[64]

Technical Report: NAVTRAEQUIPCEN 78-C-0107-1

# KNOWLEDGEABLE OPPONENT MODELS FOR ENEMY SUBMARINE TACTICS IN TRAINING SIMULATORS

Don May, Efraim Shaket, Antonio Leal
Perceptronics, Inc.
6271 Variel Avenue
Woodland Hills, California 93167

FINAL REPORT AUGUST 1978 - AUGUST 1979

JULY 1979

DoD Distribution Statement

Approved for public release; distribution unlimited.

NAVAL TRAINING EQUIPMENT CENTER
ORLANDO, FLORIDA 32813

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

| | | |
|---|---|---|
| 1. REPORT NUMBER<br>NAVTRAEQUIPCEN 78-C-0107-1 | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle)<br>Knowledgeable Opponent Models for Enemy Submarine Tactics in Training Simulators | | 5. TYPE OF REPORT & PERIOD COVERED<br>Final Technical Report<br>August 1978 - August 1979 |
| | | 6. PERFORMING ORG. REPORT NUMBER<br>PFTR-1071-79-8 |
| 7. AUTHOR(s)<br>Don May, Efraim Shaket, Antonio Leal | | 8. CONTRACT OR GRANT NUMBER(s) |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Perceptronics, Inc.<br>6271 Variel Avenue<br>Woodland Hills, CA 91367 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>Naval Training Equipment Center<br>Code N-71<br>Orlando, Florida 32813 | | 12. REPORT DATE<br>July 1979 |
| | | 13. NUMBER OF PAGES<br>75 |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) | | 15. SECURITY CLASS. (of this report)<br>Unclassified |
| | | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes four models which show promise for simulating a knowledgeable opponent for enemy submarine tactics in training simulators. While the models are primarily designed to simulate an opponent by selecting his decision alternatives, the models can also be used to simulate friendly forces as well. The four approaches are: (a) the Elicited Probability approach, (b) the Adaptive Decision Modeling approach, (c) the Heuristic Search approach, and (d) the Production Rules approach.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(20. Continued)

The Elicited Probability approach is based on a transition matrix which relates the current state vector to a set of state transformation operators. The matrix elements are conditional probabilities elicited from experts (or can be determined by collecting statistics). The state transformation operators are rules which dynamically change the state of the simulation when selected by the application of Bayesian algorithms. The basic mechanism can be used to select operators in a hierarchic manner by incorporating them in higher level transformation operators.

The Adaptive Decision approach uses pattern recognition to learn opponent behavior from instructor opponent controllers (operator). This approach is based as a pattern classifier and is used to identify biases in operator decision policy as a response to classes or patterns in the input data. The Multi-Attribute Utility (MAU) model is used to capture the decision behavior of the operator. In the MAU model, the consequences of every action are considered to be decomposable according to a single common set of attributes.

The Heuristic Search approach provides a mechanism by which the opponent responds to actions taken by friendly forces with a course of action which leads to the achievement of some enemy goal. A state space model is used to represent the problem domain. The states are a complete description of the tactical situations as they exist at a particular instant of time. An action converts one state into another. The opponent asks the question, "What sequence of actions can transform the current state into a desired goal state?" The basic search algorithm begins at a start node and expands successive nodes until a goal node is encountered. Then the path from the initial node to that goal node is the solution sought. Heuristic Search algorithms use domain specific knowledge to guide the search. Heuristic knowledge may apply to node expansion or to path evaluation. In either case heuristic knowledge is used to reduce the searching effort. Specific Heuristic Search algorithms are discussed.

The Production Rules approach uses sets of situation-action pairs, called "productions" to transform the current state to the next state. The productions represent the problem specific knowledge. In addition to productions, the Production Rule system contains a triggering mechanism that applies those that are applicable-causing the situation to change. AND/OR graphs represent human reasoning process, and can be used to answer the questions of how or why a particular conclusion was reached by the system. Also, the user can hypothesize a conclusion or desired final state and use the productions to work backward toward an enumeration of the facts that would support the hypothesis.

