JULY 1988 LIDS - P - 1788

EVALUATION OF EXPERT SYSTEMS IN DECISIONMAKING ORGANIZATIONS*

Didier M. Perdu
Alexander H. Levis

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139

ABSTRACT

A class of decision aids that is receiving attention in the development community is based on artificial intelligence and especially expert systems. This paper presents a procedure for assessing to what extent the measures of performance of an organization are modified when an expert system is introduced. First, a model of symbolic computation with fuzzy logic, using Predicate Transition Nets, is presented to model the most common kind of expert systems: the consultant expert systems. This model allows to evaluate its response time for a given input. An Air Defense problem in which command and control involves a hierarchical two decisionmaker organization, where the expert system is used as an aid in the fusion of inconsistent information, is then presented. A strategy involving the use of the expert system is compared to two other strategies expected to be used by a decisionmaker facing this problem. Measures of performance (workload, timeliness and accuracy) are evaluated for each of these strategies. The results show that the strategy involving the use of the expert system improves significantly the accuracy of the organization, but requires more time and increases the workload of the decisionmaker using it.

* This work was conducted at the MIT Laboratory for Information and Decision Systems with support provided by the Basic Research Group of the Technical Panel on C^3 of the Joint Directors of Laboratories through the Office of Naval Research under Contract no. N00014-85-K-0782.

EVALUATION OF EXPERT SYSTEMS IN DECISIONMAKING ORGANIZATIONS*

Didier M. Perdu
Alexander H. Levis

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139

ABSTRACT

A class of decision aids that is receiving attention in the development community is based on artificial intelligence and especially expert systems. This paper presents a procedure for assessing to what extent the measures of performance of an organization are modified when an expert system is introduced. First, a model of symbolic computation with fuzzy logic, using Predicate Transition Nets, is presented to model the most common kind of expert systems: the consultant expert systems. This model allows to evaluate its response time for a given input. An Air Defense problem in which command and control involves a hierarchical two decisionmaker organization, where the expert system is used as an aid in the fusion of inconsistent information, is then presented. A strategy involving the use of the expert system is compared to two other strategies expected to be used by a decisionmaker facing this problem. Measures of performance (workload, timeliness and accuracy) are evaluated for each of these strategies. The results show that the strategy involving the use of the expert system improves significantly the accuracy of the organization, but requires more time and increases the workload of the decisionmaker using it.

INTRODUCTION

Decisionmaking processes require the analysis of complex situations and the planning, initiation and control of subsequent responses. These activities are done within some constraints such as time and accuracy and so that an acceptable level of effectiveness be reached. The amount of information handled by decisionmakers is often very large and, in order to maintain performance above a certain level, decisionmaking organizations use decision support systems to help them accomplish their mission. Among them, Expert Systems with their deductive capability and their ability to handle symbolic concepts have proved to be very useful. The aim of this paper is to show to what extent the use of an expert system modifies the measures of performance of a decisionmaking organization. To allow the use of the analytical framework developed for the study of these organizations, an expert system model using Predicate Transition Nets is first defined for the evaluation of the response time. Expert systems are then studied to assess their usefulness in aiding the fusion of possibly inconsistent information coming from different sources. This assessment is done through the analysis of an application involving a two-decisionmaker organization facing this problem of inconsistent information. Three strategies used to solve this problem are described, one of them involving the use of an expert system. Measures of performance reached for each of these strategies are finally evaluated and compared.

1.0 AN EXPERT SYSTEM MODEL USING PREDICATE TRANSITION NETS

Knowledge Based Expert Systems show properties of synchronicity and concurrency which makes them suitable for being represented with the Predicate Transition Net formalism (Genrich and Lautenbach, 1981). The rules of a knowledge base have to be checked in a specific order depending on the strategy used to solve the problem and on the current facts deduced so far by the system in the execution of previous rules. A model of an expert system using production rules to represent knowledge is presented. Some previous work (Gordona and Saitta, 1985) have addressed the modeling of production rules of a knowledge base using Predicate Transition Nets. The model presented here is different because it incorporates explicitly the control done by the inference engine. Fuzzy logic (Zadeh, 1965 and 1983; Whalen and Schott, 1983) is used to deal with uncertainty and Predicate Transition Nets are used to represent the basic fuzzy logical operators AND, OR and NOT that appear in this kind of rules. An extension of the standard inference net formalism is obtained by the combination of these operators which permits to represent the dynamical behavior of an expert system. The obtained net allows the identification of the rules scanned by the system to produce an answer to a specific problem and to deduce its response time depending on the number of rules scanned and on the number of interactions with the user.

1.1 Structure of the Expert System

Knowledge Based Expert Systems, commonly called Expert Systems, are - in theory - able to reason using an approach similar to the one followed by an expert when he solves a problem within his field of expertise. A net model for the most common kind of expert system, the **consultant expert system**, as described by Johnson and Keravnov (1985), is proposed. Most systems engage in a dialogue with the user, the computer acting as a "consultant," by suggesting options on the basis of its knowledge and the symbolic data supplied by the user. Moving from known items of information to unknown information is the vital process of a consultant system. The user of a consultant expert system has "observed" some particular state of affairs within the domain of the system's expertise and submits these observations to the system. Based

* This work was conducted at the MIT Laboratory for Information and Decision Systems with support provided by the Basic Research Group of the Technical Panel on C^3 of the Joint Directors of Laboratories through the Office of Naval Research under Contract no. N00014-85-K-0782.

on the observations, the system makes inferences and suggests new routes of investigation which will yield high grade information. Interactions continue until the system finds the most likely explanation of the observations. The formalism used to represent knowledge in consultant expert systems is the production system model.

There are three distinct components in an expert system, the Knowledge Base, the Fact base, and the Inference Engine.

The Knowledge Base contains the set of information specific to the field of expertise. Knowledge is expressed in a language defined by the expert. The knowledge base is a collection of general facts, empirical rules, and causal models of the problem domain. A number of formalisms exist to represent knowledge. The most widely used is the production system model in which the knowledge is encoded in the form of antecedent-consequent pairs or IF-THEN rules. A production rule is divided in two parts :

- A set of conditions (called left-hand side of the rule) combined logically together with a AND or a OR operator,
- A set of consequences or actions (called also right-hand side of the rule), the value of which is computed according to the conditions of the rule. These consequences can be the conditions for other rules. The logical combination of the conditions on the left- hand side of the rule has to be true in order to validate the consequences and the actions.

An example of a production rule is :

IF the flying object has delta wings AND
the object flies at great speed
THEN the flying object is a fighter plane.

The conditions "the flying object has delta wings" and "the object flies at a great speed" have to be true to attribute the value true the consequence "the flying object is a fighter plane."

The relationships among the rules of a production system can be represented with an inference net. The net shows graphically the logical articulation of different facts or subgoals, and identifies which rules are used to reach a specific goal. Let us consider the following production rules:

if A AND B, then C
if D OR E, then F
if NOT G, then H.

These rules are represented in the inference net formalism on Figure 1.

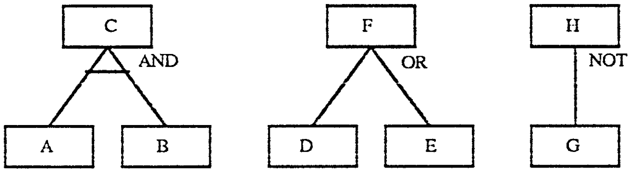

Figure 1 Representation of the logical operators in the inference net formalism

The Predicate Transition Net model developed in this paper is an extension of the inference net formalism and permits the explicit representation of the rules of a knowledge base and the relationships among them.

