsible for system malfunction. This enables operators to react more quickly and thus to shorten down times and to minimize damage to equipment, people or environment. It also allows operators to gain more insight into the state of the system, thus leading to more adequate control actions.

4. The expert-interface and structured procedures incorporated in this software allow reliability/risk analyses and RTFL to be performed by engineers other than reliability experts.

Risk analysis : system designers either have to choose the best option on paper, or build and test a small-scale version of the most promising design, or depend upon the expertise of system vendors. [Jain & Mosier 1992]

Simulation methods have a long history as an aid to design efforts. The advantage of simulation modeling is the operational evaluation of candidate designs without the high risk of 'hard' experimentation. (ibid.)

However, simulation is essentially a 'trial-and-error' methodology, guided by the expertise of the designer who most often relies on a variety of guidelines and rules of thumb. Optimality is not guaranteed, and in fact, not likely. Also depending on the complexity of the problem being investigated, simulation analysis can be incredibly time-consuming and expensive, requiring a high level of statistical and computer expertise on the part of the analyst, and requiring a surprising amount of time to investigate the possibly numerous configuration options. (ibid.)

Since the process of developing candidate systems and system options is experience based, the use of expert systems technologies can significantly reduce the time and expense associated with simulation modeling. (ibid.)

An expert system for designing industrial system which uses the output from a simulation model has been developed by Mellichamp and Wahab. 'Frames' are used to represent facts about a specific FMEA design to be evaluated, perform the actual design assessment, to represent heuristic used in the analysis, to supervise input of data from the simulation analysis, and to supervise the design process. In their system, four types of rules are stored in frames in the knowledge base (ibid.) :

1. Top rules : dealing with the design objectives.

2. Analysis rules : making utilization and queue length assessments based on specific targets.

3. Local rules : making operational diagnoses based on utilization and queue length assessments and type of loading.

4. Global rules : making global diagnoses and recommendations based on operational diagnoses and relational facts.

The inference engine is used to direct the analyst to the various alternatives in the analysis - diagnosis - recommendation process. (ibid.)

3.3 Neural Network Approaches

In recent years artificial neural networks (ANNs) or connectionist systems have come into the limelight and to some extent divided AI community. In this section, we aim to introduce ANNs and to survey its applciations in the RAs. Following is an example of applying ANNs in manufacturing systems.

Hsin-Hao Huang and Hsu-Pin Wang [Huang & Wang 1993] proposed a neural network approach for machine faults diagnostics. They employed a back propagation neural network (ART2) architecture to analyze the FFT (fast Fourier transform) spectrum in order to determine the fault classification.

There are three widely used techniques for machine monitoring and diagnostics : vibration analysis, oil analysis, and plant performance analysis. Generally, two classes of approach have been used for estimating the vibration spectrum : Fast Fourier Transform (FFT) - based methods and parametric methods. FFT-based method, such as spectrum analysis, is the most widely used technique because of its high computational speed. Furthermore, AR (autoregressive) or ARMA (autoregressive and moving average) method is used in an attempt to alleviate the inherent limitations of the FFT approach.

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + E_t$$

Where X_t = time series,

ϕ_i = the AR parameters,

p = the order of AR model,

E_t = residuals with NID $(0, \sigma_E^2)$

ART has been developed by Carpenter and Grossberg. It is an unsupervised neural network architecture which can self-organize stable recognition categories in real-time in response to arbitrary sequences of input patterns. This network implements a clustering algorithm. It starts by selecting the first input pattern as the exemplar for the first category. The next pattern is then compared to the selected exemplar. If the distance to the selected exemplar is less than a threshold, it will be considered as in the same category.

Since the parameters of the vibration signal are analog, an ART2 network is used in Huang's and Wang's research. Fig. 3.3.1 shows the framework of the ART2 approach for automatic identification of machine faults. This framework consists of three modules : a parametric model, a normalisation process and an ART2 neural network.

Once a proper parametric model is determined, the model can be used to fit machine vibration signal. After fitting the model to the vibration signal, a set of parameters can be obtained. At this point, the parameters cannot be fed into the ART2 network without preprocessing because they contain meaningful negative values which the ART2 network is not able to recognize. Therefore, a normalisation process has to be applied in order to make sure that the ART2 network can perform correctly by getting proper inputs.

Normalisation is a two step process. First, it needs to divide each parameter into two parts : positive and negative. If a parameter has a positive value, the negative part will be assigned to zero, and vice versa. Secondly, it needs to scale the parameters by dividing each parameter by the maximum parameter value. Except for residuals variance, it will only contain the positive part because its value is always positive. However, it still has to be divided by the maximum residuals variance. As such, an ARMA or AR model with n parameters will required $2n+1$ input nodes in the ART2.

During training of the network, ART2 is presented with a set of input patterns, i.e., the normalized parameters. As a result, the network self-organizes fault classifications, according to the procedure describe previously, until it runs out of the input patterns. At last, the final top-down and bottom-up weightings will be saved for later diagnostic use. During diagnosis of a fault, each input pattern represents a particular fault classification.

According to the experimental result, ART2 network has demonstrated its accuracy and robustness in identifying the fault classification. Therefore, the proposed approach can be used as a decision-support tool for machine monitoring and diagnostics.

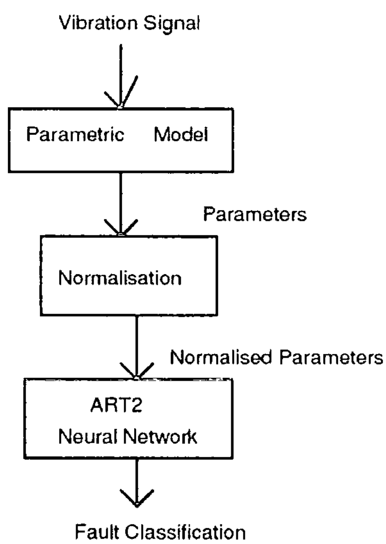

Fig. 3.3.1 Automatic machine fault identification framework

3.4 Symbolic Approaches

Symbolic approaches, or "knowledge-based" expert system, are the same thing. Fig. 3.4.1 shows the general architecture of a knowledge-based expert system, comprising of three major components :

1. The user interface component provides the communication channel between end user and the expert system.

2. Knowledge-base : this is the place where expertise has been capture in the system. Many forms of knowledge can be stored here, e.g., procedural knowledge, declarative knowledge, meta-knowledge, heuristic knowledge and structural knowledge. [Durkin 1994]

(1) Procedural knowledge : describes how a problem is solved. This type of knowledge provides direction on how to do something. Rules, strategies, agendas and procedures, are the typical type of procedural knowledge used in expert systems.

(2) Declarative knowledge : describes what is known about a problem. This includes simple statements that are asserted to be either true or false. This also includes a list of statements that more fully describes some object or concept.

(3) Meta-knowledge : describes knowledge about knowledge. This types of knowledge is used to pick other knowledge that is best suited for solving a problem. Experts use this type of knowledge to enhance the efficiency of problem solving by directing their reasoning into the most promising areas.

(4) Heuristic knowledge : describes a rule-of thumb that guides the reasoning process. Heuristic knowledge is often called shallow knowledge. It is empirical and represents the knowledge compiled by an expert through the experience of solving past problems. Experts will often take fundamental knowledge about the problem (called deep knowledge), such as fundamental lows, functional relationships, etc. and compile it into simple heuristics to aid their problem solving.

(5) Structural knowledge : describes knowledge structures. This type of knowledge describes an expert's overall mental model of the problem. The expert's mental model of concepts, subconcepts, and objects is typical of this type of knowledge.

Furthermore, several techniques for representing the above knowledge are : object-attribute-value triplets, rules, semantic networks, frames, and logic.

3. Inference engine : this mechanism simulates human being's reasoning function. There are several reasoning techniques possessed by humans : deductive reasoning, inductive reasoning, abductive reasoning, analogical reasoning, common-sense reasoning, and non-monotonic reasoning. In order to make computer simulate these reasoning techniques, usually we adapt several searching techniques, e.g., forward chaining, backward chaining, top-down, bottom-up, deep-first, bread-first, best-first, heuristic search, etc.

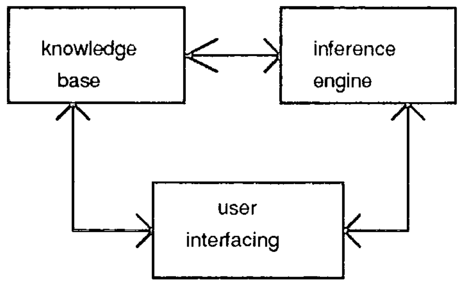

Fig. 3.4.1 Knowledge-based system

The earliest application of knowledge-based system in diagnosis process is system called CATS-1 (a rule-based Computer-aided Trouble Shooting system) [Rauch-Hindin 1988] designed by General Electrics (GE). The system originally is designed as a general tool for troubleshooting electronic equipment. Since then, rule-base systems have had always acted as a major role in diagnosis system. However, a more recent trend in the field relies on a model-based reasoning approach, which models the system's normal behavior, and detects and diagnoses faults from deviations in expectations.

