Def Sci J, Vol 37, No. 4, October 1987, pp 443-456

# Implementation of Combat Simulation Through Expert Support Systems

M.R. Subramanian

*Combat Vehicles R&D Establishment, Avadi, Madras-600 054*

**ABSTRACT**

The battlefield simulation is often faced with a bewildering array of conflicting stresses and challenges. Communication is currently slower and more costly than computation. Expert System technologies such as production rule systems allow one to acquire and represent the collection of heuristic rules in computer compatible form. The system also include master control programs that determine the order in which these rules should be applied against the monitored system performance to arrive at appropriate system control. These expert systems are used in two nodes, both as an intelligence assistant to the expert, amplifying the capacity and quality of his work, and as a surrogate for an expert when he is not available. An Expert support System (ESS) designed and developed for combat simulation has been enumerated in this article. The quality and the reliability of the inferred tactical situation is improved by using PROLOG. This formal AI language is used for validating and checking sensor detections for consistency and logical plausibility. The supremacy of PROLOG for creating and interrogating a data base helps maintaining a reasonably coherent feature of the tactical situation. The perils and pitfalls of tackling with expert systems have also been underscored.

## 1. INTRODUCTION

Command, Control and Communications ($C^3$) has always been a vital aspect of military operations. Recent technological advances have had a significant impact in such sophisticated areas as missiles, sensors and satellites for communications and surveillance. Today's weapon systems have greater range, speed and accuracy. The real size of the battlefield has increased. Sensor coverage and capabilities now overlap and need careful coordination. Consequently command and control of forces will depend to an unprecedented degree on communications and the ability to process information. Recognition, formulation and solution of military strategy and tactics

Received 23 June 1987

present considerable difficulties[1]. The main difficulty stems from the nature of battles themselves. They are non-repetitive, destructive experiments. They are not adequately observed and recorded. Computerized war-gaming and combat simulation represent a very fertile area for Artificial Intelligence (AI) applications.

Information about friendly forces is often communicated with jammable channels. The information about hostile forces is derived from sensors which may be erroneous in degree or classification. Lastly, the environment consists of the divisions in the battelfield which demand changes in tactical deployment and tactical doctrines. The combat requires abstraction into mathematical and symbolical models. The criteria for choosing a model are simplicity and appropriateness. Processes which are clearly heuristic should be modelled by expert systems, while processes which are deterministic and continuous should be modelled as dynamic systems.

An attempt is made towards a methodology in this paper for target acquisition in combat simulation through Expert Support Systems (ESS). These are computer programs that use specialised symbolic reasoning to help people solve difficult problems well. A taxonomy has been developed combining simulation and ESS and the capabilities of such an approach has been demonstrated.

## 2. TARGET ACQUISITION – NEED FOR FIGHTING VEHICLES

CVRDE is entrusted with the prestigious task of designing and developing 'Arjun' Main Battle Tank for Indian Army. Apart from prototype development, systems integration should be viewed in proper perspective and implemented effectively. Countries are beginning to look at lighter and cheaper tanks and to consider other criteria than simply the classic trio of armour, gunpower and mobility. The currently fashionable aspect is 'survivability' which means the ability to go to a battelfield and come back again after performing the task. The current HOT missile which, by now is in the forefront of technology can penetrate one metre of homogeneous steel with ease. The thought of having to put one metre of steel all around a tank is ridiculous. Compound armour, still in the developing stage has not done more than delaying the inevitable. Therefore the tank must rely more upon other attributes than simple thickness. Its profile must be lower, its performance must display greater agility and its crew must be given as much technical aid in order to alert them to potential dangers. But the technical aid leads to a profusion of 'black boxes' inside the tank, detectors to announce the enemy's use of laser beams, infrared sensors and radar. And it is these very devices that take up valuable space escalating the cost of the fighting vehicle. Moreover in the light of recent experiences and developments, one of the first priorities has to be the question of defending against attack by anti-armour helicopter. Therefore, improvements in fuzing mechanisms, metallurgy for long gun tube life, higher breech pressure and better propellants for added range must be given priorities.

But above all, land warfare does not end with development and deployment of these costly tanks. It is essentially about keeping in contact with rapidly moving forces and the front line. When communication is lost, resulting in an armoured division

misdirected or a counter attack ill-timed, the whole plan can be reduced to shreds in minutes. The essence of $C^3I$ (Command, Control, Communications, Intelligence) is about information and control for decision making. The term 'control' requires the order to be sent, received, understood and acknowledged resulting in communication.