A set of attributes for rating each approach are defined and described. The attributes are in three general categories. Attributes related to the modeling capability of the approach, those related to the development required to use the approach in a sub simulation, and those that relate to the expected performance of a simulation system based on a given approach. These attributes are then used to rate each approach. Finally, several representative decisions are discussed and the method of application for each approach described.

# PREFACE

Simulators are employed to train military personnel in a wide range of combat-related skills, from the performance of simple procedural tasks to the execution of complex interactive missions. A primary design goal in the specification of simulator equipment is a sufficient degree of functional fidelity to allow a high degree of transfer of training to manifest itself in the later performance of the operational task.

For the training of simpler, procedural tasks an acceptable level of fidelity can be achieved by creating a simulation of the operational equipment. However, when tasks with a high cognitive component are simulated, such as those associated with tactical performance, it becomes necessary to simulate the external environment under which the operational mission is carried out.

In the context of tactics training, the most important aspect of the combat environment is the adversary. Current tactics simulators, such as the Submarine Combat Systems Trainers (21A37 series), have an adversary which is controlled by an instructor during training exercises. This approach has several shortcomings, among them: 1) the instructor is a valuable resource who should be used more effectively in other functions, such as monitoring the performance of the trainees, 2) the tactical abilities of instructors vary widely, 3) it is very difficult for an instructor to maneuver multiple adversaries, and 4) since the instructor has the advantage of knowing exactly what own ship is doing, it is difficult for him to maneuver the target(s) in a realistic fashion.

One approach to unburdening the instructor and, at the same time, creating adversary targets with a higher degree of fidelity lies in automating the maneuvering of the targets. The computer modeling of physical systems is a cornerstone of training simulation. Many of the same techniques can be applied to modeling an adversary. However, the modeling of intelligent behavior appears to be a much more complex problem.

The objective of the current study was to survey a spectrum of modeling techniques and isolate several candidates which could be applied to the problem. These candidate techniques were then further analyzed and evaluated against certain training criteria. Recommendations are made concerning each modeling approach.

Robert Ahlers
Scientific Officer

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

## LIST OF ILLUSTRATIONS

# LIST OF TABLES

# SECTION I

# INTRODUCTION

## OBJECTIVES

This final report provides a presentation and evaluation of several alternative models potentially useful as an intelligent opponent model. These models are intended to be used to simulate realistically the tactical behavior of enemy submarines within the Navy Submarine Combat System Trainers (SCST).

The objectives of the program are to:

a. Analyze the requirements of Navy submarine tactical trainers with respect to the tactical behavior of simulated enemy submarines.

b. Identify the knowledgeable opponent model algorithms and techniques applicable to submarine tactics.

c. Evaluate each model to assess its tactical maneuvering capabilities, trainability, software requirements, trainee performance measurement, and required research and development.

This report covers all these three objectives and specifically includes, with minor changes, the two quarterly reports that cover objectives (a) and (b). It goes beyond these reports in providing a detailed evaluation of each model, a compatibility analysis of each model for some of the specific decision tasks needed in the submarine combat mission, and a recommendation for an overall, best model.

## BACKGROUND

Current Navy submarine tactical simulators provide enemy submarine maneuver capability in the form of either (1) pre-determined maneuver patterns or (2) controlled tactics performed by human operators. These forms of tactical control are inadequate for modern Naval training objectives. "Canned" maneuver patterns are not responsive to friendly submarine tactics performed by the student trainee and present an unrealistic environment. Further, the student may learn the pre-determined enemy tactical patterns with continued simulator experience, thus invalidating performance measures. On the other hand, the human controller's main function is to monitor the trainee and evaluate his performance. This function permits little time to maneuver enemy submarines in response to the trainees' tactics. The problem is compounded when multiple targets are involved. Asssigning a full-time controller to each target is prohibitively expensive in terms of manpower requirements. Further, the target behavior resulting from a human controller will not exhibit the consistency necessary to train students on all types of tactical maneuvers he may encounter.