The fact base, also known as context or working memory, contains the data for the specific problem to be solved. It is a workspace for the problem constructed by the inference mechanism from the information provided by the user and the knowledge base. The working memory contains a trace of every line of reasoning previously used by memorizing all the intermediate results. Therefore, this can be used to explain the origin of the information deduced or to describe the behavior of the system.

The Inference Engine is used to monitor the execution of the program by using the knowledge base to modify the context. It uses the knowledge and the heuristics contained in the knowledge base to solve the problem specified by the data contained in the fact base. In the production system modeled in this paper, the rules are of the kind, A -> B, saying that, if A is valid, B can be deduced. The inference engine selects, validates, and triggers some of these rules to reach the solution of the problem.

In order to deal with uncertainty in items of evidence, fuzzy logic has been implemented in the model to combine logically the conditions of the left-hand side of the production rules. The value of a rule or a fact is either unknown or a number, pi, between 0 and 1, representing the degree of truth associated with it. The operators AND, OR, and NOT execute operations on these degrees of truth as follows:

p1 AND p2 = min(p1, p2)
p1 OR p2 = max(p1, p2)
NOT p1 = 1 - p1.

Among the strategies used by the inference engine to select the rules, forward chaining and backward chaining are the most common. In **forward chaining**, the inference mechanism works from an initial state state of known facts to a goal state. It finds first all the rules that match the context, then it selects one rule based on some conflict resolution strategy, and then execute the selected rule. Facts are inputs to the system. The most appropriate hypothesis that fits the facts is deduced. For **backward chaining**, the system tries to support a hypothesis by checking known facts in the context. If these known facts do not support the hypothesis, the preconditions needed for the hypothesis are set up as subgoals. The process for finding a solution is to search from the goal to the initial state, it involves a depth-first search.

In order to simulate the behavior of an expert system, the process of selection and firing of rules done by the inference engine has been modeled when a backward chaining strategy is used. A trigger is associated with every rule (or operator). A rule is selected by the inference engine when the trigger is activated. Only one rule at a time can be activated and the continuation of the selection and firing process is done according to the result of the rule :

- If the result is unknown, the rule is put in memory and the rule which gives the value of the first unknown precondition is selected.
- If the result is known, the last rule which was put in memory is selected again because the produced result is the value of one of its preconditions.

Let us consider the example where we have two rules:

B => C (1)
A => B (2)

and where the degree of truth of the fact A is known.

The inference engine selects first rule (1). The degree of truth of C is unknown because the degree of truth of B is unknown. Rule (1) is then de-activated and put in memory. Then rule (2) is selected. Since the value of A is known, the value of B is deduced. Rule (1), which is the last to have been put in memory, is selected again and the answer C is obtained.

The process of selection and firing of rules described above is repeated by recursion until the final answer is found; the process can last a long time. In the search for efficiency and performance, unnecessary computations must be avoided. In some cases, there is no need to know the values of all the preconditions of a rule to deduce the value of its consequence. For example, in Boolean logic, if we have the rule :

A AND B => C.

and we know that :

A is false,

then the consequence C is false and there is no need to look for the value of B to conclude that ; the set of rules giving the value of B can be pruned.

In systems using fuzzy logic, this avoidance of unnecessary computations is all the more important as computations are more costly in time and memory storage than in systems using Boolean logic. The problem is that little improvement in performance is obtained, if extra computation is avoided only in the case of complete truth (for the operator OR) or of complete falsity (for the operator AND). The solution lies in the setting of thresholds for certain truth and certain falsity. For example, in the case of the operator AND, if we have:

A AND B => C

and if we know that the degree of truth of A is less than the threshold of certain falsity, then we can deduce that the degree of truth of the consequence C is less than the degree of truth of A and, therefore, less than the threshold of certain falsity. There is no need to know the degree of truth of the precondition B. The thresholds for which no further search is required in the execution of the operators are set to 0.8 for certain truth in the operator OR and 0.2 for certain falsity in the operator AND. A rule or fact having a degree of truth larger or equal to 0.8 (resp. less or equal to 0.2) will be considered to be true (resp. false). Therefore, the logic takes into account the unknown rules or facts.

1.2 Characteristics of the Predicate Transition Nets Used in the Model

Predicate Transition Nets have been introduced by Genrich and Lautenbach (1981) as an extension of the ordinary Petri Nets (Peterson, 1980; Reisig, 1985) to allow the handling of different classes of tokens. The Predicate Transition Nets used in the model have the following characteristics.

Tokens. Each token traveling through the net has an identity and is considered to be an individual of a given class called variable. Each variable can receive different names. For this model, two classes of tokens are differentiated :

(1) The first class, denoted by P, is the set of the real numbers between 0 and 1, representing the degrees of truth of the facts or items of evidence. The names of the individual tokens of these classes will be p, p1, p2.

(2) The second class is denoted by S. The individuals of this class can only take one value. Only one token of this class will travel through the net and will represent the action of the inference engine in triggering the different rules.

Places. Places are entities which can contain tokens before and after the firing of transitions. Three kinds of places are differentiated:

(1) places representing a fact or the result of a rule and containing tokens of the class P or no token at all,

(2) places used by the system as triggers of operators and containing the token of the class S. These places and the connectors connected to these places are represented in bold style in the Figures and constitute the **system net.**

(3) places allowed to contain different kinds of tokens (P and S) and which are used to collect the tokens necessary for the enabling of the transitions of which they are the input places.

The marking of a place is a formal sum of the individual tokens contained in the place. For example, a place A containing a token of the class P, p1 and the token of the class S has the marking M(A) :

M(A) = p1 + S

Connectors and Labels. Each connector has a label associated with it which indicates the kinds of tokens it can carry. A special grammar is used on the labels to define in what way tokens can be carried. The labels of connectors linking places to transitions contain conditions that must be fulfilled for them to carry the tokens. The labels of connectors linking transitions to places indicate what kind of token twill appear in the places after the firing of the transition.

The following notation in labels is used :

When token names are joined by the sign "+" then the tokens defined by these names have to be carried at the same time. For example, the label "p + S" indicates that one token of the class P and one token of the class S have to be carried together at the same time by the connector.

When token names are joined by the sign "," then the tokens defined by these names can be carried at different times but not together. For example, the label "p, S" indicates that either a token of the class P or a token of the class S can be carried.

Mixing of notation is possible. The label "p+S, S" indicates that the connector can carry either a token of the class P and a token of the class S or only one token of the class S.

A connector without label has no constraint on the kind of tokens it can carry.

In some cases, the connector has to carry the token of class S when there is no token of the class P involved in the firing of a transition. The statement "absence of token of the class P" is denoted by the symbol Ø. This symbol is used in the labels, as if it was a class of tokens, in association with the names of the other classes. The symbol Ø is used in the following cases :

(1) The label "S+Ø" means that the connector can carry a token of the class S, if there is no token of the class P.

(2) The label "(S+p), (S+Ø)" means that the connector can carry either a token of the class S and a token of the class P, or a token of the class S, if there is no token of the class P.

Transitions. Transitions have attached to them a predicate which is a logical formula (or an algorithm) built from the operations and relations on variables and tokens in the labels of the input connectors. The value (true or false) taken by the predicate of a transition depends on the tokens contained in the input places of the transition. When the predicate has the value "true", the transition is enabled and can fire. In the model of the consultant expert system, predicates are conditions on tokens of the class P.