The type of communication will thus need to be assessed carefully, be it flares, smoke signals or radio. Jamming an enemy radar only eliminates a single weapon. By jamming the enemy's $C^3I$ system, a complete arsenal can be wiped out. It must be said unfortunately that progress and improvements made in the area of tank mobility and command facilities since World War II can only be described as modest. Time consuming target acquisition and designation procedures, recognition errors and complicated communications were, and still are, the reasons why tanks move slowly through unknown terrain instead of charging at top speed as in advertisements and in demonstrations. Mobility becomes even more of a problem at midnight because of the low level performance of the driver's night vision equipment. It is only in this context that combat simulation for target acquisition enumerated in this article through AI techniques can be of immense value and needs careful consideration. An AI system uses computers to manipulate knowledge by employing reasoning techniques. Reasoning interwines logic with knowledge to make decisions, reach conclusions and to solve problems. As the technology continues to advance and the tools become easier to apply, the integration of AI into the military environment will escalate dramatically.

## 3. REQUIREMENTS FOR ARTIFICIAL INTELLIGENCE

War gaming itself has existed for over a century stretching back to the sand tables of European general staff. Simulation of battles (analytic methods of estimating their outcome, without any intervention of human players) began with Lanchester in the 1900s who wrote differential equations for attrition of forces engaged in frontal attack. These equations are the underpinnings of any large scale computer simulation, but there is no scope beyond an initial resource allocation of forces.

AI is well suited for performing the tasks such as hypotehsis formulation, pattern recognition, planning, scheduling and resource allocation performed in an environment full of uncertainty and incomplete, distorted or disguised information.

Winston defines AI as the study of ideas which enables computers to do the things that make people seem intelligent. AI systems attempt to accomplish this by dealing with qualitative as well as quantitative information, ambiguous and 'fuzzy' reasoning and rules of thumb that give good but not always optimal solutions[2].

## 4. FRAMEWORK FOR EXPERT SUPPORT SYSTEMS

Another way to characterize AI is not in terms of what it attempts to do, but in terms of the programming techniques and philosophies that have evolved from it. In light of the insights garnered from the field of AI, a distinction between expert systems will be exaggerated, as they are often conceived and a variation of expert systems

which we call Expert Support Systems (ESS). While both systems use the same techniques, ESS help people (the emphasis is on people) solve a much wider class of problems. This is done by pairing the human with the expert system in such a way that the expert system provides some of the knowledge and reasoning steps, while the human provides over-all problem solving direction as well as specific knowledge not incorporated in the system. Much of this knowledge may be imprecise and will remain below the level of consciousness, to be recalled to the conscious level of the decision maker only when it is triggered by the evolving problem context.

Simon separated decision making into three phases : Intelligence, design and choice[3]. A structured decision is one where all three phases are fully understood and 'computable' by the human decision maker. We can extend this distinction, for our pruposes, by taking Alan Newell's insightful categorisation of problem solving which consists of goals and constraints, state-space, search control knowledge and operatios[4]. The provision of tools like analyst's work bench and advanced AI techniques are effective in conventional data processing. When the data are incompletely represented and the goals and constraints are only partially understood, Decision Support System use 'what-if' analysis. The users follow a flexible strategy of proposing an action, letting the computer predict its consequences and then deciding what action to propose next. Two aspects which make expert systems essential are: (a) The decisions are made by humans on the basis of limited and inaccurate information. (b) Large number of complex entities interact in complex ways in dynamical systems.

Samuel Johnson said, 'Knowledge is of two kinds: We know a subject ourselves, or we know where we can find an information upon it.' An expert has both kinds of knowledge. He can analyse a problem, assemble facts, use knowledge to infer other facts, evaluate and postulate decisions, explain his reasoning and learn. An expert system attempts to emulate an expert but does not necessarily model precisely his processes. Like a human expert, an expert system may occasionally err. Moreover, there are many levels of expertise. What is important, in these cases is to design expert systems with very good and deeply embedded 'user interfaces'. We should focus our attention on designing systems that support expert users rather than replacing them. A good ESS should be both accessible and malleable.

### 4.1 Need for Knowledge Representation

Decision making for combat simulation and modelling aggregate available information about forces both friendly and hostile, their dynamics, plans and characteristics as well as the environment. In the early modelling efforts, 'if-then-else' statement of FORTRAN represented the tactical knowledge. The entire state of simulation could be determined by which rule was being checked.