A computer-driven "knowledgeable opponent" submarine model will alleviate many of the problems inherent in pre-determined or human-driven models in the following ways (Ahlers, 1978):

a. Provides Action Feedback for Trainee's Inputs. The trainee will receive "operationally valid" feedback rather than abstract performance measures which are not presented in real time. The feedback in the form of target responses, will be displayed on the trainee's primary display. Thus, no time-sharing between task and performance displays would be necessary; full attention could be directed to the task display.

b. Provides an Optimum Model for the Trainee to Emulate. This is particularly important for individualized instruction as it allows the trainee to "discover" effective tactics.

c. Provides Infinite Variety of Tactical Configurations. Since the target will be responsive to the trainee's tactics and will be maneuvered differently as learning takes place, broad experience in unique situations will be provided.

d. Provides an Equally Matched Opponent at any Level of Trainee's Expertise. By varying the responsiveness and the appropriateness of its maneuvers, the target can be modified to remain challenging, but beatable, for a trainee at any level of proficiency. The complexity of the target could range from a straight-running target, for use in early training, to a highly sophisticated opponent with optimum sensor information for use with highly experienced approach officers.

e. Enhances Intrinsic Motivational Properties of the Training Task. Training scenarios will become true "one-on-one" contests, and the possibility of defeat will encourage the trainee to attend to the task and maintain interest in it.

f. Enhances Evaluation of the Trainee's Mastery of the Task. Certain aspects of the knowledgeable opponent model may be exploited to provide measures of the trainee's performance. For example, the length of time the opponent maintains a tactical advantage is expected to decrease as the trainee gains tactical knowledge and experience.

g. Allows Training Exercises to Reach a Legitimate Conclusion. The knowledgeable opponent will win when it achieves a significant tactical advantage. A "canned" target cannot win, it can only lose.

# SECTION II

# DECISION ENVIRONMENT

Before requirements for a knowledgeable opponent model can be identified, the decision environment for the model must be established. Since the opponent model represents the rational actions of an enemy submarine commanding officer (CO), a general description of his thought processes and decision options is necessary. Figure 1 shows, in flowchart form, some of the major decisions that an opposing submarine commanding officer must consider. This flowchart was obtained through the cooperation of the tactical instructors at the SCST facility in San Diego, California.

The first contact a submarine has with a possible enemy submarine is via acoustic sensors. These sensors are "passive" since they only listen for sounds and emit no signals of their own. "Active" sensors (sonar) emit signals and listen for their echos. When a sound source is determined to be a possible enemy submarine, a decision must be made as to its threat. If it is determined to be threatening due to its location, a decision is made to evade counter-detection, or to close and investigate with the possibility of attacking.

Once the distance between the submarines is close, it is very likely that the enemy has counter-detected, and therefore, active sensors may be used for more accurate information. Such sensors are not used early since this would immediately alert the opposing submarine. Active sensors are available in various types, and the specific one chosen depends on factors such as ocean temperature, currents, range, etc.

If the new information confirms the presence of a submarine, tactical maneuvers begin. These maneuvers are to: (1) track the opposing submarine's movements, (2) position the possible attack, and (3) prepare to evade or escape enemy attack, if necessary. If there is no war in progress, only tracking is considered. However, if a wartime situation exists, a weapon (torpedo) is launched when the range is sufficiently close. After the launching of a weapon, the submarine commander must decide whether to evade a possible counter-attack or, if the attack was unsuccessful, to attack again.

The types of situations described above are typical of the high-level decisions a submarine commander must make. Therefore, a knowledgeable opponent model should be able to choose among similar types of alternatives at the proper times. These include not only decisions concerning strategy such as evasion, sensor resources, attack methods, weapons choice, etc., but also tactical maneuvers involving course, speed, depth, etc.