A transition without predicates is enabled as soon as all the input places contain the tokens specified by the labels of the connectors.

Transitions with predicates are represented graphically with squares or rectangles. The predicate is written inside. Transitions without predicates are represented with bars as in ordinary Petri Nets.

Firing Process. The conditions of enabling of a transition are: (1) the input places contain the combination of tokens specified by the labels of the connectors, and (2) the predicate of the transition is true. If these two conditions are fulfilled, the transition can fire. In the firing process, tokens specified by the input connectors are withdrawn from the corresponding input places and tokens specified by the output connectors are put in the output places. Let us consider the example shown on Figure 2 :

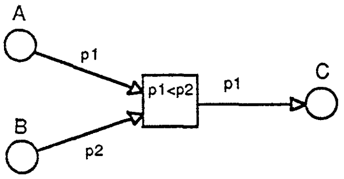

Figure 2 Example of a transition with a predicate

The condition "p1 < p2" written in the transition represented by a square is true when the value of the token named p1 coming from place A is less than the value of the token named p2 coming from place B, as specified by the connectors. In this case, the transition is enabled and can fire; the tokens p1 and p2 are withdrawn from the places A and B and a token p1 is put in place C.

1.3 Logical Operator Models

In order to construct the model of the expert system using Predicate Transition Nets, it is necessary to construct first models of the logical operators AND, OR, and NOT. The results are shown in Figures 3, 4 and 5. Let us describe now what happens in the operator AND (the operators OR and NOT behave in a similar way).

The operator drawn in Figure 3 realizes the operation :

A AND B => C.

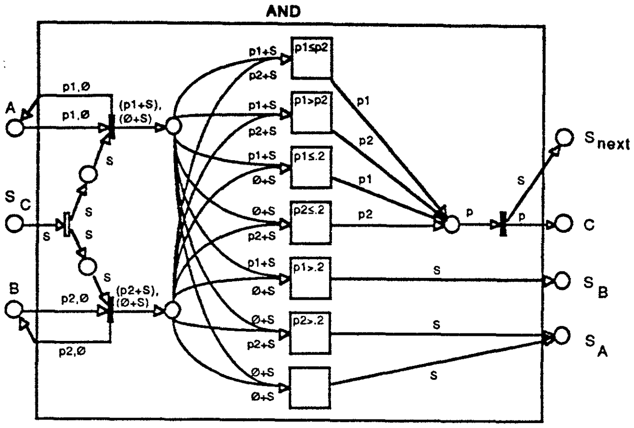

Figure 3 Model of the operator AND

It can be represented as a black box, having three inputs: A, B and S_C (the trigger) and six outputs : C (the result), A, B (memorizing of the input value) and three system places S_A, S_B and S_{next}. Only one of those system places (represented in bold style in the figures) can have a system token at the output. S_{next} will contain a system token, if the result of the operation is known, i.e., if C contains a token of the class P. This shows that the next operation can be performed. If the result is unknown, i.e., the two inputs are not sufficient to yield a result, the system token is assigned to S_A or S_B in order to get the values of these unknown inputs. A system token will be assigned to S_A if (i) C is unknown and (ii) A is unknown or if A and B are both unknown. The system token will be assigned to S_B if C is unknown and only B is unknown.

The execution of the operation will start only if there is a system token in S_C. We denote by S_C the trigger place of the operator computing C. As soon as there is a token in S_C, the two input transitions are triggered by the allocation of a system token (S) at the input places of these transitions. The values of A and B are therefore reproduced in A and B and in the output place of each of the transitions. These places contain also a system token, which will ensure the enabling of the following transition (i.e., that the two inputs are present). These two places are the input places of seven different transitions which have disjoint conditions of enabling. Only one of these transitions can be enabled and can fire. At the firing, the result, if any, is given in the result place and then in C, while the system token is assigned either to S_{next}, or to S_A, or to S_B.

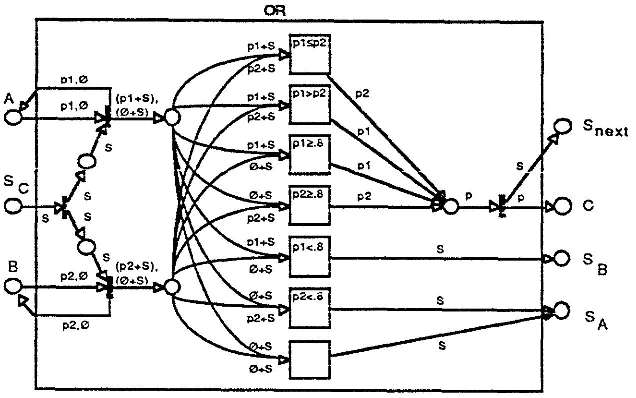

Figure 4 Model of the operator OR

These operators can be compounded in super-transitions. The model can be generalized to operators with more than two inputs by combining these basic operators.

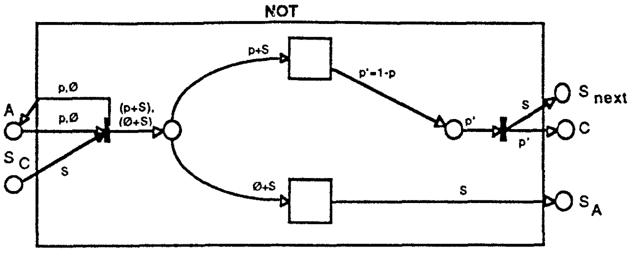

Figure 5 Model of the operator NOT

An example of the use of these logical operators is shown on the next section, where a simple inference net is modeled and the search process in this net is simulated.

1.4 Dynamic Representation of an Inference Net

The connection of the super-transitions representing the logic operators to places representing the items of evidence leads to a dynamic representation of an inference net. It allows to show explicitly how the inference engine scans the knowledge base. By running a simulation program, we can see in real time what the steps of reasoning are, the possible deadlocks, or mistakes. It allows one to identify the parts of the knowledge base where the knowledge representation is incorrect.

Let us consider the simple symbolic system containing the following rules :

if A and B => E
if C and D => D
if E or F => G

The standard representation of the inference net of this system (see section 3.1) is shown in Figure 6.

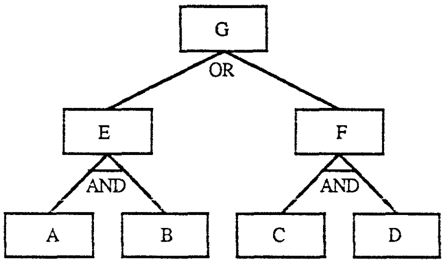

Figure 6 Standard representation of the inference net of the example.

The representation of the inference net with Predicate Transition Net is deduced from this representation by:

(1) replacing the rectangles representing the subgoals with the places of our model.

(2) replacing the formalism AND, OR, and NOT by the models of the operators aggregated in super-transitions, and linking these places to those transitions (including the self loops).

(3) linking the system places of each operator according to the rules described in section 4 for the scheduling of the checking of the unknown subgoals.

The representation of the inference net of the simple symbolic system in Figure 6, using the Predicate Transition Net models of the logic operators, is shown on Figure 7. The interface module with the user has been added through the places IA, IB, IC and ID, where the user can enter the degrees of truth of A, B, C and D.