3.4.1 Production System

The best known diagnosis system probably is MYCIN [Waterman 1986] which is designed for assisting physicians in the selection of appropriate antimicrobial therapy for hospital patients with bacteremia, meningitis, and cystitis infections. However, considering our discussion is based on industrial application, we are going to introduce CATS-1 as our example here.

In late 1981, GE started the project of constructing a rules-based troubleshooting system. Through many testing and trail stages, finally , by September 1984, the system was accepted by the general public. During a troubleshooting session, CAT-1 starts by collecting background information and problem symptoms. Upon startup, the system asks a number of questions about the locomotive model number, model year, and reported symptoms. CATS-1 tables provide additional information, such as the locomotive's standard features, history of failures, and that model's propensity for failures.

The loading of the reported symptoms and background information into the knowledge system triggers its diagnostic procedures, which culminate not only in a diagnosis but also in recommended corrective actions. The diagnostic procedure involve both backward chaining and forward chaining techniques for reasoning. To perform its deductions and diagnosis, CATS-1 uses information input by users or sensors, in addition to its own knowledge in the form of IF-RULES; IFF (if-and-only-if) Rules; WHEN Rules, which activate new procedures associated with newly inferred facts; and Meta Rules, which control, recognize, and reorder the reasoning processes.

The diagnostic process begins with backward chaining. Based on the initial symptoms input by the user and the possible causes that the expert has suggested, the backward chaining proposes a likely hypothesis for the particular problem area in question. Then it attempts to find rules which substantiate that hypothesis.

A typical CATS-1 question to the user, to determine the symptoms and hypothesis that start the backward-chaining process, might be :"Is the governor steady?" If the answer is no, CATS-1 knows five possible causes. The order in which these causes are proposed as hypotheses for evaluation are in order in which these causes are proposed as hypotheses for evaluation are in order of increasing cost of the test to prove or disprove the hypothesis. In the case of factors that could make an unsteady governor, the impulse thing to check first is whether there is enough oil; second is a test to determine whether or not the oil is clean. If neither of these solves the problem, CATS-1 tries out the next, more complex, least expensive hypothesis.

Determining a specific faulty component may involve interaction with several rules and with the forward chainer. For example, while the backward chainer is active, CATS-1 rules 760 might hypothesize that a locomotive has a fuel system fault. Loosely translated, rule 760 says :"IF the engine is set at idle AND the fuel pressure is below normal AND the readings were taken from the locomotive gauge AND we are sure that the gauge is accurate THEN we can conclude that the duel system is faulty (1.00) ." The number 1.00 associated with the action part of the rule is a certainty factor. Defined by the experts, it can range between -1 and 1 and indicates the strength of the implication or conclusion of the rule.

In contrast to the backward chainer, which is goal directed (tries to prove a goal), the forward chainer simply reacts to changes. Whenever the forward chainer spots new facts that have been added to the knowledge ones, it scans the CATS-1 rules to see if its known information is now sufficient to execute the THEN parts of any rules that contain the new facts as their conditions. The execution of these rules allows CATS-1 to infer still more facts, which the forward chainer uses to try to execute other rules. Some of the information inferred, and subsequent rules executed, causes the forward chainer to submit a new hypothesis to the backward chainer.

WHEN rules may be used to submit new hypotheses. Used only by the forward chainer, they take the form, "WHEN this is true THEN do that" or "WHEN this is true THEN that is also true."

When the forward chainer cannot find any more rules to react to and execute, it returns control to the backward chainer. The backward chainer continues its deductive process until it has proved a hypothesis or has exhaustively evaluated and either proved or disproved all its hypotheses.

CATS-1 does not assume that only one component is at fault, because the GE engineers do not consider that realistic. Instead, every time CATS-1 finds a fault at any level of the locomotive system, it expects it to be fixed immediately. It then reevaluates the locomotive symptoms to see if the fix resets the system to normality or if there are more faulty components.

Upon request from the user, a CATS-1 Help system provides additional information, such as location and identification of repair procedures. Since typical users are skilled blue collar workers, the Help system is organized so as not to intimidate the user in terms of spelling or excessive choices.

The GE diagnostic system will play a role in repair and maintenance training as well as in the troubleshooting process. It will also allow its on-the-job technicians to incorporate in the system their own knowledge and experience. In its original form, CATS-1 confines its troubleshooters to the ideas and disciplines of the expert whose knowledge was professionally embedded in the system. This is considered by GE to be a "quasi-optimal" rather than an "optimal" situation.

3.4.2 Model-base System

Rule-based diagnosis systems have proven to be effective tools in increasing the productivity of complex industrial manufacturing processes. However, the development of such a diagnostic system is strongly hampered by the generally acknowledged bottleneck of knowledge acquisition. Most researchers in the field expect a solution to this problem to arise from the development of application specific expert system shells, which already contain the conceptual framework of the domain to be covered by the system. Therefore, knowledge engineers only need to carry out a shallow level of knowledge acquisition, and can take reasonable short cuts. Knowledge acquisition thus becomes a relatively easy affair consisting of a few interviews with the expert, focusing on how to recognise problems and what the associated solutions are.

This results in the construction of a set of heuristic which reflects the behaviour of the expert, and is usually implemented directly in rules. In fact, most of the effort in implementation phase. This fast track method of building expert systems works well for small scale systems, and there is therefore no reason to stop using it. However, it is not applicable to the process of building the large and complicate system like industrial applciations.

In order to meet these extra requirements expert systems need to be able to draw upon much larger and richer pools of knowledge than is bound in shallow system. The new generation of expert systems are called model-based because they store the extra knowledge in terms of models of different aspects of the domain [Steels & Velde 1989]. There are three domains which can be modeled :

1. Model of information concerning the task structure, which reflects the scope of the behaviour of the expert system.

2. Model of the phenomena which the objects in the domain show, concerning such things as their properties, structural and functional relationships.

3. Model of strategies which the expert uses to solve problems in the domain.

The Fig. 3.4.2-1 shows a suggested structure for a Model-based Reliability Analysis system (MORA) which is developed as a tool for fast and reliable of reliability and risk analysis of, and real-time fault location in, complex systems as they are encountered in industry. RAs allows both reliability engineer and end-user to create (and edit) compact component and system models in a convenient and faster manner which is able to perform Fault Mode Effect Analysis (FMEA) and Real-Time Failure Location (RTFL) virtually automatically.

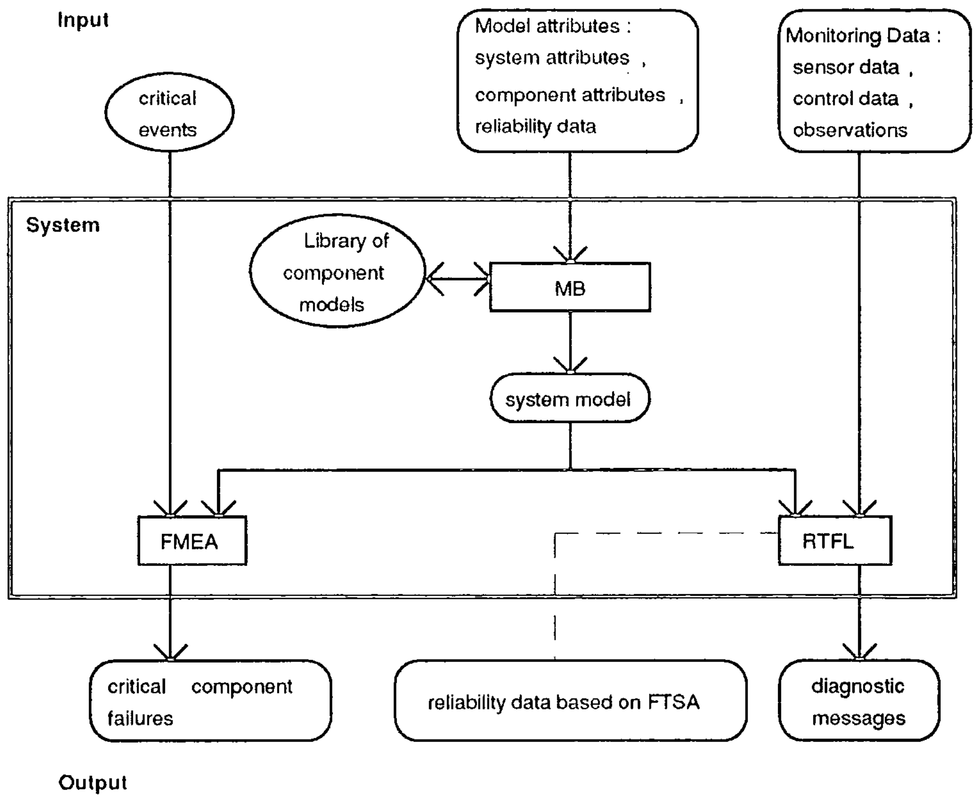

Fig. 3.4.2-1 Architecture of Model-based RAs system

There are two different approaches for constructing model-based expert system [Dvorak & Kuipers 1991] :

1. Engineering approach (quantitative models) : fault detection and isolation techniques generally rely on a precise mathematical model of the process and on preenumerated symptom-fault patterns know as fault signatures. Examples, parameter estimation method, use plant data to determine the values of process parameter. Faults are detected when one or more parameters of outside a given range or change significantly from their normal values. These methods detect deviations but cannot diagnose the root causes of faults. Another example, model selection methods, it fits several models (including fault models) to incoming data; the model that best fits is taken to be the diagnosis. However, quantitative methods don't always work, since failures often cause unexpected behaviors that the models do not cover or for which quantitative are unavailable.