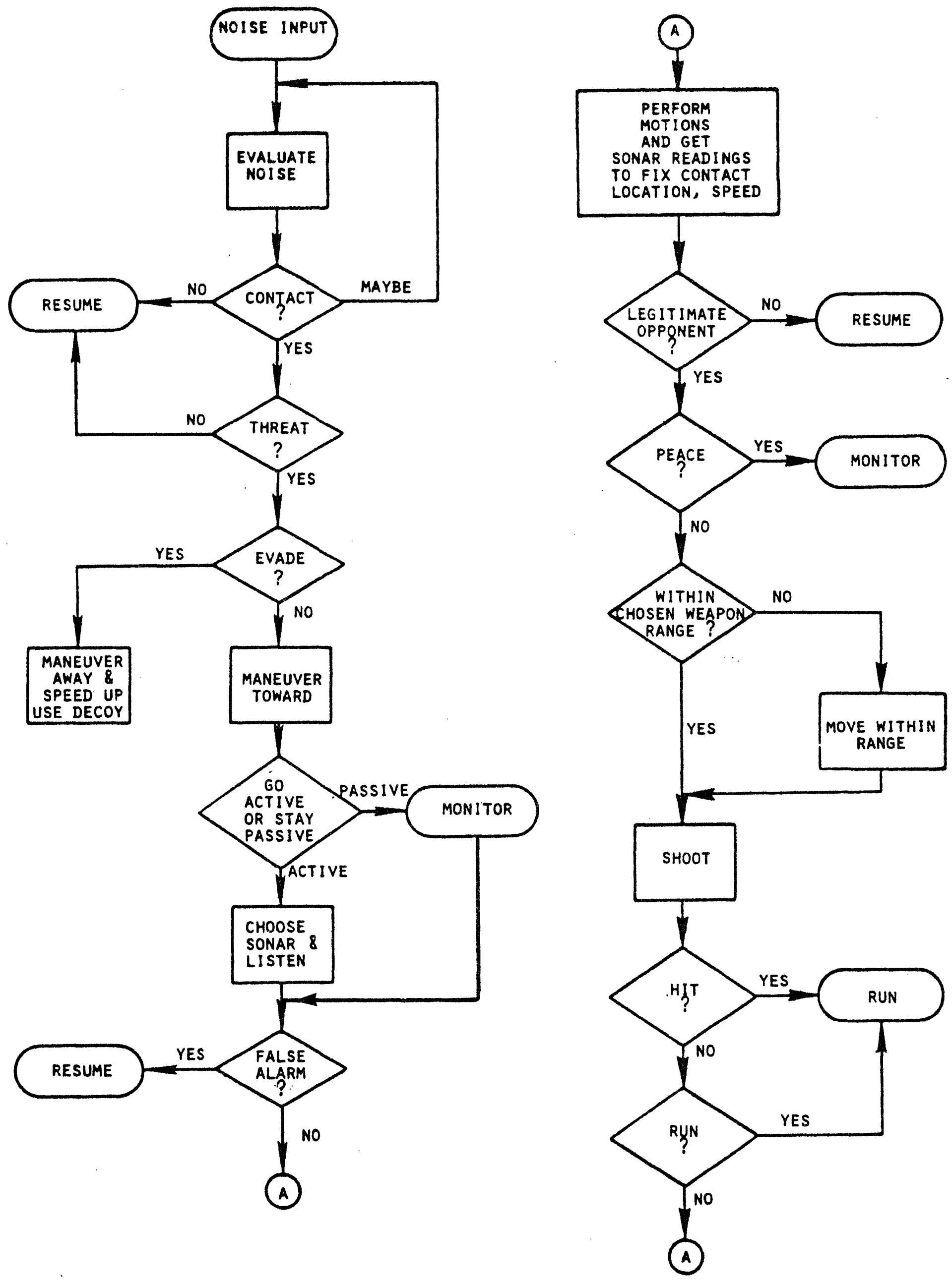



Figure 1. Decision Diagram For Submarine Commander

# SECTION III

# REQUIREMENTS ANALYSIS

## GENERAL

Requirements of the model refer to those model characteristics associated with the training objectives, training facility, and submarine behavior which are necessary for realistic training exercises. The opponent model should be compatible with the following requirements.