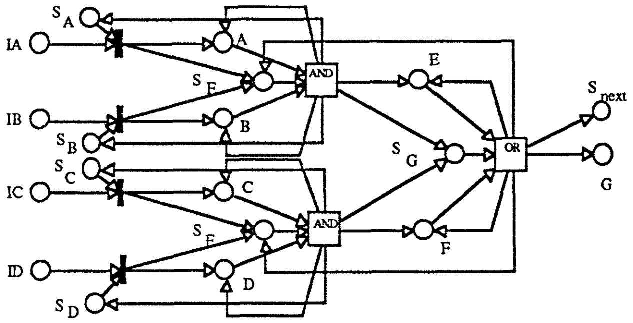

Figure 7 Inference net of a simple symbolic system, using the Predicate Transition Nets formalism

The simulation of the propagation of the tokens in this net allows one to observe the reasoning process followed by the system. The mapping of the different places of the net at each step of the process of the simulation is shown on Table 1.

TABLE 1 Mapping of the Places at the different steps of the simulation

	A	IA	B	IB	C	IC	D	ID	E	F	G	S_A	S_B	S_C	S_D	S_E	S_F	S_G	S_{next}
Step 1																		S	
Step 2																S			
Step 3												S							
Step 4		0.9										S							
Step 5	0.9															S			
Step 6	0.9												S						
Step 7	0.9			0.8									S						
Step 8	0.9		0.8													S			
Step 9	0.9		0.8						0.8									S	
Step 10	0.9		0.8						0.8		0.8								S

The search for the degree of truth of the goal G starts when the system token is put in the system place S_G, at the beginning of the search (step 1). The degree of truth of G cannot be evaluated when the operator OR is executed. The system token is therefore assigned to S_E for the checking of the subgoal E (step 2). The execution of the operator AND cannot lead to a result for E and the system token is allocated to S_A (step 3), which triggers an interaction session with the user to get the degree of truth of A. The user enters this value (say 0.9) through IA (step 4) which is assigned to A, while the system token is assigned to S_E (step 5). Since, the degree of truth of A is larger than 0.2, the result of the operator AND cannot be given in E and the system token is assigned to S_B (step 6) to get the degree of truth of B (say 0.8) through IB (step 7). The system token is then reassigned to S_E to trigger the operator AND (step 8), which can now be executed. The minimum of the degrees of truth of A and B, 0.8, is put in E, while the system token is assigned to S_G (step 9). Since the degree of truth of E is equal to 0.8, the operation OR can be performed to produce the result G equal to 0.8. The system token is allocated in S_{next} (step 10). The subgoal F has not been checked and all the part of the net which is used to evaluate F has been pruned.

1.6 Evaluation of the Response Time of an Expert System

The model allows the evaluation of the time needed to produce an output; this is then used to assess the timeliness of an organization using an expert system.

The response time of an expert system is related to the number of rules in the rule base scanned by the system to give an answer to a specific problem or goal, and to the number of interactions with the user. The model we have defined allows a quick identification of the parts of the rule base which have been scanned, given a certain set of inputs, to reach a specific goal, since each place contains the token symbolizing the value of the rule or fact it represents.

Let us consider an expert system being used to give a certain answer in a certain environment. We represent the input X_i to the system as a n-tuplet where n is the total number of questions which can be asked by the system. The answer to the questions are contained in this n-tuplet at the location corresponding to the question asked (this may not be listed in the order of appearance in time). The locations for the unasked questions are left empty. We denote by n_i the number of questions asked by the system. The number of X_i's might be very large but it is bounded. Given a certain environment, we can define a distribution $p_i(X_i)$ for the occurrence of the input X_i.

For a specific input X_i, we can identify N_i, the number of places scanned by the system to reach its goal, since they still contain the degrees of truth of the subgoals they represent. If τ is the average time to check a rule and t is the average time taken by a user to answer a question asked by the system, then the time t_i to get an answer given an input X_i will be :

$$t_i = N_i \tau + n_i t$$

Therefore, the average time of use T of the expert system for the set of inputs X_i will be given by:

$$T = E[t_i] = \sum_i p_i t_i = \sum_i p_i N_i \tau + \sum_i p_i n_i t$$

which leads to :

$$T = E[N_i] \tau + E[n_i] t$$

where E[X] denotes the expected value of the variable X.

The time T obtained is the average time needed to get an answer from the expert system. This model of a consultant expert system will be used to evaluate the effect that inconsistent information can have on the command and control process.

2.0 USE OF THE EXPERT SYSTEM FOR THE FUSION OF INCONSISTENT INFORMATION

An important problem faced by decisionmaking organization is the inconsistency of information which can degrade substantially their performance. This inconsistency can be attributed to different causes: inaccuracy in measured data, lack of sensor coverage, presence of noise, bad interpretation of data. In a military context, inconsistency of information can also be explained by the attempt by the enemy to mislead the organization about his actions through the use of decoys or jamming techniques. This presence of inconsistent information jeopardizes the successful execution of the mission of an organization.

Three strategies to fuse inconsistent information are considered in this paper : (1) ignore information sharing, (2) weighted choice among contradictory sets of data and (3) use of an expert system which has additional knowledge on the problem to be solved.

The first strategy occurs when the decisionmaker performing the information fusion uses only his own assessment and ignores the assessment of the other decision maker. This strategy is related to the way a human being assigns value to information which is transmitted to him, while executing a specific task. The study of Bushnell, et al. (1988) develops a normative-descriptive approach to quantify the processes of weighting and combining information from distributed sources under uncertainty. Their experimentation has shown that one of the human cognitive biases, which appears in the execution of a task, is the undervaluing of the communications from others, which occurs independently of the quality of the information received. The decisionmaker is, therefore, expected to have the tendency to overestimate his own assessment and to assign a lower value to the others' assessments.

The second strategy is to perform a weighted choice among the contradictory assessments which are transmitted to him and compared to his own. This weighting strategy involves the confidence which can be given to the information and which depends on the manner this information has been obtained, or on its certainty. In many models of organizations facing this problem of inconsistent information and using the weighted choice strategy, measures of certainty are the basis for the weighting of different items of evidence. Among the methods used, the Bayesian combination has given valuable results.

The third strategy involves the use of an expert system. Expert systems can consider additional knowledge and facts which would be too costly in terms of time, effort, and memory storage to be handled efficiently by the decisionmaker on his own. For each instance of contradictory data, it can check if their values are consistent with the knowledge it has and give an indication of their correctness. With this additional attribute, the decisionmaker can perform a more precise information fusion.

In order to illustrate how these strategies modify the measures of performance of an organization and to emphasize the role of an expert system in the fusion of inconsistent information, an illustrative application will be used.

2.1 Command and Control in an Air Defense problem

Mission and Organization: The illustrative application involves an organization, the mission of which is to defend a set of facilities against attacking missiles. This set of facilities consists of three cities, two military bases and two production facilities located in a square with 30 mile sides, as shown on Figure 8. To destroy incoming missiles, the organization can either use a laser beam or send an antimissile rocket. The laser beam is used in case of urgency, when the time before the missile hits its target is less than a certain threshold. The antimissile rocket is used when enough time is available. Both weapons require different targeting solutions. The performance of the organization is measured by its ability to send the right weapon at the right place for each incoming threat.