Every new situation required a new FORTRAN decision tree. Soon the new tree looped back upon itself to form a decision network. New variables had to be added to keep a track of how many times a path in the network had been followed. Keeping track of the tactics for even a minimally realistic combat simulation became impossible. The goal of 1957 designers of FORTRAN was to eliminate programming. FORTRAN

was an automatic coding system, designed to allow programs to be replaced by quasi-mathematical formulas. As the designers understood it, their goal was largely achieved. But, of course the same problems of designing, coding, debugging and testing familiar in assembly programming simply re-emerged at a new and higher level[5]. The two problems of the decision network are: (a) The structure is unmodifiable, unmaintainable and almost impossible to check for correctness or completeness. The user interface is very bad. (b) The most crucial problem is that the decisions about which tactics are applicable at any point in time and the tactics themselves are in exactly the same place and form. The control structure for the rules and the rules themselves are identical.

The separation of the control structure from the tactical knowledge itself makes it easy for the non-computer-oriented users to write and maintain tactics. Knowledge engineering tools are the right ones to use for modelling this aspect of simulation. There are a variety of possible ways of encoding knowledge. These include state-space representations, logic based representations, procedural representations, semantic nets, production systems and frame systems. Although many of these approaches may be mapped on to each other, logic based representation allows us to directly encode knowledge as inference.

The thrust of both simulation and expert systems is to provide a computer based model for decision making. Both attempt to model the uncertainty inherent in the system under consideration. The software produced for both models must access and interact with other software systems, for instance, a database. What makes simulation

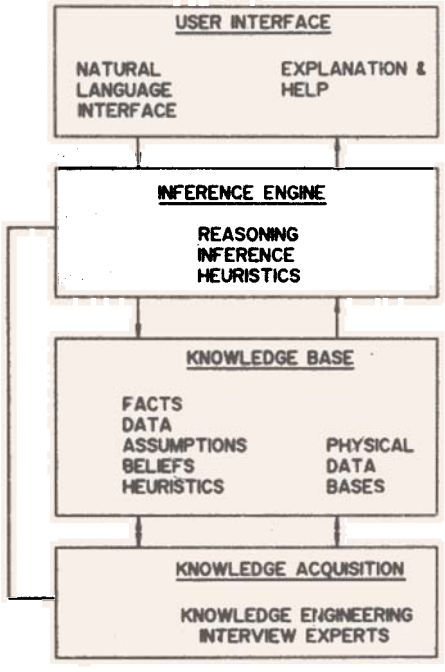


**Figure 1.** Structure of an expert system.

and expert systems similar is that both are based on a modular representation of a system with an inference mechanism that drives this representation (Fig. 1). This similarity is, being exploited. However, the cross-fertilisation between the two areas

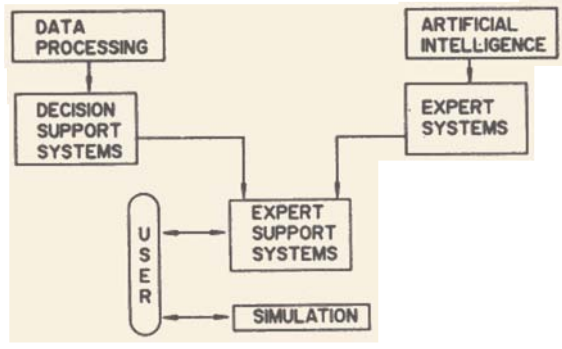


**Figure 2.** Taxonomy for simulation with expert support system.

will not solely rely on this similarity. Expert systems that execute and use the results from simulations are of increasing interest to knowledge engineers (Fig. 2). Rather than test an expert system on a user or a real environment, the expert system can be tested on a simulation. Not only is development time reduced, but also testing can be more comprehensive. The experimenter uses four classes of knowledge :

(i) Knowledge about the domain in which the model is built.

(ii) Knowledge about statistics – how to interpret results, what measurements are appropriate etc.

(iii) Knowledge about the simulation (and hence the real system) behaviour.

(iv) Knowledge of the language used for implementation.