## MODEL STRATEGIES

General submarine strategies employed by the model will be determined by the instructional objectives. The following are three typical training objectives that would warrant different strategies:

a. Battle Stations. A wartime encounter between the friendly submarine and one or more hostile opponent submarines where torpedo attack is possible.

b. Surveillance. A wartime encounter between the friendly submarine and one or more hostile opponent submarines where information gathering, and not attacking, is the mission.

c. KILO. A peacetime encounter between a friendly submarine and a non-hostile opponent submarine where observation and tracking are the primary objectives.

These strategies determine the general behavioral characteristics of the opponent submarine which will govern and control the manuever tactics.

## PRE-CONTACT TACTICS

Pre-contact tactics are determined by the particular engagement scenario being exercised. Since pre-contact tactics do not depend on the movement or responses of the friendly submarine, they can be predefined according to established and accepted tactical doctrines and practices. Pre-contact behavior will include tactics implementing the following mission activities:

a. Barrier Patrol Search. This mission is a submarine search pattern along barriers such as coastlines, shipping lanes, known submarine routes, etc.

b. Broad-Area Patrol. Patrolling a large expanse of ocean for enemy submarines requires different tactical maneuvers, as well as specific sensor types.

c. Choke-Point Narrow Pass. Patrolling a narrow undersea pass demands different and more specialized tactics than monitoring a broad area.

d. Transient Movement. During a transient movement mission, the submarine is assumed to be traveling from one location to another for some specific purpose. Pursuing a straight line course is not the best way to avoid detection; thus, various tactical maneuvers must be simulated.

## CONTACT TACTICS

Tactics for submarine maneuvers during contact with enemy submarines must be compatible with existing tactical doctrine. The decisions to be made at each point are the course ($0^{\circ}$ - $359^{\circ}$), speed (knots), and depth (feet). The objectives which determine the values of these parameters are:

a. Manuevers to fix the location of the friendly submarine.

b. Maneuvers to gain attack position.

c. Maneuvers to evade opponent attack.

d. Maneuvers to evade contact.

The tactics doctrine that fulfills the above objectives can be found in Navy tactics manuals.

## RELATED DECISIONS

Many decisions not directly connected with tactical maneuvers are vital to a complete model. The three parameters described in the previous section are enough to specify particular maneuver tactics. However, many other related decisions must be made. The model must be able to make the following decisions at the proper time during the simulation. The model must decide:

a. The probability of a contact based on passive sensors.

b. Whether a contact represents a possible threat.

c. Whether to approach the contact or evade.

d. Whether to stay passive or use active sonar.

e. Which weapon to fire and when.

f. Whether or not the submarine is within the weapon range.

g. Whether or not the opponent ship has fired a weapon.

h. Whether or not to use decoys.

i. Whether to run or hide in deep water while evading contact.

FEATURES NOT INCLUDED

The knowledgeable opponent model will not be required to support the following simulation features:

a. Surface Ship and Periscope Contact. Since almost all training exercises deal with submarine-to-submarine encounters, the SCST instructors felt that simulation of either surface ship contact or periscope contact should not be necessary.

b. Sonar and Acoustic Equipment Performance Variations. During simulation exercises, the performance of the acoustic equipment aboard the friendly submarines is sometimes degraded for training purposes. It will not be necessary for the model to operate under such conditions.

d. Multiple Submarine Strategies. Since the radio silence will usually be maintained between submarines during wartime, coordinated strategies are not a necessary requirement. Each enemy submarine can possess its own independent knowledge opponent model with provisions only for collision avoidance and mutual attack avoidance.