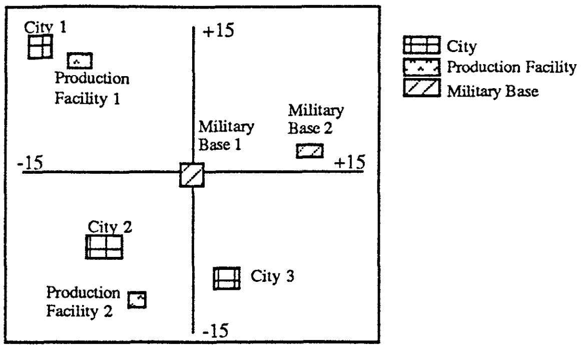

Figure 8 Location of facilities to be defended by the organization

The considered organization is a hierarchical two-decisionmakers organization with the Petri net representation (Tabak and Levis, 1984; Remy et al., 1987) shown in Figure 9. The two decisionmakers, DM1 and DM2, perform their own situation assessment producing the results Z1 and Z2. DM2 sends Z21, which is equal to Z2, to DM1 who is in charge of performing the information fusion with one of the three strategies available. One of them is to use an expert system. Using the revised situation assessment Z'1, the response Y1 is selected and transmitted to DM2. DM2 takes into account this new information in his information fusion stage and realizes the final response selection of the organization, Y.

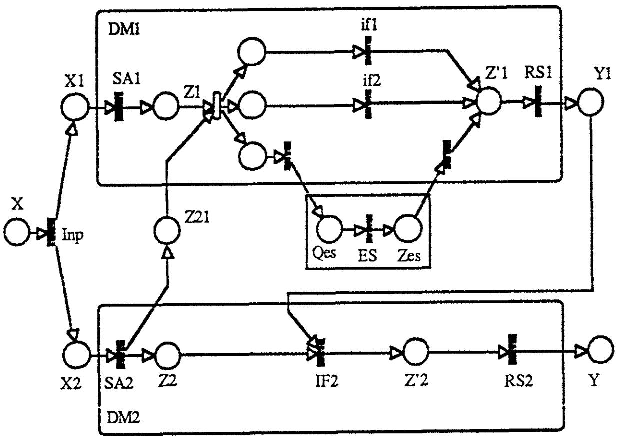

Figure 9 Petri net of the hierarchical 2-DM organization.

Inputs and situation assessments: Each decisionmaker receives as input two points of the trajectory of the missile. The first one is its position at time t, which is the same for the two decisionmakers to make sure they are assessing the same missile. The second point is determined by the tracking center of each decisionmaker. The tracking center is defined as the sum of the human and hardware means assembled to process the information. The use of decoys by the enemy and the presence of noise result in these positions being not the same for each of the decisionmakers. When this is the case, we assume that one of the two is the actual one. In addition to these different coordinates, the input contains also the confidence factors associated with each position. These confidence factors have been generated by a preprocessor (say, a tracking algorithm) and measure the quality that can be attributed to each set of data.

After receiving these inputs, the two decisionmakers, DM1 and DM2, perform the same situation assessment. DM1 (resp. DM2) computes the velocity of the missile and evaluates its impact point, according to the set of coordinates he has received, and produces the result Z1 (resp. Z2). DM2 sends Z21, which is equal to Z2, to DM1 who is in charge of performing the information fusion.

Information fusion of DM1: In his information fusion stage, DM1 makes first the comparison between Z1 and Z21. If they are equal, Z'1 = Z1 is produced. If they are different, DM1 has to choose from the three different strategies described in the previous section.

The first one is to ignore information sharing. In this case, DM1 produces Z'1 = Z1 without considering the situation assessment Z21, transmitted to him by DM2.

The second strategy is the weighting of the information according to the confidence factors associated with each set of data. DM1 considers the confidence factors Conf1 and Conf2 given with the input and which measure the quality of the information to choose Z1 or Z21. If Conf1 is greater than or equal to Conf2, DM1 produces Z'1 = Z1. In the opposite case, DM1 produces Z'1 = Z21.

The last strategy involves the use of an expert system. The simple knowledge base system which has been developed for this application evaluates the degree of threat a missile represents as a function of the distance between the location of the different facilities and its impact point estimated by the user. A more sophisticated system could make the assessment of the threat by taking into account the type of missile, the geographical aspect of the area, the direction of winds, the interest for the enemy to destroy the aimed facility, ... The threat assessment of the missile is done for the two possible trajectories, one after another. If the first threat assessment shows that the target is one of the facilities with enough certainty, the computer stops its search. In the opposite case. the computer evaluates also the threat that the missile would have if it followed the second trajectory. The answer of the expert system consists of two numbers between 0 and 1 representing the severity of the threat posed by the missile (according to each assessment). When the answer is given, DM1 does not use a strategy to make a comparison with a result from an internal algorithm, as shown by Weingaertner and Levis (1987). This is due to the fact that the decisionmaker has not enough data on his own to be able to double check the answer of the decision aid. If the degree of threat according to the assessment of DM1 is greater than or equal to the one according to the assessment of DM2, the result is Z'1 = Z1.In the opposite case, the result is Z'1 = Z21.

Response of the organization: Having chosen the trajectory which seems to be the most likely, DM1, in his response selection stage, determines the type of threat the missile represents by computing the time before impact and sends it to DM2 with the fused information. DM2, in his information fusion stage selects the weapon to use and performs the targeting solution in his response selection stage.

2.2 Measures of Performance

The measures of performance considered in this paper are workload (Boettcher and Levis, 1982; Levis, 1984), timeliness (Cothier and Levis, 1986) and accuracy (Andreadakis and Levis, 1987). They have been defined for the two possible types of interaction between the computer and the user :

The user initiated mode when the decisionmaker enters all the data he has in a specified order and the machine produces a result. Not all entered data may be needed by the machine in its search process.

The computer initiated mode when the user enters specific data only in response to requests from the computer.

Thirty three equiprobable inputs to the organization have been considered. Twenty four inputs contain inconsistent information. We assume that for half of these inconsistent inputs, the tracking center of DM1 is correct (the tracking center of DM2 is correct for the other half because we assume that for each input, one of the two contradictory positions is correct).

2.2.1 Workload

The evaluation and the analysis of workload in decisionmaking organization uses an information theoretical framework (Levis, 1984). It allows to evaluate the activity of a decisionmaker by relating, in a quantitative manner, the uncertainty in the tasks to be performed with the amount of information that must be processed to obtain certain results.

The information theoretic surrogate for the cognitive workload of a decisionmaker is computed by adding all the entropies of all the variables used to model the procedures he uses to perform his task. The distributions of all the variables are generated by executing the algorithms for all the inputs. This process of generation starts with a probability equal to zero for all the values that each variable can take. When the execution of the algorithm is performed with the input X_j having a probability p_j, the internal variable w_i , if it is active, takes the value a_i. The probability mass function of this variable w_i is updated by adding the probability p_j to the probability this variable had to take the value a_i before the execution of the algorithm with this input X_j. The operation for all the variables w_i affected by the input is:

$$p(w_i = a_i \mid X_1, ..., X_{j-1}, X_j) = p(w_i = a_i \mid X_1, ..., X_{j-1}) + p_j$$

However, to take into account the effect of the different strategies, the workload of the decisionmakers has to be computed for all the mixed strategies. A mixed strategy is a convex combination of the three pure strategies, and is noted (p_1, p_2, p_3), where p_i, $[i = 1,2,3]$ is the probability of using strategy i in the mixed strategy. The quantities p_1, p_2 and p_3 verify :

$$p_1 + p_2 + p_3 = 1$$

To compute the workload of DM1 and DM2 for all the mixed strategies, the system of all the variables has to be divided in three subsystems.