2. Computer Science / Artificial Intelligence approach (qualitative models) : model-based diagnosis relies on models of structure and behaviour. For example, given symptoms of misbehaviour (as detected by the behaviour model), fault candidates are identified using the structural model by following a dependency chain back from a violated prediction to each component and parameter that contributed to that prediction.

Examples, dynamic qualitative models, extended signed directed graphs, semiquantitative model, fuzzy qualitative models, and causal models and confluence equations.

MORA has evolved within the AI community and we adapt causal model in constructing our model-base system. A causal model consists of a set of properties of components which are causally related in the sense that the value of one property is determined by the values of one or more other properties. Some of these properties are observable, most of them are not, or only with difficulty. Causal relations can easily be represented in a network (use semantic network, or frames) whether the nodes represent the properties of the components. [Kodratoff & Hutchinson 1989]

The key cognitive skill for process operators is the formation of a mental model that not only accounts for current observations but also lets the operators predict near-term behaviour as well as the effect of possible control actions. This observation underlies our framework for process monitoring, RTFL (details is given in next section). The basic idea is mimic the physical system with a predictive model, and when the system changes behaviour due to a fault or repair, change the model accordingly so that it continues to give accurate predictions of expected behaviour. RTFL takes the role of tracking and it advances the model's state in step with observations from the physical system. When observations disagree with predictions, RTFL use model-based diagnosis to determine the possible faults. After identifying a fault, the diagnosis task injects it into the current model so that the predictions will continue to agree with actual observations. To be precise, RTFL maintains a set of candidate models since a given behavior might be caused by one of several faults. Each candidate model represents a possible condition of the system, including its state and faults. The key benefit of this approach is that we can use the model as our window into the physical system. Specifically, the model can :

1. Detect early deviations from expected behaviour more quickly than with fixed threshold alarms, uses known analytical relationships among sets of signals to check for mutual consistency.

2. Predict the values of unobserved variables (signal reconstruction) to permit alarms or other inferences on unseen variables, and to help the operator understand process conditions.

3. Predict near-term undesirable or hazardous conditions, thus providing early warnings.

4. Predict the effect of proposed control actions to test if they will have the desired effect - a valuable capability in complex system.

The end purpose of monitoring and diagnosis is advice to the operator about what's happening and what to do about it. The advising task applies expert knowledge of safety conditions, recommended operating procedures, and performance objectives to produce advice in the form of alarms, warnings, and recommended actions.

3.4.2.1 The Modeling Procedure

The scope of modeling carried out by a reliability engineer will include :

1. The bi-directional relationship between the different physical variables in the component (fault propagation).

2. The effect of component failures on the component variables.

3. The bi-directional interaction between the component and its environment.

4. The impact or (extreme) deviations in component or environmental variables on the component state (secondary failures).

5. The library contains specific description of individual component.

This modeling procedure is very much like the 'knowledge acquisition' in a knowledge-based system. The component model, the system model was abstract into systems library for later usage in RTFL and FMEA. However, those models are not fixed, they can be accessed and updated by the end-user (e.g., system designer and real time operators) through MB component.

The model developed here should show all fault initiation within the system and fault propagation through the system. Each component will be modeled as a black box with in- and outputs reflecting the physical connections to other system components and environment. Each connection might be associated with multiple physical variables, e.g., one connection through which a fluid is passed to another component might be associated not only with flow and pressure but with flow temperature as well. System models describe the interconnection of the components. When defined these models allow virtually automatic FMEA and RTFL.

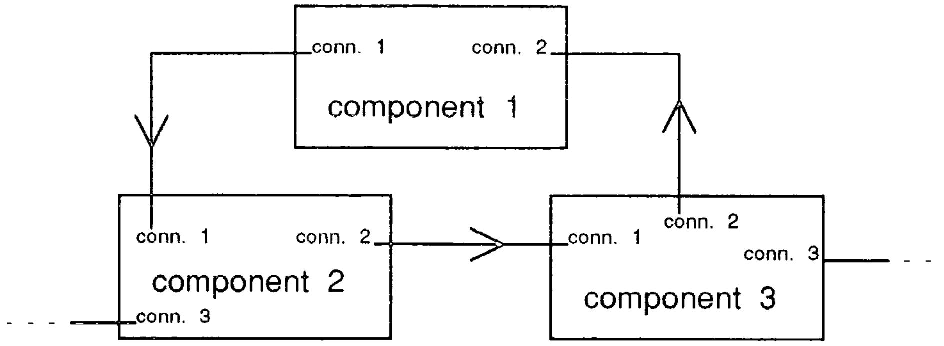

Fig. 3.4.2.1-1 Examples of Component models

Basically these signals will be classified into two groups :

1. Twin variables which exhibit the special relation between the transport variable (flow, current, etc.) and the potential variable (pressure, voltage, etc.) of continuous mass and energy flows. These relations are special in that they depend on the direction of fault propagation with respect to the direction of transport, i.e., the relations are different for upstream and downstream fault propagation. For mass transport a further subdivision will be made according to the different states of aggregation of the substances.

2. Single variables which describe single-valued signals such as : temperature, information signals, concentration (e.g., ppm) and position and orientation of discrete products.

Although the number of deviation levels to be distinguished will differ from variable to variable, the frame of possible deviations in the MORA system will at least comprise the following deviations : very high, high, normal (desired value), low, very low, absent (no signal at all) and reversed (e.g., backflow). In the model definition phase of the project, this frame will be extended as required.

Fault initiation by component failures and fault propagation through the component will be described by relations. The relations must be able to describe :

1. Deviation propagation (static and dynamic) from one variable to another within the same component (fault propagation). This might involve cross-dependencies between different physical properties within one connection as well. An example of this is the dependence of gas pressure on gas temperature in a fixed volume.

2. Variable deviations introduced by component failures (fault initiation).

3. Component state transitions caused by extreme variable deviations, the so called induced failures.

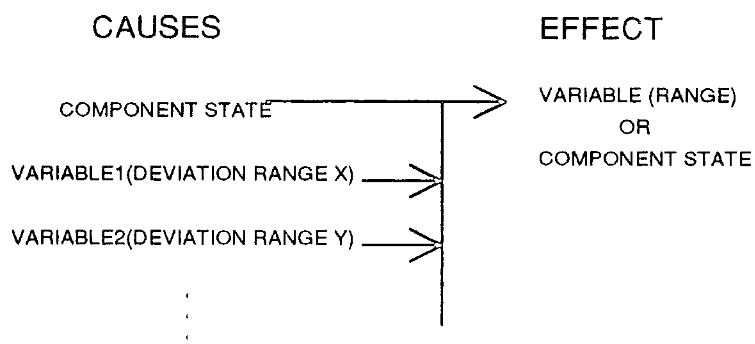

Fig. 3.4.2.1-2 Fault Tree Model

All physical systems have three essential elements in common : a set of component, and environment and a configuration. As a consequence, the system model to be developed should contain : a list of components, a list of system input/output variables describing the interaction with the environment, and a component interconnection scheme describing the system configuration.

These modeling procedures will be implemented into the MORA system (computer program) that enables designers and engineers to model their systems even when they are not reliability experts. Computer programs will be developed for automatic FMEA and RTFL.

3.4.2.2 Design of MB, RTFL and FMEA

The Model-based RAs system (MORA) could be developed comprising the following program modules :

1. MB (Model-builder) - a menu-based computer program that allows end-user fast creation or updating of model-base and libraries contained in MORA (i.e., add, delete, edit, copy, etc.). The questions asked to the user will be straight forward and simply require only a minimal expertise from the user. The user only needs to have enough knowledge about the physical behaviour of the component in order to be able to predict the fault propagation within the component. While only a very shallow system knowledge is required to decide which variables are important for neighbour components and which are not. The user will be asked to define :

(1) the number of input/output connections of the component including undesired connections with other components due to environment coupling.