### 4.2 Suitability of Prolog as a Production System

In the case of procedural languages like COBOL, FORTRAN, PL/1 etc., the knowledge embodied in the computer prgoram is expressed as the specific steps for doing it. The order and interdependence between two lines makes the programs difficult to modify. An alternative approach has emerged as a result of an interesting blend of computer science and formal linguistics. This is based on the so called 'production systems' which enable the knowledge of the program to be expressed in a form that is independent of its execution sequence[6]. A production system consists of a database and a collection of production rules. A sign of the potential viability of production systems has been the rapidly increasing popularity of the language PROLOG. It is a 'backward inferencing' production system. PROLOG programs are written to deduce backwards from a specified goal to the available facts in the program's database. Partial procedurality may be introduced through a special device called a 'cut'. PROLOG programs may be written as purely declarative rules without using the cut, but may be made increasingly procedural through extended uses of the cut

operator[7]. Several expert systems have been developed based on PROLOG and it has been adopted by the Japanese Fifth Generation Project. The advent of PROLOG is relatively recent, most early expert systems having been written in LISP. The reluctance in some circles to accept PROLOG at the present time seems to be prompted by the relative dearth of powerful programming environments for the language.

PROLOG supports the need of expert systems to use inexact or probabilistic reasoning by permitting association of probabilities or certainty factors with assertions and rules. The derived conclusion itself yields an associated probability. PROLOG holds more promise than LISP in the area of tools for performance measurement and tuning. PROLOG also provides better support for incremental modification of its knowledge base owing to its rule based programming paradigm.

## 5. DRAWING INFERENCE FROM AVAILABLE EVIDENCE

The process of drawing inference from available evidence is done by either extrapolation or pooling. In extrapolation, the scope of a single body of evidence is extended from low level to high level hypotheses. In pooling, a conclusion is based on a consensus of hypotheses, each based on disparate sources of knowledge. Most operational AI systems use a conventional or modified Bayesean approach, probably because it has proven useful and is well understood. However, the Bayesean approach has certain inherent flaws. AI systems attempt to draw logical inferences from both numerical data that is often subject to error or inexactitude and other non-numerical data that may be abstract and qualitative in nature. Hence much study has been given to the handling of uncertainty. Two of the newest approaches are Dempster Shafer Theory and Theory of Fuzzy Sets. Dempster Shafer Theory is an extension of Bayesean probability theory to include uncertainty. Fuzzy Set theory is a similar extension, but it starts from classical set theory.

### 5.1 Advantages of Dempster Shafer Theory

Clearly, there are advantages in applying the Dempster Shafer confidence interval to a rule based reasoning process over Bayesean probabilistic approach.

(a) When there is no clear reason to prefer one proposition over another, judgement can be suspended. Therefore, a threat warning system can express some belief that a threat is at a given location without being required to speculate as to its type. A Bayesean approach would require that a precise probability be assigned to each possible threat type, no matter how poor the sensor data and no matter how meagre the statistical data to make such an estimate.

(b) The ability to represent and reason from information characterised by varying degrees of ignorance is critical. A Bayesean approach does not properly capture this aspect of information but instead forces it into a form that exaggerates its precision. The Dempster Shafer approach does not require such overstatement.

(c) Using Dempster Shafer, belief arising from ambiguous evidence need not be arbitrarily apportioned. Conflict can be handled without regard to requirements for unitary probability for the sum of all possible outcomes.

## 6. DEVELOPMENT OF A PROTOTYPE EXPERT SYSTEM SHELL

Our aim in developing an expert system shell was to analyse the requirements of an ESS for integration of knowledge from disparate sources (called sensor fusion). It could contribute to the Army's knowledge by processing our intelligence information and utilising the associated decisions for delivering weapons to incapacitate enemy forces. The methodology for decision dissemination is depicted in Fig. 3. The shell provides a tactical intelligence system performing in a simulated battlefield environment. The heuristic decision rules reduce the computational resources needed. The implementation of the scheme is based on the use of Dempster Shafer notation for the sensor output[8]. It was decided to use PROLOG as the production system rather than decision tables as the representation is simpler and more general in its application.