The first subsystem is composed of the internal variables of the algorithms for situation assessment of DM1 and DM2. The execution of these algorithms and the values taken by their internal variables for each input do not depend on the strategy chosen in the information fusion stage. Therefore, these algorithms are executed only once for each input to generate the probability mass functions of their internal variables. This subsystem allows to compute the invariant part of the workloads of DM1 and DM2, $G_{inv}{}^1$ and $G_{inv}{}^2$.

The second subsystem is made of the variables of the different algorithms of the information fusion stage. This subsystem has for input (Z1,Z21) and produces the output Z'1 with three different algorithms. Each algorithm i is executed independently of the others for all the inputs and the sum of the entropy of its internal variables (Z'1 is considered to be an internal variable of each algorithm) gives the activity of coordination of the algorithm of the strategy i, $g_c{}^i$. The contribution of this subsystem to the workload of DM1 is evaluated by using the Partition Law of Information (Conant,1976).

The throughput, G_t, is given by :

$$G_t = T(Z1, Z21 : Z'1)$$

The blockage term, G_b, which represents information in the input not reflected in the output, is given by :

$$G_b = H(Z1, Z21) - G_t$$

We assume that the data are noiseless and that the algorithm are deterministic. This assumption is made only to simplify the presentation. The noisy case with stochastic algorithms leads to additional terms in various expressions. In this case, the noise, G_n , is only caused by the internal choice in the decisionmaking process and is simply given by:

$$G_n = H(u)$$

where u is the decision variable specifying the choice among the different algorithms. H(u) is equal to :

$$H(u) = H(p_1) + H(p_2) + H(p_3)$$

As stated by Boettcher (1981), the coordination term is given by :

$$G_c = \sum_{i=1}^{3} (p_i\, g_c^i + \alpha_i\, \boldsymbol{H}(p_i)\ )$$

where :

$$\boldsymbol{H}(p_i) = p_i \log_2(p_i) + (1 - p_i) \log_2(1 - p_i)$$

and α_i is the number of internal variables of the algorithm i. We have therefore the activity of the subsystem, $G_{subsystem}$.

$$G_{subsystem} = H(Z1, Z2) + H(u) + \sum_{i=1}^{3} (p_i\, g_c^i + \alpha_i\, \boldsymbol{H}(p_i))$$

Finally, since the entropies of Z1 and Z21 have been evaluated in the first subsystem, the contribution $G_{if}(p_1, p_2, p_3)$ of the second subsystem to the workload of DM1 for the mixed strategy (p_1, p_2, p_3) is:

$$G_{if}(p_1, p_2, p_3) = H(u) + \sum_{i=1}^{3} (p_i\, g_c^i + \alpha_i\, \boldsymbol{H}(p_i))$$

The third subsystem is composed of the variables of the algorithms used after the information fusion stage. These algorithms are the response selection of DM1, the information fusion and the response selection of DM2. The variables of these algorithms can take three values that are different for each input according to the pure strategy used. Therefore, for each variable of this subsystem, three probability mass functions are generated for all the inputs and for each pure strategy. To compute the entropies of these variables for the mixed strategies, a convex probability mass function is deduced from the probability mass functions determined for each pure strategy. By summing these entropies, the variable contribution to the workload of DM1 and DM2 is deduced, G_{var}^1 (p_1, p_2, p_3) and G_{var}^2 (p_1, p_2, p_3). The workload of DM1 and DM2 can now be evaluated:

$$G_{DM1} = H(input) + H_{inv}^1 + H(Z21) + G_{if}(p_1,p_2,p_3) + G_{var}^1(p_1,p_2,p_3)$$

$$G_{DM2} = H(input) + H_{inv}^2 + H(Y1) + G_{var}^2(p_1,p_2,p_3)$$

2.2.2 Timeliness

The measure of timeliness considered in this application is related to the response time of the organization. A deterministic processing time has been associated with every algorithm. Again, each processing time can be described by a probability density function and the probability density function of the response time can be computed (see Andreadakis and Levis, 1987). The use of stochastic model does not add to the presentation of the example, but would be the model to use for an experimental investigation. For the strategy involving the use of the expert system, the time to give an answer has been computed using the expert system model described in the first section of the paper. The response time of the expert system is function of the number of rules scanned by the system for each input to the organization and of the number of interactions with the user. This time is likely to vary with the mode of interactions used.

We assume that DM1 and DM2 perform their situation assessment concurrently and synchronously, and that the same amount of time is needed by the two to give an answer. Therefore, only one of the two processing times is considered. T_{SA1}, (resp. T_{RS1}, T_{SA2}, T_{IF2} and T_{RS2}) denotes the time needed to execute DM1's situation assessment algorithm (resp. Dm1's response selection, DM2's situation assessment, information fusion and response selection). $T_{IF1}(i)$ is the time needed to perform the information fusion using a pure strategy i [i = 1, 2, 3]. $T_{IF1}(3)$ is a function of the average response time of the expert system computed from its response time for all the inputs. The response time for the strategy i, T(i) is therefore:

$$T(i) = T_{SA1} + T_{IF1}(i) + T_{RS1} + T_{IF2} + T_{RS2}.$$

The response time for each mixed strategy (p_1, p_2, p_3) is given by a convex weighting of the response time for each pure strategy. If T (p_1, p_2, p_3) denotes the response time of the organization when the strategy (p_1, p_2, p_3) is used, we have :

$$T(p_1, p_2, p_3) = p_1 T(1) + p_2 T(2) + p_3 T(3)$$

2.2.3 Accuracy

Accuracy of the organization has been evaluated by comparing the actual response of the organization with the desired or optimal response expected for each input. This desired response is known to the designer. A cost of one has been attributed when the incorrect type of weapon is used or when the target point is not accurate. For each input X_j having a probability $p(X_j)$, the use of the pure strategy i generates the response Y_{ij} which is compared to the desired response Y_{dj}. The cost function $C(Y_{ij}, Y_{dj})$ has the following characteristics :

$$C(Y_{ij}, Y_{dj}) = \begin{cases} 1 & \text{if } Y_{ij} \neq Y_{dj} \\ 0 & \text{if } Y_{ij} = Y_{dj} \end{cases}$$

The accuracy J(i) obtained for the pure strategy i is :

$$J(i) = \sum_j p(X_j)\, C(Y_{ij}, Y_{dj})$$

The accuracy for the mixed strategy (p_1, p_2, p_3), $J(p_1, p_2, p_3)$, is obtained by computing the convex combination of the accuracy for each pure strategy :

$$J(p_1, p_2, p_3) = p_1 J(1) + p_2 J(2) + p_3 J(3)$$

Consequently, J represents the probability that an incorrect response will be generated. The lower the value of J, the better the performance is. The next section provides an analysis of the results obtained by using these measures of performance.

3.0 RESULTS AND INTERPRETATION

Using the method described above, measures of performance have been evaluated for the three strategies. For the strategy involving the use of an expert system, we have considered two different options for dealing with uncertainty in the firing of rules, Fuzzy logic or Boolean logic; and two modes of interaction between the user and the decision aid: user initiated mode or computer initiated mode. The results are summarized in Table 2.