(2) the number and type of all important physical variables associated with each connection.

(3) the fault propagation relation describing the way in which the physical variables affect each other and, in case of induced component failure, the component state.

(4) the fault initiation relations that describe the effect of component failures upon variables, and relations between variables.

Above all, the MB component should be possible to display the model or the system in more than one type of representation.

2. FMEA (Failure Mode Effect Analysis) - a procedure that automatically develops cause-consequence paths for all component failures or other basic events. Starting at the initial event, all propagation paths of the cause-consequence diagram will be developed simply by connecting the relations from component and system models in the right manner and for appropriate parameter values. Subsequently the cause-consequence diagram will be adapted for compensating actions of control loops that might prevent some faults from further propagation. When the program is fed with the essential deviations in system parameters and their respective criticalities, it will perform a Criticality analysis as well. As an output the user will get information on critical component failures, which have to be prevented by appropriate design measures. Furthermore, the FMEA program will allow the user to manipulate the presentation and ordering of the effect and criticality data according to individual demands.

3. RTFL (Real-Time Fault Location) - a procedure that is able to extract all faults and fault combinations which are most consistent with the set of measured variables even when some sensor circuits provide faulty information. When deviations in one or more system variables are reported, causal trees (multi-level fault trees) in which all consistencies and inconsistencies with these measured variables have been considered can be generated automatically. Subsequently, the RTFL algorithm will modify the causal trees into diagnostic diagrams, which determine the order in which diagnostically relevant data, i.e. results of measurements or human inspection are collected and evaluated. Minimal cut sets can be extracted from theses trees. However, the more inconsistencies that are involved with a cut set, the less likely it is that the cut set is

responsible for system malfunction, thus allowing a substantial reduction in minimal cuts. The RTFL procedure combines these reduced cut sets and sorts the results according to the number of inconsistencies involved. In this way the procedure is able to produce all faults and fault combinations that are most consistent with the set of measured variables, even when some sensor circuits provide fault information.

The procedure of FMEA and RTFL should be able to deal with feed back and feed forward control loops since theses loops might compensate (or overcompensate) for the effects of certain failures. Previous studies have shown this possible. The RTFL software will be able to extract all faults and fault combinations that are most consistent with the set of measured variables, even when some sensor circuits provide faulty information. In this case system operators will be able to gain more insight into the state of the system thus allowing a more adequate control action.

3.5 RAs in the Future

As different approaches of constructing Reliability Analysis systems (RAs) provide different advantages and disadvantages, the following is the summary :

1. Neural network approach : Through the experiment of Huang and Wang, it shows that Artificial Neural Networks (ANNs) have a powerful and general technique for unsupervized learning. This could be one of the best approach for building learning RAs. However, ANNs have several well-known shortcomings; perhaps the most significant of which is that a trained ANN is essentially a "black box". That is, determining exactly why an ANN makes a particular decision is on a shaky ground. This is a significant shortcoming, for without the ability to produce understandable decisions, it is hard to be confident in the reliability of networks that address real-world problems. Also, the fruits of training neural networks are difficult to transfer to other neural networks, and all but impossible to directly transfer to non-neural learning systems. Hence, the network can tell you that it has discovered something "wonderful," but then does not tell you what it has been discovered. To overcome this, the best way is to combine neural network with model based reasoning. [ECAI92 1992]

With the above combination, the advantages are that the unsupervised learning neural networks are capable of extracting regularities form data. Due to the distributed subsymbolic representation, neural networks are typically not able to explain inferences. However, to avoid this, the system can extracting symbolic rules out of the network. The acquired rules can be used like the expert's rules therefore it is possible to diagnose new unknown examples. Another ability is to handle a (large) data set for which a classification or diagnosis is unknown. For such a data set classification rules are proposed to an expert. The integration of a connectionist module realizes "learning from examples." Furthermore the system is able to handle noisy and incomplete data. Therefore, it is the best design to equip a rule based expert system with the ability to learn from experience using a neural network.

2. Knowledge-based approach : Rule-based diagnosis system had a long lasting history. However, its power was limited by knowledge acquisition bottleneck. Although, expert shell provide a short cut for the solution, the knowledge contained in this type of expert system is narrowed which is not suitable to be applied to a large range or complicated area. To extend its application, Steel proposed a new approach of constructing expert system, model-based systems. Model-based expert system have an additional component in the form of a deeper model which gives them an understanding of the complete search space over which the heuristics operate. This makes two form of reasoning (heuristic rules reasoning and deep search space) possible. However, as the problem encounter in rules-based system, the knowledge acquisition bottleneck remains unsolved. Again, expert shell for model based knowledge acquisition opens a short cut to the building of model-based systems, however we believe only by applying Machine Learning can achieve automatic model acquisition, and many other benefits from ML can also be brought to knowledge-based reliability analysis system.

To conclude this section, we suggest that the future architecture of Reliability Analysis System should be the combination of model-based system (software) embedded in a neural network machine (hardware). This is the closest design to human beings. The model-based RAs take the responsible of risk analysis and diagnosis whereas the neural network provides the function of learning complex, nonlinear functions. However, as the model-based RAs constructed by deep knowledge and heuristic rules, there are many other learning techniques which can be applied to improve the performance. Details are discussed in the following chapter.

CHAPTER 4

APPLICATIONS OF ML IN RAs

4.1 Introduction

In the previous chapters, we have already discussed some techniques of machine learning (ML) and three different types of Reliability Analysis System (RAs). In addition, in chapter 3, we reached to a conclusion that current techniques of building the RAs will be based on a hybrid structure of using a combination of Model-based RAs (MORA) and neural network system. Following, we are going to focus our discussion on the application of ML in the MORAs.

4.2 Neural Network Learning

As we deduced from the last chapter, the combination of MORA and neural network structure is currently the best approach for building RAs. The advantages are :

1. The unsupervised learning neural networks are capable of extracting regularities form data - automatic knowledge acquisition.

2. The acquired rules can be used like the expert's rules. Therfore it is possible to diagnose new unknown examples.

3. Another ability is to handle a (large) data set for which a classification or diagnosis is unknown.

Such a data set classification rules are proposed to an expert. The integration of a connectionist module realizes "learning from examples." Furthermore the system is able to handle noisy and incomplete data. It is the best to equip a model-based expert system with the ability to learn from experience using a neural network.

One successful implementation is an existing hybrid system, developed by Bechtel AI Institute and Neurocomputing, used in manufacturing inspection systems to identify defects or to position components. Comment by the manager of the institute, George Polzer (adapted from Expert Systems User, April 1990, p.6) : " The combining of neural networks and expert systems is beneficial because of the complementary strengths of each technique. Neural networks are highly adaptive at learning through examples. Expert systems work by following rules which direct the decision-making process." The integrated software packages are the Nexpert Object expert system package from Neuron Data Inc. and the image recognition software of Nestor, Inc. The resulting application runs on a 286-based PC with the necessary imaging equipment attached.

4.3 Finding the Inconsistency

By applying a machine learning technique, the major improvement will be in maintaining the consistency of input data in any representation forms in frame-structured representations to represent the models, or in rule formalisms to represent the heuristic knowledge, or in constraints to drive deep reasoning, etc. The following is an existing system which applied in maintaining the consistency of database.

Carper [Schlimmer 1993] is a learning system applying in maintaining database consistency, it has been tested on XCON (Digital Equipment Corporation). Carper relies strongly on attribute models that capture how values describe entries, it will let the user specify models and supplement them with models it learns by studying database entries. The system detects database problems by applying attribute models to database entries and generating predictions. If an entry violates a prediction, Carper raises an alarm. Fig. 4.3-1 shows Carper's overall organization.

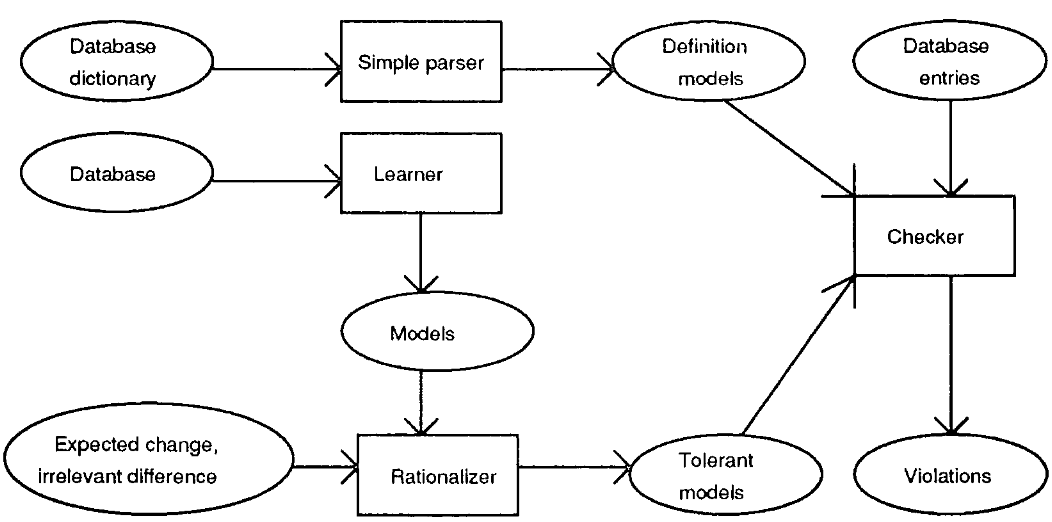

Fig. 4.3-1 Carper's architecture.