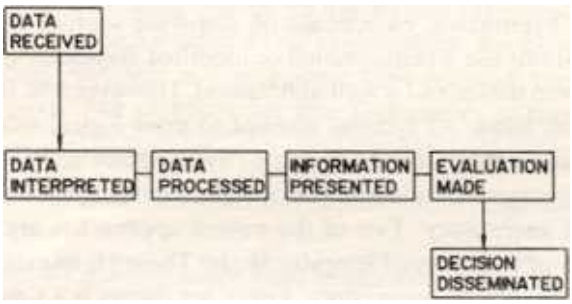


**Figure 3.** Methodology for target acquisition.

An expert system shell is the backbone of an expert system without its expert knowledge. It typically provides a collection of inference mechanisms and facilities for providing explanations, aids for system development, debugging etc. The evolution of AI Shells is given in Fig. 4. PROLOG provides good support for expert system shells since its pattern matching, rule based knowledge representation, backward

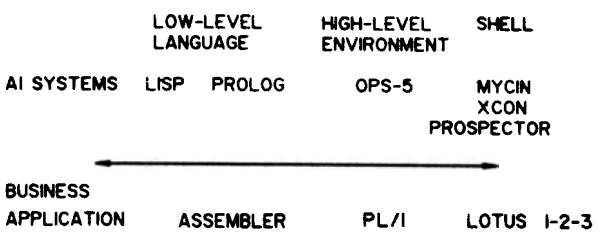


**Figure 4.** Evolution of AI shells.

chaining and backtracking mechanisms are directly inherited by the shell. The prototype shell supports the interactive process of knowledge engineering and provides the inferencing capability without loss of functionality, robustness and power. The shell written in PROLOG 1 (from Expert Systems Limited) runs on 8088/8086 based microcomputer in an MS-DOS environment with a minimum of 128 K memory. It is the result of about 6 months effort. The shell consists of approximately 700 lines of code and occupies 39680 bytes. The heuristics resulted in an average 40 per cent saving in CPU time.

## 7. TARGET ACQUISITION AND SENSOR REPRESENTATION

The four stages in target acquisition in combat situation are :

(i) Detection – The discovery of a potential target because of contrast or discontinuity with the surroundings.

(ii) Recognition – Determination of the class of target.

(iii) Identification – When the true identity is established.

(iv) Location – Where the target's position is fixed with the accuracy needed.

Heuristic search techniques developed in the field of AI form the basis of investigation into the problem of object identification and sensor fusion. The sector processor collects data from the reporting sensors and performs sensor fusion. The fusion processor then passes its results onto a cluster analyser, which cluster the data by geographic location. These results are finally taken by a cluster maching processor for information display. The simulation takes place in a sectored square battlefield where sector designators represent army divisions. Dempster Shafer notation is used for sensor fusion. Cluster matching and analysis is related to implementing frame/slot knowledge representation scheme by Minsky[9]. The issue involved with cluster matching is determination of the best fit between the frames and the empirical data.

The sensor fusion programs integrate knowledge from different sources constantly and evaluate the quality of the information provided to them[10]. The indicator for invalid information, when reaches a threshold value proves incidence of sensor disagreement and further information is ignored. The programs have the ability to delete ineffective heuristics but cannot modify them. This increases the speedup by avoiding fruitless computation in the current situation. The algorithm parameters can be modified only after full simulation run.

Sensor output is in the form of a confidence level list. They are combined by forming the 'orthogonal sum'. The presentation of the list is as per Dempster Shafer notation and has the form:
OBJECT IDENTIFIER, cTANK,cPC,cRADAR,cSAM,cSSM,cTRC, cHELICOPTER,cDRONE,cRPV,cUNCERTAINTY LEVEL.

Infrared size and motion detectors and frequency and modulation analyzers are the variety of techniques for ground based and aerial surveillance. Each target object has a confidence level assigned to it by each one of the sensors applicable. Sensor

fusion combines the confidence lists from sensors. The object data must be aggregated to form division size clusters. The fused data is formed into cluster totals by object type. The object type is that which has the highest confidence level after the sensor fusion. For cluster/division matching, unique attributes of each division are obtained by comparing the known division values for a given attribute and finding the minimum number of objects which would uniquely identify a division for that object type. Confidence levels of the matches ranged from 1.00 to 0.45. Rules concerned with sensor fusion and cluster analysis are given in Table 1. Results obtained from the two programs are given in Table 2.