TABLE 2: Measures of Performance for the three strategies

	Strategy 1 Ignoring other assesment	Strategy 2 Weighted Choice	Expert System Fuzzy Logic		Expert System Boolean Logic	
			user initiated	computer initiated	user initiated	computer initiated
J prob of error	0.360	0.270	0.210	0.210	0.240	0.240
T seconds	18.240	18.960	21.015	20.850	20.974	20.364
G1 bits/symbol	63.414	64.921	70.293	70.293	65.969	65.969
G2 bits/symbol	43.920	43.847	43.240	43.240	43.287	43.287

3.1 Measures of Performance

3.1.1 Pure Strategies.

The three first columns of table 2 display the measures of performance (MOPs) of the organization for each pure strategy. These results show that the taking into account of more knowledge, either about the way data are obtained, in the case of the weighted choice strategy, or about the meaning of information, when the expert system is used, yields greater accuracy. Accuracy is an important issue for the kind of mission this type of organization is expected to carry out. The results show also that taking into account more knowledge requires the handling of more data. Therefore, more time is needed and more effort, expressed in terms of workload, is required. This increase in workload is caused more by the extra decisions which must be made when the knowledge is taken into account than by operations or manipulation done with the additional knowledge. These manipulations are done by the decision aids, out of the control of DM1.

When DM1 ignores the situation assessment of DM2, very few operations are performed. The response time is the smallest of the three. If the measure of timeliness is the ability of the organization to give a response as fast as possible, this strategy leads to a more timely response than the two others. The simplicity of the algorithm results in low workload for DM1 in comparison with the other strategies which can be explained by the fact that DM1 handles fewer variables. This strategy has low accuracy in comparison with the other strategies, because the choice made on the information to be fused is arbitrary and has no rational justification. Thus, a clear assessment of the cost and valuse of coordination can be made.

For the weighted choice strategy, no operation on variables received is performed. DM1 makes only a comparison between the weights of the information. We have assumed that the weighting process was carried out outside the organization by a preprocessor and, consequently, DM1 performs only few operations more than in the first strategy. Therefore, workload and response time are slightly larger than for the first strategy because of the extra information obtained by comparing the confidence factors. An increase of 3.9 % in response time and of 2.4 % in the workload of DM1 is found. The measure of how the data have been obtained, given through the confidence factors, brings a large gain in accuracy. An improvement of 25 % in the accuracy of the organization in comparison to the first strategy is observed. These results show, as expected, that taking into account the quality of information plays an important role in the accuracy of the organization, without degrading substantially the other measures of performance.

When the expert system is used, the increase in workload of DM1 is about 8.3 % from the level of strategy 2, and 10.8 % from the level of the first strategy. This can be explained by the handling by DM1 of the assessments given by the expert system. These assessments are variables which have greater entropies and which require more processing. The increase in response time (of 10.8 % from the level of strategy 2 and of 15.2 % from the level of strategy 1) is mainly caused by the time taken by DM1 to interact with the system and the time needed to get the answer. This response time of the expert system can get larger as the size of the knowledge base and of the magnitude of the problem to solve increase. In the example, the simplicity of the expert system hides the real effect on timeliness which can be expected with the use of such interacting system. The gain in accuracy is very significant, about 22 %, in comparison with the accuracy reached with the second strategy and 41.7 % from the level reached when the situation assessment of the other DM is ignored. This shows the extent to which the accuracy is improved when additional knowledge is used to verify the correctness of information. By using the expert system to evaluate the threat and to estimate the severity of the threat for each possible trajectory, DM1 has a broader assessment which allows him to perform more accurate information fusion.

Finally, we note that the workload of DM2 remains almost constant for all the strategies. A variation of 1.5 % can be observed. He uses always the same algorithms, and only the different distributions of the variables of the algorithms obtained, when different strategies are used by DM1, explain this small variation in his workload.

3.1.2 Mixed Strategies

The performance measures (accuracy, timeliness, and workload of DM1) reached by the organization, when mixed strategies are used by DM1 in his information fusion stage, have been obtained using CAESAR (Computer Aided Evaluation of System ARchitectures). Measures of Performance have been evaluated for all mixed strategies and have led to a surface in the space (J-T-G1) represented on Figure 10. The projections of this surface on the Accuracy - Workload (J-G1), and Timeliness - Workload (T-G1) planes are drawn on Figure 11. Measures of performance reached for each pure strategy are located at the three cusps of the figures. The convex combination of any two pure strategies gives a U-shaped curve (Boettcher and Levis, 1982) which can be explained by the fact that when a mixed strategy is used, there is an additional activity due to switching from one algorithm to another.

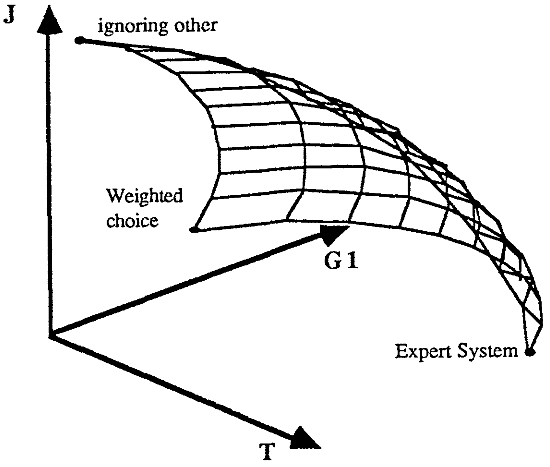

Figure 10 Locus of the Measures of Performance attained by the organization

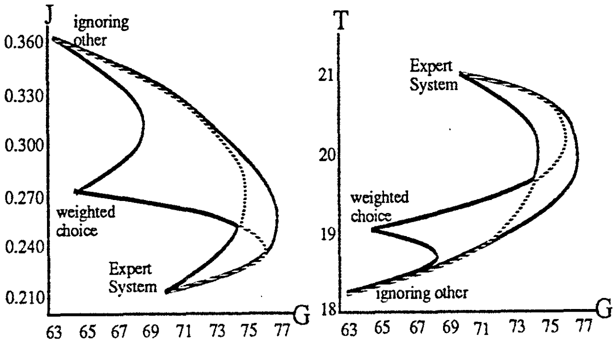

Figure 11 Mixed Strategies : Accuracy / Timeliness vs. workload for DM1

The projection of the surface of the Measures of Performance on the Accuracy - Timeliness plane (J-T) gives the triangle shown on Figure 12, which shows the performance attained by the organization. The corners of this triangle indicate the level reached in accuracy and response time for each pure strategy. For all binary variations between pure strategies or for all successive binary combinations of mixed strategies, J and T are linear combinations of each other. Figure 12 shows clearly the trade-offs between response time and accuracy and how the requirements of the mission will justify a strategy. Thus, if the requirements in accuracy are too binding, the strategy of ignoring information sharing will not be acceptable. In the same way, if the time available to process each input is too short, the expert system would be useless because too much time will be needed to perform the information fusion.

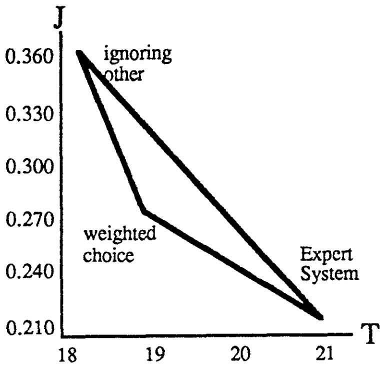

Figure 12 Mixed Strategies : Accuracy and Timeliness of the Organization

3.2 Effect of the mode of interaction

The effect of the mode of interaction on the measures of performance is shown on the last four columns of Table 2. There is no change in accuracy or workload; however, a slight change in timeliness is observed. This is caused by the fact that, in the user initiated mode of interaction, all the data which have a chance to be processed by the expert system are entered at the beginning of the session. In the example, the position of the impact points according to the two different situation assessments are entered, even if the first set is sufficient to assess the threat. Therefore, more time is needed than in the computer initiated mode where data are entered at the request of the system during the search.