Carper uses an inductive method for learning from examples to construct more specific models, where other entries are examples, and the classes to be learned are the values of the attribute being checked. In the absence of information to the contrary, all other attributes can be used in prediction as features of the examples.

The Carper's learning strategies can be applied in the MORA, so MORA can learn to identify inconsistency as well.

4.4 Learning From Past Experience

A principal shortcoming of existing diagnosis systems is that they learn nothing from any given task. Upon from facing the same task a second time, they will incur the same computational expenses as were incurred the first time. In this section, we consider applying explanation-based learning to speed up the FMEA and RTFL procedure, by applying meta-reasoning to consult "past experience", MORA will be able to learn from its experience. [Fattah & O'Rorke 1993]

4.5 Other ML Applications in MORA

Apart from the above discussion, some other performance of the MORA system will be improved by applying some of the ML techniques :

1. Applying heuristic searching to speed up the process of RTFL and FMEA, i.e., searching speed of the cause-consequence tree.

2. The explanation-base learning assists MORA to cope with uncertain information and also to help speeding up diagnosis and risk analysis.

3. The genetic algorithm will act as a major role of learning from observation.

4. Learning by being told keeping the inconsistency away from our model-base.

Discussion concerned with above techniques in the MORA system were stressed in Chapter 2.

4.6 Learning MORA System

By introduing the 'learning element' (see Fig. 4.6-1), the MORA should be able to perform learning function based on our discussion above. The learning element contained different types of learning strategies, similar to PRODIGY (Minton *et al.*, 1989) utilizes all different types of learning techniques.

In theory, the final learning MORA system (La MORA) will be able to learn like human beings by implanting La MORA (software) on neural-based machine (hardware). Apart from that, La MORA should contains many heuristic knowledge - for explanation-based learning, and meta knowledge - rules for learning other rules, and control knowledge - for control the utilizing of different learning strategies. The architecture of Learning MORA (La MORA) system is showed in Fig. 4.6-1. Actually there is not much different from the MORA system, except the insertion of 'learning element'. So, it should not be very difficult to achieve that. However, application of learning will slow down the process of RAs, also, learning element will take up a lot more memory and slow down the process. In future, we should find a way of improving this.

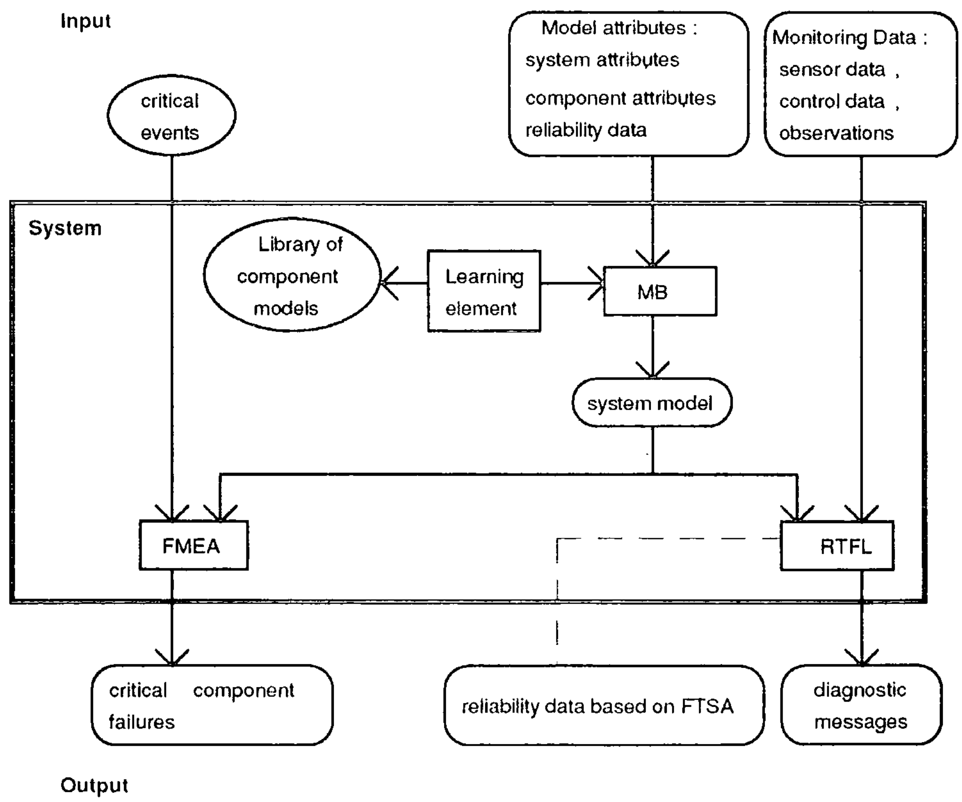

Fig. 4.6-1 La MORA System

CHAPTER 5

CONCLUSIONS AND FUTURE WORKS

5.1 Machine Learning and Its Applications

Our research goal in machine learning (ML) is to explore all types of learning techniques and its applciations, the result of our survey has been discussed in Chapter 2. Two levels of learning model (conscious and neural net level) and six types of learning technique were explored : rote learning, learning from experience (inductive learning), explanation-base learning (deductive learning), learning by being told (learning from instruction), learning from observation (application of genetic algorithm) and neural network learning. Furthermore, by comparing different learning models, we believe that all the learning strategies are actually events of neural network activity, so the future research direction of ML will be towards the neural network learning and the knowledge representation of this kind of learning.

5.2 Reliability Analysis Systems

In chapter 3, we explored three types of reliability analysis systems (RAs) and we conclude that a hybrid structure of combining Model-based RAs (MORA) with neural network based system is currently the best design of RAs. However, the application of Machine Learning will attract the major attention in the future development of MORA.

5.3 ML in RAs - La MORA

Apart from creating an automatic knowledge acquisition system by applying neural network learning, machine learning can be applied to improve the performance of MORA in many other aspects :

1. Applying heuristic searching to speed up the process of RAs, i.e., finding the shortest path within the cause-consequence tree.

2. The explanation-base learning assists MORA to cope with uncertain information and also to help speed up diagnosis and risk analysis.

3. The genetic algorithm will act as a major role of learning from observation.

4. Learning by being told and learning from induction keeps the inconsistency away from our model-base.

Insertion of 'learning element' into MORA can give all the advantages mentioned above and we proposed a way of constructing the learning MORA (La MORA) in chapter 4. However, application of learning function will slow down the operation speed of RAs. This will hamper the on-line useage of RAs. In future, we should try to find out the way of avoiding this. In addtion, the insertion of 'learning element' will take up a lot of space in the memory resulting from the amount of informatin contained in the learning element. To solve this problem, we will have to study the knowelge representation and find out the best way of storing them.

References

[Abu-Hanna, Benjamins & Jansweijer 1991] Ameen Abu-Hanna, Richard_Benjamins, and Wouter Jansweijer 1991 "Device Understanding and Modeling for Diagnosis." In IEEE Expert - Intelligent Systems and Their Applications, April, 1991, pp.26-32.

[Alty & Coombs 1984] J. L. Alty, and M. J. Coombs 1984 "Expert Systems (Concepts and Examples)." NCC Publishing.

[Arbib 1964] Michael A. Arbib 1964 "Brains, Machines, and Mathematics." McGraw-Hill Book Company.

[Arbib 1972] Michael A. Arbib 1972 "The Metaphorical Brain : An Introduction to Cybernetics as Artificial Intelligence and Brain Theory" Wiley-Interscience, a Division of John Wiley & Sons Inc.

[Barr, Cohen & Feigenbaum 1981a] A. Barr, Paul R. Cohen, and Edward A. Feigenbaum 1981 "The Handbook of Artificial Intelligence (Vol. I)." Heuris Tech Press, Stanford, California.

[Barr, Cohen & Feigenbaum 1981b] A. Barr, Paul R. Cohen, and Edward A. Feigenbaum 1981 "The Handbook of Artificial Intelligence (Vol. II)." Heuris Tech Press, Stanford, California).

[Barr, Cohen & Feigenbaum 1981c] A. Barr, Paul R. Cohen, and Edward A. Feigenbaum 1981 "The Handbook of Artificial Intelligence (Vol. III)." Heuris Tech Press, Stanford, California.