**Table 1. Rules concerned with sensor fusion/cluster analysis**

r1 (if
Level 1 sensor agrees with more than half of the level 2 sensors
then
'Confidence distribution' is – 'CLASS I').

r2 (if
Disagreeing sensors add 1 to their tally of disagreements,
then
'reversal figure' is – 'CLASS II').

r3 (if
Agreeing sensors reduce 1 from reversal figure
then
'confidence level' is – 'CLASS III').

r4 (if
80 % of the level 2 sensor disagree with level 1 sensor
then
'orthogonal sum' is – 'CLASS IV').

r5 (if
Level 1 and level 2 sensors agree completely 100% on the primary object type
then
'Sensor Fusion' is – deferred).

r6 (if
None of the heuristic results apply
then
'Fusion Result' is – sum of all data).

r7 (if
Minimum number of objects to identify a division is – 'Type 1
then
'Unique attribute' is – '1+Type 1').

r8 (if
'Low confidence level of objects are 'Type 2'
then

Unique attribute is – 'Type 1 – Type 2).

r9 (if
Cluster groups subtracted by low confidence level objects
is – 'Type 3'
then
'Unique Attribute of the division is – 'combination of
object types').

**Table 2. Results of sensor fusion and cluster analysis programs**

(a) Division parameters

| Equipment | DIV 1 | DIV 2 | DIV 3 | DIV 4 |
|---|---|---|---|---|
| Tanks | 22 | 51 | 28 | 6 |
| PC | 12 | 9 | 38 | 40 |
| Radar | 3 | 2 | 1 | 5 |
| SAM | 2 | 3 | 4 | 3 |
| SSM | 2 | 2 | 1 | 1 |
| TRC | 1 | 2 | 1 | 1 |
| Helicopter | 10 | 12 | 22 | 4 |
| Drone | 3 | 2 | 4 | 3 |
| RPV | 1 | 2 | 2 | 1 |

(b) Cluster parameters

| Equipment | Clus 1 | Clus 2 | Clus 3 | Clus 4 |
|---|---|---|---|---|
| Tanks | 24 | 48 | 48 | 3 |
| PC | 12 | 10 | 15 | 22 |
| Radar | 2 | 1 | 1 | 4 |
| SAM | 1 | 2 | 4 | 2 |
| SSM | 1 | 3 | 2 | 1 |
| TRC | 9 | 1 | 1 | 4 |
| Helicopter | 9 | 11 | 21 | 4 |
| Drone | 2 | 2 | 3 | 3 |
| RPV | 1 | 1 | 2 | 1 |

(c) Low confidence level objects

| Equipment | Clus 1 | Clus 2 | Clus 3 | Clus 4 |
|---|---|---|---|---|
| Tanklow | 2 | 3 | 3 | |
| PClow | 1 | 2 | 2 | 11 |
| Radarlow | 0 | 0 | 0 | 2 |
| SAMlow | 0 | 1 | 1 | 0 |
| SSMlow | 0 | 0 | 0 | 1 |
| TRClow | 0 | 1 | 1 | 4 |
| Helcprlow | 2 | 2 | 3 | 1 |
| Dronelow | 0 | 0 | 1 | 1 |
| RPVlow | 1 | 1 | 2 | 3 |

## 8. ACTION PLAN FOR FUTURE

*(i) Notation for Query Expression at the Conceptual Level*

Conventional languages had a serious deficiency in the testing and proving area. Software engineering and modular programming improved things somewhat, but in the end debugging and difficult modification would always be expensive companions of FORTRAN, COBOL and the rest. Program maintenance is a euphemism. If a car needed a few mechanics working on it regularly, it would not be called maintenance – it would be rectification of design faults.

It is our belief that in the context of threat warning systems, a wider use of logic would have a positive effect on the data field. It provides not only a conceptual framework for formulating various database concepts but also a tool for implementing them. Three well known data models are relational, hierarchical and network models. In relational model data are assumed to be stored in the form of tables. In hierarchical model data are assumed to be stored in the form of tree structures. And in network model data are assumed to be stored in the form of general graph structures. In comparison with them, the relational approach, grounded in a well established mathematical discipline, is a significant approach to the logical description and manipulation of data. A relational database is manipulated by powerful operators for extracting columns and join them. Such a high degree of logical data manipulation is not readily available in network and hierarchic database system. But the relational databases had a considerable technical snag. The entities in the database had to be retrieved associatively, which either meant a slow down serial search by the von Neumann processor or complex pointer system.