It is important to note that in the air defense, no workload have been assigned to the process of entering the information in the expert system. The process consists only of replication of the information the decisionmaker already has. If the inputs asked by the expert system do not correspond to the data the decisionmaker has, he would have to perform some operations to deduce these inputs from the information he has. Let us consider an example where the decisionmaker has computed or received from another member of the organization the value of the speed of an object being analyzed. If the expert system asks the decisionmaker the question : "speed of the object : [possible answer : low, moderate, high]," the decision maker will have to deduce from the actual value of the speed the attribute asked by the system. A small algorithm will have to be executed, increasing his workload. It can be expected therefore that, in this case, a change in workload similar to the change in response time would be observed. This issue raises the problem of the adequate design of the expert system, or more generally, of the decision aid in which the mode of interaction has to be thought very carefully to avoid an unnecessary increase in the workload of the decisionmaker and in the response time.

3.3 Fuzzy Logic vs. Boolean Logic

For this illustrative application, the levels of performance reached when different expert systems are used have been studied. The performance achieved with an expert system using fuzzy logic as the means of inference, which has been developed for the example, has been compared to the performance obtained by using an expert system which does not deal with uncertainty and uses Boolean logic. This version of the expert system has been obtained by changing the mappng functions (only values 0 and 1 could be processed instead of the real numbers between 0 and 1). It has been assumed that a statement having a degree of truth greater (resp. smaller) than 0.6 was true (resp. false). Therefore, the assessment of the threat of the missile for each trajectory has only the values true or false. The different measures of performance obtained for the two systems are summarized in the last four columns of Table 2.

The organization has a response time slightly lower with an expert system using Boolean logic than with the expert system using fuzzy logic (2.3 %). This is due to the fact that by assigning the value true or false to the severity of threat, the system can reach a conclusion (which is not always the best one) by examining fewer possibilities. It can prune a larger part of the knowledge base than the fuzzy logic system when it reaches the conclusion that the missile is threatening a specific facility. When this conclusion is reached for the first possible trajectory, the other trajectory is not examined. This results in a shorter time to produce the answer and in fewer interactions with the user and therefore in a shorter response time.

Since the expert system with Boolean logic assesses the threat only with the value true or false, the answer of the expert system has a lower entropy. The workload of the decisionmaker is therefore lower (about 6.8 %) when he uses the expert system with Boolean logic than when he uses the expert system with fuzzy logic.

By pruning a larger part of the knowledge base when it reaches a conclusion, the system has more chance to make the wrong assessment of the threat. The results show that, indeed, the system with Boolean logic exhibits lower accuracy than the system with fuzzy logic. The level of accuracy is, nevertheless, better than for the two other strategies expected to be used in the information fusion stage and is explained by the fact that more knowledge is taken into account in the information fusion process.

4.0 CONCLUSION

In this paper, a model with fuzzy logic as a means for dealing with uncertainty has been developed using the Predicate Transition Net formalism. A method to make time-related measures from this representation has been introduced, taking into account the portion of the rule base scanned by the system and the number of interactions. Then, the assessment of the role of an expert system has been made through the study of an example which involves a two decisionmaker organization facing the problem of fusion of inconsistent information. The decisionmakers must identify the trajectories of threats that they then have to destroy to protect a set of facilities. In the example, the expert system helps the decisionmaker to clarify the contradictory situation assessment he has to fuse. This strategy has been compared to two others expected to be used in this situation : (1) ignoring the assessment of the other decisionmaker, (2) making a weighted choice among the two contradictory situation assessments, by taking into consideration the way the data used to produce these assessments have been obtained by each decisionmaker. Measures of performance (workload, timeliness and accuracy) have been evaluated. The results show that the use of the expert system improves significantly the accuracy of the organization, but requires more time and increases the workload of the decisionmaker using it. The comparison of the two modes of interaction between the user and the system has shown variations in workload and in response time : the computer initiated mode requires less workload and less response time for a same level of accuracy. This result tends to show that the design of an interacting decision aid must take into account not only the characteristics of the problem to be solved, but also the way the decisionmaker would use it.

REFERENCES

Andreadakis, S. K. and Levis, A. H., 1987, "Accuracy and Timeliness in Decisionmaking Organizations," *Proc. 10th IFAC World Congress*, Pergamon Press, New York.

Boettcher, K. L., 1981, *An Information Theoretic Model of the Decisionmaker*, S.M. Thesis, LIDS-TH-1096, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

Boettcher, K. L. and Levis, A. H., 1982, "On Modeling Teams of Interacting Decisionmakers with Bounded Rationality," *Automatica*, Vol. 19, No.6 (1983), pp. 703-709.

Bushnell, L. G., Serfaty, D. and Kleinmann, D. L., 1988, Team Information Processing: A normative-Descriptive Approach," *Science of Command and Control: Coping with Uncertainty*, Edited by S. E. Johnson and A. H. Levis, AFCEA International Press, Washington, D.C.

Conant, R. C., 1976, "Laws of Information Which Govern Systems," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-6, No.4, pp. 240-255.

Genrich H.J., Lautenbach, K., 1981, "System Modeling with High Level Petri Nets", *Theoretical Computer Science, 13* , pp. 109-136.

Giordona A. and Saitta L., 1985, "Modeling Production Rules by Means of Predicate Transition Networks", *Information Sciences, 35*, pp. 1-41.

Johnson, L. and Keravnov, E. T., 1985, *Expert Systems Technology, A Guide*, Abacus Press, Tunbridge Wells.

Levis, A. H., 1984, "Information Processing and Decisionmaking Organizations: A Mathematical Description," *Large Scale Systems* , Vol. 7, pp. 151-163.

Cothier P. H. and Levis A. H., "Timeliness and Measures of Effectiveness in Command and Control", *IEEE Transactions on Systems. Man and Cybernetics*, SMC-16, No 6.

Peterson, J. L., 1980, *Petri Net Theory and the Modeling of Systems* , Prentice Hall, Inc., Englewood Cliffs, NJ.

Reisig, W., 1985, *Petri Nets, An Introduction*, Springer Verlag, Berlin.

Remy P., Levis A. H., and Jin V. Y. Y., 1988, "On the Design of Distributed Organizational Structures," *Automatica*, Vol. 24, No.1, pp 81-86.

Tabak, D. and Levis, A. H., 1984, "Petri Net Representation of Decisions Models," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-15, No.6, pp. 812-818.

Weingaertner, S.T. and Levis A. H., 1987, " Evaluation of Decision Aiding in Submarine Emergency Decisionmaking," *Proc. 1987 Symposium on* C^2 *Research*, National Defense University, Fort Mc Nair, Washington DC.

Whalen, T. and Schott, B., 1983, "Issues in Fuzzy Production Systems," *International Journal of Man-Machine Studies*, Vol. 19, pp. 57-71.

Zadeh, L. A., 1965, "Fuzzy Sets," *Information and Control*, Vol. 8, pp. 338-353.

Zadeh, L. A., 1983, "The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems," *Fuzzy Sets and Systems*, Vol. 11, pp. 199-227.