[Bau & Brézillon 1992] Dong-Yih Bau, and Patrick J. Brézillon 1992 "Model-Based Diagnosis of Power-Station Control Systems." In IEEE Expert - Intelligent Systems and Their Applications, February, 1992, pp36-45.

[BCS 1987] British Computer Society, Ed. by M. A. Bramer 1987 "(British Computer Society Workshop Series) Research and Development in Expert Systems III - Proceedings of Expert Systems '86, The Sixth Annual Technical Conference of the British Computer Society Specialist Group on Expert Systems - Brighton, 15-18, December, 1986." Cambridge University Press, Cambridge London.

[Biswas, Manganaris & Yu 1993] Gautam Biswas, Stefanos Manganaris, and Xudong Yu 1993 "Extending Component Connection Modeling for Analyzing Complex Physical Systems." In IEEE Expert - Intelligent Systems and Their Applications, February, 1993, Vol. 8 (1), pp.48-57.

[Bonnet 1985] Alain Bonnet 1985 "Artificial Intelligence : Promise and Performance." Prentice-Hall.

[Buckles & Petry 1992] Bill P. Buckles, and Frederick E. Petry 1992 "(IEEE Computer Society Press Technology Series) Genetic Algorithms." IEEE Computer Society Press, Los Alamitos, California.

[Bundy 1980] A Bundy 1980 "Artificial Intelligence (An Introduction Course)." Edinburgh University Press.

[Carbonell 1989] Jaime G. Carbonell 1989 "Introduction: Paradigms for Machine Learning." In Artificial Intelligence , 1989, Vol. 40, pp.1-9.

[Castillo & Alvarez 1991] E. Castillo and E. Alvarez 1991 "Expert Systems : Uncertainty and Learning." Computational Mechanics Publications.

[Charniak & McDermott 1985] Eugene Charniak, and Drew V. McDermott 1985 "Introduction to Artificial Intelligence." Addison-Wesley.

[Charniak, Riesbeck, McDermott & Meehan 1987] Eugene Charniak, Christopher K. Riesbeck, Drew V. McDermott, and James R. Meehan 1987 "Artificial Intelligence programming (2nd ed.)." Lawrence Erlbaum Associates, Hillsdale, New Jersey.

[Chi 1992] Robert T. H. Chi 1992 "Using an Integrated Model Learning System to Construct the Model Base of a Decision Support Systems." In International Journal of Intelligent Systems, 1992, Vol. 7, pp.373-389.

[Chorafas & Legg 1988] Dimitris N. Chorafas, and Stephen J. Legg 1988 "(Computer-Aided Engineering Series) The Engineering Database." Butterworths.

[Christaller, di Primio, Schnepf & Voss 1992] Thomas Christaller, Franco di Primio, Uwe Schnepf, and Angi Voss, Translated by Stephanie Kuhnel 1992 "(Knowledge-Based Systems Book Series) The AI Workbench BABYLON : An Open and Portable Development Environment for Expert systems." Academic Press Limited.

[CIAM86 1987] International Conference on Artificial Intelligence, 2nd Marseilles, 1986 "(CIAM 86) Advanced in Artificial Intelligence : Proceedings of the 2nd International Conference on Artificial Intelligence - December 1-5, 1986, Marseille." Kogan Page.

[Clark 1990] Dominic A. Clark 1990 "Numerical and symbolic approaches to uncertainty management in AI." In Artificial Intelligence Review, 1990, Vol. 4, pp.109-146.

[Clark, Baldwin, Berenji, Cohen, Dubois, Fox, Lemmer, Prade, Spiegelhalter, Smets & Zadeh 1988] Dominic Clark, Jim Baldwin, Hamid Berenji, Paul Cohen, Didier Dubois, John Fox, John Lemmer, Henri Prade, David Spiegelhalter, Philippe Smets, and Lotfi Zadeh 1988 "Responses to An AI view of the treatment of uncertainty" by Alessandro Saffiotti." In The Knowledge Engineering Review, March, 1988, Vol. 3 (1), pp.59-86.

[Durkin 1994] John Durkin 1994 "Expert Systems - Design and Development." Macmillan Pub. Comp., New York.

[Dvorak & Kuipers 1991] Daniel Dvorak, and Benjamin Kuipers 1991 "Process Monitoring and Diagnosis : A Model-Based Approach." In IEEE Expert - Intelligent Systems and Their Applciations, June, 1991, Vol. 6, pp.67-74.

[ECAI88 1988] European Conference on Artificial Intelligence (9th : Munich), Ed. By Bernd Radig, Yves Kodratoff, Birgit Veberreiter, and Klaus Peter Wimmer 1988 "(ECAI88) Proceeding of the 8th European Conference on Artificial Intelligence - Munich, August 1-5, 1988." Pitman, London.

[ECAI92 1992] European Conference on Artificial Intelligence (10th : Vienna), Ed. by Bernd Neumann 1992 "(ECAI92) Proceeding of the 10th European Conference on Artificial Intelligence - Vienna, Austria, Aug. 3-7, 1992." John Wiley & Sons, Ltd.

[Escobedo, Smith & Caudell 1993] Richard Escobedo, Scott D. G. Smith, and Thomas P. Caudell 1993 "A Neural Information Retrieval System." In The International Journal of Advanced Manufacturing Technology, 1993, Vol. 8, No. 4, pp.269-274

[EWSL88 1988] European Working Session on Learning (3rd: 1988: Turing Institute University of Strathclyde), Ed. by Derek Sleeman, and James Richmond 1988 "(EWSL88) Proceedings of the Third European Working Session on Learning - Turing Institute, Glasgow, 3-5 October 1998." EWSL.

[EWSL89 1990] European Working Session on Learning (4th: 1989: Montpellier), Ed. by Katharina Morik, Jean Sallantin, and Joel Quinqueton 1990 "(EWSL89) Proceedings of the Fourth European Working Session on Learning, Montpellier, 4-6 December 1989." Pitman, London.

[Fattah & O'Rorke 1993] Yousri el Fattah, Paul O'Rorke 1993 "Explanation-Based Learning for Diagnosis." In Machine Learning, 1993, Vol. 13 (1), pp.35-70.

[Forsyth & Naylor 1986] Richard Forsyth, and Chris Naylor 1986 "(IBM PC BASIC Version) The Hitch-Hiker's Guide to Artificial Intelligence." Chapman and Hall, London.

[Forsyth & Rada 1986] Richard Forsyth, and R. Rada 1986 "Machine Learning : Applications in Expert Systems and Information Retrieval." Ellis Horwood Limited.

[Forsyth 1984] Ed. by Richard S. Forsyth 1984 "Expert Systems (Principles and Case Studies)." Chapman and Hall Computing.

[Gale 1986] William A. Gale (AT & T Bell Laboratories) 1986 "Artificial Intelligence & Statistics." Addison-Wesley Pub. Company.

[Gammack & Rowe 1991] John Gammack, and Gene Rowe 1991 "Expert systems: the future is assured." In The Computer Bulletin - For Information systems Professionals, September / October, 1991, Vol. 3 (7), pp.20-21.

[Gardner 1987] Anne von der Lieth Gardner 1987 "An Artificial Intelligence Approach to Legal Reasoning." The MIT Press.

[Glorioso & Osorio 1980] Robert M. Glorioso, and Fernando C. Colon Osorio 1980 "Engineering Intelligent Systems (Concepts, Theory, and Applications)." Digital Press.

[Grefenstette 1993] John J. Grefenstette 1993 "Genetic Algorithms." In IEEE Expert - Intelligent Systems and Their Applications, October, 1993, Vol. 8 (4), pp.5-8.

[Gruber 1992] Thomas Gruber 1992 "Learning Why by Being Told What." In IEEE Expert - Intelligent Systems and Their Applications, August, 1992, pp.65-75.

[Hayes-Roth, Waterman & Lenat 1983] Frederick Hayes-Roth, Donald A. Waterman, and Douglas B. Lenat (ed.) 1983 "(Teknowledge Series in Knowledge Engineering) Building Expert Systems (Vol. I)." Addison-Wesley Pub. Comp., Inc., Reading.

[Hillis 1985] W. Daniel Hillis 1985 "(An ACM Distinguished Dissertation, 1985) The Connection Machine." The MIT Press, Cambridge, Massachusetts, London, England.

[Huang & Wang 1993] Hsin-Hao (Tom) Huang, and Hsu-Pin (Ben) Wang 1993 "Machine Fault Classification Using an ART 2 Neural Network." In The International Journal of Advanced Manufacturing Technology, 1993, Vol. 8, No. 4, pp.194-199.