Query Processing

$$\text{QUERY} \xrightarrow{\text{Understanding}} \text{LOGIC} \xrightarrow{\text{Optimisation}} \text{PROLOG} \xrightarrow{\text{Execution}} \text{ANSWER}$$

*(ii)Knowledge Engineering Through Deductive System*

When attempting to mechanise a human reasoning process, such as a pilot's thought process when he fuses the information from several individual displays and draws a conclusion, several difficulties arise. For one, not all pilots would interpret the data the same way, we all base our perceptions and decisions on our own unique experiences. Secondly, part of the process may not even be conscious. It is the task of the knowledge engineer to determine the process by which the experts make their decisions. Third, there is no way to tell that the candidate reasoning process is optimum, hence the emphasis on numerous simulations for process validation in many AI systems. Along this line the foundation of the future work is to combine concepts from three areas of computer science research. From AI and Logic Programming we will use the concept of knowledge representation in deductive question answering systems and the concept of first order predicate calculus; from Data Base Management Systems we will build on the concept of a relational database, and finally, based on associative processing we may avoid the von Neumann bottleneck.

## 9. CONCLUSION

A prototype of expert system shell in PROLOG has been described for problem solving in combat simulation applications. The ESS designed makes the problem solving accessible to users by providing explanation facilities. The ESS is also made to be malleable for users to change data, procedures, goals or strategies at any important point in the problem solving process. Expert system techniques based on heuristic rules are used in this methodology dramatically to extend the capabilities of traditional decision support systems. The demonstration of the prototype proves that problem solving is not optimising but rather satisfying, i.e., they do not necessarily search for the optimal solution ot the problem, but rather for a sub-optimal solution that satisfies the majority of the problem constraints and conditions. Emphasis is laid upon flexibility of tactics, elaborate control structure options and a mild user interface both in simulation and tactical language.

We wish to conclude that expert systems and ESS are in their infancy. Already there is an apparent risk that an expert system will be poorly defined and oversold ; the resulting backlash may hinder progress. There is also a danger of proceeding too quickly and too recklessly. We may very well embed our knowledge (necessarily incomplete at any moment of time) into a system that is only effective when used by the person who created it. If this system is used by others, there is a risk of misapplication. The state-of-the art is such that everyone building an expert system must endure this primitive phase to learn what is involved in this fascinating field. The challenge for scientists is to proceed at an appropriate pace and to harness these tools for the effectiveness of the organisation.

## ACKNOWLEDGEMENTS

The author wishes to express his gratitude to Shri T.K. Das, Director, CVRDE for his encouragement and kind permission accorded for publication of this paper.

He also wishes to express his appreciation of the module on PROLOG provided by Shri C. Ravi Shankar and his able team of CMC Ltd., Secunderabad.

## REFERENCES

1. Chatter Singh, *Def. Sci. J.*, **36** (1), (1986), 35-44.
2. Winston, P.H., Artificial Intelligence, (Addison-Wesley, Reading), 1984.
3. Simon, H.A., The New Science of Management Decision, (Harper & Row, New York), 1960.
4. Newell, A., Reasoning : Problem Solving and Decision Processes, (Nickerson-Erlbaum), 1980.
5. Editor's Report, *Computer,* August, (1986).
6. *Informations Systems,* **8** (1983).
7. Subrahmanyam, P.A., *IEEE Trans. Software Eng.*, **11** (1985).
8. Garvey, T. et al., An inference technique for integrating knowledge from disparate sources, Proc. of the seventh ICJAI, 1981.
9. Minsky, Marvin, A Framework for Representing Knowledge, Mind Design, (Bradford Books, Montegomery), 1981.
10. Ferrante, Richard, Master's Thesis, University of Lowell, Massachussetts, 1983.

## APPENDIX

### DOMAIN INDEPENDENT METAKNOWLEDGE

This important feature exists in this protoype in two forms. The first form of this feature provides the user with the capability of determining the scope of the knowledge in the domain of this system. The facility providing this capability is a list of objects each with a list of possible values that it may take in the domain of the system. This is represented by options – list (<object>(<value>)) or in the more general sense where value is itself an object : options – list (CLASS(OBJECT)) where OBJECT $\Xi$ CLASS AND OBJECT$\subset$OBJECTi = 1,2,----.By inspecting the set OBJECT for a particular CLASS and checking whether a particular OBJECTi is a legal member, the user can determine the limits of knowledge present.

The second form of metaknowledge provides the user with the capability of determining the nature of the knowledge in the domain of the system. That is, the user can determine whether the value of a particular object can be inferred by the system or whether the object can be used to infer information about other objects. From the rules in the knowledge base, an index is created consisting of each separate object contained together with a reference to the rules in which it appears, both in the antecedent part and in the consequent part. This takes the form of : object ([ALIST], [CLIST]).