[Irani, Cheng, Fayyad & Qian 1993] Keki B. Irani, Jie Cheng, Usama M. Fayyad, and Zhaogang Qian 1993 "Applying Machine Learning to Semiconductor Manufacturing." In IEEE Expert - Intelligent Systems and Their Applications, February, 1993, Vol. 8 (1), pp.41-47.

[Jackson 1992] Mary Jackson 1992 "Understanding Expert Systems : Using Crystal." John-Wiley & Sons.

[Jain & Mosier 1992] Piyush K. Jain, and Charles T. Mosier 1992 "Artificial Intelligence in Flexible Manufacturing." In The International Journal of Computer Integrated Manufacturing, Nov. - Dec., 1992, Vol. 5, No. 6, pp.378-384.

[Kanal & Lemmer 1986] L. N. Kanal, and J. F. Lemmer 1986 "(Machine Intelligence and Pattern Recognition 4) Uncertainty in Artificial Intelligence." North Holland.

[Kanal & Lemmer, 1988] L. N. Kanal, and J. F. Lemmer 1988 "(Machine Intelligence and Pattern Recognition 5) Uncertainty in Artificial Intelligence." North Holland.

[Khanna 1990] Tarun Khanna 1990 "Foundations of Neural Networks." Addison-Wesley Pub. Company.

[Kodratoff 1988] Yves Kodratoff 1988 "Introduction to Machine Learning." Pitman.

[Kramer 1991] Mark A. Kramer 1991 "First International Workshop on Principles of Diagnosis." In IEEE Expert - Intelligent Systems and Their Applciations, June, 1991, Vol. 6, pp.86-87.

[Lackinger & Nejdl 1993] Franz Lackinger, and Wolfgang Nejdl 1993 "Diamon: A Model-Based Troubleshooter Based on Qualitative Reasoning." In IEEE Expert - Intelligent Systems and Their Application, February, 1993, Vol. 8 (1), pp.33-40.

[Leonard & Kramer 1993] James A. Leonard, and Mark A. Kramer 1993 "Diagnosing Dynamic Faults Using Modular Neural Nets." In IEEE Expert - Intelligent Systems and Their Applications, April, 1993, Vol. 8 (2), pp.44-53.

[Lerner 1972] A. Ya Lerner 1972 "Fundamentals of Cybernetics." Chapman and Hall, London.

[Leung & Lam 1988] K. S. Leung, and W. Lam 1988 "Fuzzy Concepts in Expert Systems." In Computer, September, 1988, pp.43-56.

[Levi, Perschbacher, Hoffman, Miller, Druhan & Shalin 1992] Keith R. Levi, David L. Perschbacher, Mark A. Hoffman, Christopher A. Miller, Barry B. Druhan, and Valerie L. Shalin 1992 "An Explanation-Based-Learning Approach to Knowledge Compilation : A Pilots Associated Application." In IEEE Expert - Intelligent Systems and Their Applications, June, 1992, pp.44-51.

[Liu & Iyer 1993] T. I. Liu, and N. R. Iyer 1993 "Diagnosis of Roller Bearing Defects Using Neural Networks." In The International Journal of Advanced Manufacturing Technology, 1993, Vol. 8, No. 4, pp.210-215.

[Low, Lui, Tan & Teh 1991] B. T. Low, H. C. Lui, A. H. Tan, and H. H. Teh 1991 "Connectionist Expert System with Adaptive Learning Capability." In IEEE Transactions on Knowledge and Data Engineering, June, 1991, Vol. 3 (2), pp.200-207.

[Luger & Stubblefield 1989] George F. Luger, and William A. Stubblefield 1989 "Artificial Intelligence and the Design of Expert Systems." The Benjamic / Cummings Publishing Company, Inc.

[Mark, Robert & Simpson 1991] William S. Mark, Robert, and Robert L. Simpson, Jr 1991 "Knowledge-Based Systems : An Overview." In IEEE Expert - Intelligent Systems and Their Applciations, June 1991, Vol. 6 (3), pp.12-17.

[McDonald 1989] Carlton McDonald 1989 "Machine learning: a survey of current techniques." In Artificial Intelligence Review, 1989, Vol. 3, pp.243-280.

[McDowell, Kramer & Davis, 1991] James K. McDowell, Mark A. Kramer, and James F. Davis 1991 "Knowledge-Based Diagnosis in Process Engineering." In IEEE Expert - Intelligent Systems and Their Applciations, June, 1991, Vol. 6, pp.65-66.

[Michalski, Carbonell & Mitchell 1984] Ryszard S. Michalski, Jaime G. Carbonell, and Tom M. Mitchell 1984 "(Symbolic Computation) Machine Learning - An Artificial Intelligence Approach." Springer-Verlag, Berlin Heidelberg, New York, Tokyo.

[Mirzai, 1990] A. R. Mirzai ; Artificial Intelligence : Concepts and Applications in Engineering ; Chapman and Hall Computing.

[Monostori & Barschdorff 1992] L. Monostori, and D. Barschdorff 1992 "Artificial Neural Networks in Intelligent Manufacturing." In Robotics & Computer-Integrated Manufacturing (An International Journal), December, 1992, Vol. 9, No. 6, pp.421-437.

[Moore 1980] Robert Carter Moore 1980 "(Outstanding Dissertations in the Computer Science - A Continuing Garland Research Series) Reasoning from Incomplete Knowledge in a Procedural Deduction System." Garland Pub., Inc.

[Munakata 1993] Toshinori Munakata 1993 "Practical AI : Where it's been, and where it is now." In IEEE Expert - Intelligent Systems and Their Applications, April, 1993, Vol. 8 (2), pp.3-5.

[NATO 1971] North Atlantic Treaty Organization, ed. by N. V. Findler, and Bernard N. Meltzer 1971 "(Edinburgh) Artificial Intelligence and Heuristic Programming." Edinburgh University press.

[Naylor 1983] Chris Naylor 1983 "(Artificial Intelligence for the Aspiring Microcomputer) Build Your Own Expert System." Sigma Technical Press.

[Ng 1991] Hwee Tou Ng 1991 "Model-Based, Multiple-Fault Diagnosis of Dynamic, Continuous Physical Devices." In IEEE Expert - Intelligent Systems and Their Applications, December, 1991, pp.38-43.

[Nilsson 1982] Nils J. Nilsson 1982 "(Symbolic Computation) Principles of Artificial Intelligence (with 139 figures)." Springer-Verlag Berlin Heidelberg, New York.

[Norman 1991] Donald A. Norman 1991 "Approaches to the study of intelligence." In Artificial Intelligence, 1991, Vol. 47, pp.327-346.

[O'keefe 1993] Robert M. O'keefe 1993 "Expert system verification and validation: a survey and tutorial." In Artificial Intelligence Review, February, 1993, Vol. 7 (1), pp.3-42.

[O'Neil 1978] Harold F. O'Neil, Jr. 1978 "Learning Strategies." Academic Press.

[O'Shea, Self & Thomas 1987] Ed. by Tim O'Shea, John Self, and Glan Thomas 1987 "Intelligent Knowledge-based Systems (An Introduction)." Harper & Row, Pub.

[O'shea & Eisenstadt 1984] Tim O'shea, and Marc Eisenstadt 1984 "Artificial Intelligence : Tools, Techniques, and Applications." Harper & Row, Publishers, New York.

[Padalkar, Karsai, Biegl, Sztipanovits, Okuda & Miyasaka 1991] Samir Padalkar, Gabor Karsai, Csaba Biegl, Janos Sztipanovits, Koji Okuda, and Nobuji Miyasaka 1991 "Real-Time Fault Diagnostics." In IEEE Expert - Intelligent Systems and Their Applciations, June, 1991, Vol. 6, pp.75-85.

[Partridge & Paap 1988] D. Partridge, and K. Paap 1988 "An introduction to learning." In Artificial Intelligence Review, 1988, Vol. 2, pp.79-101.

[Paul 1993] Paul F. M. J. 1993 "Formal Minds and Biological Brains : AI and Edelman's Extended Theory of Neuronal Group Selection." In IEEE Expert - Intelligent Systems and Their Applications, October, 1993, Vol. 8 (4), pp.66-75.

[Paul 1993] Gabriele Paul 1993 "Approaches to abductive reasoning : an overview." In Artificial Intelligence Review, 1993, pp.109-152.

[Pegah, Sticklen & Bond 1993] Mahmoud Pegah, Son Sticklen, and William Bond 1993 "Functional Representation and Reasoning About the F/A - 18 Aircraft Fuel System." In IEEE Expert - Intelligent Systems and Their Applications, April, 1993, Vol. 8 (2), pp.65-71.

[Penalva, Coudouneau, Leyval & Montmain 1993] Jean Michel Penalva, Laurent Coudouneau, and Lydie Leyval, Jacky Montmain 1993 "A Supervision Support System for Industrial Processes." In IEEE Expert - Intelligent Systems and Their Applications, October, 1993, Vol. 8 (4), pp.57-65.

[Pfau-Wagenbauer & Nejdl 1993] Monika Pfau-Wagenbauer, and Wolfgang Nejdl 1993 "Integrating Model-Based and Heuristic Features in a Real-Time Expert System." In IEEE Expert - Intelligent Systems and Their Applications, August, 1993, Vol. 8 (4), pp.12-18.

[Pollock 1989] John L. Pollock 1989 "How to Build a Person : A Prolegomenon." The MIT Press.

[Porter, Bareiss & Holte 1990] Bruce W. Porter, Ray Bareiss, and Robert C. Holte 1990 "Concept Learning and Heuristic Classification in Weak-Theory Domains." In Artificial Intelligence, 1990, Vol. 45, pp.229-263.

[Rauch-Hindin 1985a] Wendy B. Rauch-Hindin 1985 "Artificial Intelligence in Business, Science, and Industry (Vol. I) : Fundamentals." Prentice-Hall.

[Rauch-Hindin 1985b] Wendy B. Rauch-Hindin 1985 "Artificial Intelligence in Business, Science, and Industry (Vol. II) : Applications." Prentice-Hall.

[Rauch-Hindin 1988] Wendy B. Rauch-Hindin 1988 "A Guide to Commercial Artificial Intelligence : Fundamentals and Real-World Applications." Prentice Hall.

[Rich & Knight 1991] Elaine Rich, and Kevin Knight 1991 "Artificial Intelligence (Second Edition)." McGraw-Hill, Inc.

[Rzevski & Adey 1991] G. Rzevski, and R.A. Adey 1991 "Applications of Artificial Intelligence in Engineering, Vol. VI." Computational Mechanics Publications / Elsevier Applied Science.

[Rzevski 1989] G. Rzevski 1989 "Artificial Intelligence in Manufacturing." Computational Mechanics Publications / Springer-Verlag.

[Saffiotti 1987] Alessandro Saffiotti 1987 "An AI view of the treatment of uncertainty." In The Knowledge Engineering Review, June, 1987, Vol. 2 (2), pp.75-97.

[Saffiotti 1988] Alessandro Saffiotti 1988 "The treatment of uncertainty in AI: Is there a better vantage point." In The Knowledge Engineering Review, March, 1988, Vol. 3 (1), pp.87-91.

[Schlimmer 1993] Jeffrey C. Schlimmer 1933 "Self-Modeling Databases : Learning and Applying Partial Integrity Constraints to Detect Database Errors." In IEEE Expert - Intelligent Systems and Their Applications, April, 1993, Vol. 8 (2), pp.35-43.

[Schnelle & Mah 1992] Karl D. Schnelle, and Richard S. H. Mah 1992 "A Real-Time Expert System for Quality Control." In IEEE Expert - Intelligent Systems and Their Applications, October, 1992, pp.36-42.

[Schultz, Grefenstette & De Jong 1993] Alan C. Schultz, John J. Grefenstette, and Kenneth A. De Jong 1993 "Test and Evaluation by Genetic Algorithms." In IEEE Expert - Intelligent Systems and Their Applications, October, 1993, Vol. 8 (4), pp.9-14.

[Segre 1992] Alberto Maria Segre 1992 "Applications of Machine Learning." In IEEE Expert - Intelligent Systems and Their Applications, June, 1992, pp.30-34.

[Shapiro 1990] Stuart C. Shapiro 1990 "Encyclopedia of Artificial Intelligence (Vol. I)." Wiley-Interscience.

[Sharma & Conrath 1993] Ravi S. Sharma, and David W. Conrath 1993 "Evaluating Expert Systems: a Review of Applicable Approaches." In Artificial Intelligence Review, 1993, pp.77-91.

[Sheridan 1991] F. K. J. Sheridan 1991 "A survey of techniques for inference under uncertainty." In Artificial Intelligence Review, 1991, Vol. 5, pp.89-119.

[Sikora 1992] Riyaz Sikora 1992 "Learning Control Strategies for Chemical Processes." In IEEE Expert - Intelligent Systems and Their Applications, June, 1992, pp.35-43.

[Simpson 1989] Reported by Alan Simpson 1989 "Real Time Control with ES." In Expert Systems User - The Professionals' Guide to Knowledge-Based System, January, 1989, Vol. 5 (1), pp.10-12.

[Spur & Specht 1992] G. Spur, and D. Specht 1992 "Knowledge Engineering in Manufacturing." In Robotics & Computer-Integrated Manufacturing (An International Journal), Aug. - Oct., 1992, Vol. 9, No. 4/5, pp.303-309.

[Steels & Velde 1989] Luc Steels, and Walter Van De Velde 1989 "Learning in Second Generation Expert Systems in Machine and Human Learning (Advances in European Research, Ed. by Yves Kodratoff, and Alan Hutchinson)." Kogan Page, London.

[Swift 1987] K. G. Swift 1987 "Knowledge-Based Design for Manufacture." Kogan Page.

[Tesauro & Sejnowski 1989] G. Tesauro, and T. J. Sejnowski 1989 "A Parallel Network that Learns to Play Backgammon." In Artificial Intelligence, 1989, Vol. 3, pp.357-390.

[The Open University 1978] The Open University 1978 "(Cognitive Psychology) Learning and Problem Solving [Part 1] : D303 Block 4 Units 22-23." The Open University.

[Tim Rajan 1989] Tim Rajan 1989 "Knowledge Acquisition for model based expert systems." In Expert Systems User - The Professionals' Guide to Knowledge-Based System, May, 1989, Vol. 5 (5), pp.18-21.

[Towell & Shavlik 1993] Geoffrey G. Towell, Jude W. Shavlik 1993 "Extracting Refined Rules from Knowledge-Based Neural Networks." In Machine Learning, 1993, Vol. 13 (1), pp.71-101.

[Vinson, Grantham & Ungar 1992] Jonathan M. Vinson, Stephen D. Grantham, and Lyle H. Ungar 1992 "Automatic Rebuilding of Qualitative Models for Diagnosis." In IEEE Expert - Intelligent Systems and Their Applications, August, 1992, pp.23-30.

[Waterman 1986] Donald A. Waterman 1986 "A Guide to Expert Systems." Addison-Wesley Pub. Comp.

[Whiteley & Davis 1993] James R. Whiteley, and James F. Davis 1993 "Qualitative Interpretation of Sensor Patterns." In IEEE Expert - Intelligent Systems and Their Applications, April, 1993, Vol. 8 (2), pp.54-63.

[Winograd & Flores 1986] Terry Winograd, and C. Fernando Flores 1986 "Understanding Computers and Cognition (A New Foundation for Design)." Ablex Publishing, Corporation, Norwood, New Jersey.

[Winston & Brown 1979] Patrick Henry Winston, and Richard Henry Brown 1979 "Artificial Intelligence (An MIT Perspective, Vol. I)." The MIT Press, Cambridge, Massachusetts, and London, England.

[Winston & Brown 1979] Patrick Henry Winston, and Richard Henry Brown 1979 "Artificial Intelligence (An MIT Perspective Vol. II)." The MIT Press, Cambridge, Massachusetts, and London, England.

[Winston 1984] Patrick Henry Winston 1984 "Artificial Intelligence (2nd ed.)." Addison Wesley.

[WPCSS 1989] Working Party Council for Sciences and Society 1989 "Benefits and Risks of Knowledge-Based Systems : Report of a Working Party Council for Sciences and Society." Oxford University Press.

[Wright & Bourne 1988] Paul Kenneth Wright, and David Alan Bourne 1988 "Manufacturing Intelligence." Addison-Wesley Pub. Comp. Inc.

[Wright 1990] Jim Wright 1990 "ES technology is set to come to the aid of Multimedia interactions." In Expert Systems User - The Professionals' Guide to Knowledge-Based System, February, 1990, Vol. 6 (2), pp.12-14.

[Wu 1993] Xindong Wu 1993 "Inductive Learning: Algorithms and Frontiers." In Artificial Intelligence Review, 1993, pp.93-108.

[Wusteman 1992] Judith Wusteman 1992 "Explanation-Based Learning : A Survey." In Artificial Intelligence Review, 1992, Vol. 6 (3), pp.243-263.

[Ye, Zhao & Salvendy 1993] Nong Ye, Baijun Zhao, and Gavriel Salvendy 1993 "Neural-Networks-Aided Fault Diagnosis in Supervisory Control of Advanced Manufacturing Systems." In The International Journal of Advanced Manufacturing Technology, 1993, Vol. 8, No. 4, pp.200-209.

[Yerramareddy, Tcheng, Lu & Assanis 1992] Sudhakar Yerramareddy, David K. Tcheng, Stephen C-Y. Lu, and Dennis N. Assanis 1992 "Creating and Using Models for Engineering Design : A Machine -Learning Approach." In IEEE Expert - Intelligent Systems and Their Applications, June, 1992, pp.52-59.