TRAINING REQUIREMENTS

The model must be compatible with current SCST training objectives. Independent of the specific features of the model are considerations and characteristics that are required for the training objectives of the SCST to be met.

a. Training Management. The model must perform adequately enough so that the training instructors will actually be relieved of their responsibilities for scenario management.

b. Model Override. The instructors must be able to take control of the opponent submarine at any time and maneuver it as they are currently able to do.

c. Performance Measurement. The tactics and behavior of the enemy submarine must be conducive to the collection of meaningful student performance evaluation data.

d. Modification Ease. The model must be designed so that tactical and behavioral changes are not only easy to make but can also be made in real time by the instructors during a simulation exercise.

e. Real-World Fidelity. Real-world fidelity should be maintained as much as possible. This requirement was considered to be more important than fidelity to training objectives by the interviewed submarine trainer instructors. The apparent reason for this preference is that the training objectives are under the control of the training facility and can be modified easily. However, if the real-world fidelity is sacrificed for training objectives, modification is considerably more difficult. It is not clear that real-world fidelity and training objectives fidelity are incompatible; however, priority should be given to making the model as close to actual circumstances as possible.

# SECTION IV

# MODELS DESCRIPTION

## GENERAL

This chapter presents an overview and some details of four types of decision models which are potentially appropriate for simulating an intelligent opponent within the SCSTs. Considering variations of each model, and the possibility of combining models, many useful combinations can be derived to represent the intelligent opponent.

Since the opponent and friend have essentially the same decision structure, the same model which is developed for the opponent can also model the friend. This brings up a number of interesting and useful possibilities:

a. Play one model against the other. By doing this, it will be easier to debug the software. Also, it is possible to develop a set of performance baselines which can be used for further model development and to develop evaluation guidelines.

b. The opponent model easily contains a model of the friend. Further levels of recursion are possible. For example, the friend can be aided by an opponent model which contains a friend model.

c. Different models can play each other to evaluate which model is best.

d. Different parameter values can be set for each model and the models can play each other in order to evaluate the effectiveness of various strageties and various assumptions regarding opponent capabilities.

It should be emphasized that when the same model is used for several purposes, different behavior can be created by varying model parameters, even the same model will display different behavior patterns in slightly different circumstances. Furthermore, some of the model behavior will be generated randomly (e.g., the specifics of an evasion maneuver), thus defying the student from capturing a standard response.

## POTENTIAL MODELS

From an analysis of the requirements of the knowledgeable opponent model and from an analysis of existing simulation and modeling techniques, four major approaches have been identified which show potential for model implementation. These approaches are:

a. Elicited probability approach.

b. Adaptive decision modeling approach.

c. Heuristic search approach.

d. Production rules approach.

The elicited probability approach to scenario generation is a derivative of the Bayesian analysis. It essentially selects randomly among the alternative actions available at each point, but the probability of selecting each alternative is elicited from experts to resemble actual behavior.

The adaptive decision model is based on the adaptive linear pattern recognizer. The model "learns" the proper choices it has to make by following those made by an expert--a trainer. It then uses the trained parameters to make the right choices even in situations which are dissimilar to those under which it was originally trained.

In the heuristic search approach, the problem domain is represented as a network of "states" each representing a specific tactical situation. The objective of the CO is to reach some desired goal (mission), which is also a state in the "state space."

From the state he is in, the CO will perform a "Look ahead" search to identify which alternative action open to him will bring him closer to the goal state. This goal directed behavior is continued even if the state is changed by external events or actions of the adversary, thus depicting intelligent behavior.

In the production rule approach, the expertise of the problem domain is represented as "condition--action" chunks. A control mechanism activates the relevant productions and generates a chain of actions that would lead from the current situation to the desired goal.

It is clear that these models are quite different from each other. The rest of this chapter will describe them in detail and specify the advantages and disadvantages of each for our purpose--the modelling of an intelligent opponent.

## THE ELICITED PROBABILITY APPROACH

INTRODUCTION. The elicited probability approach to scenario generation and opponent simulation uses an incremental, descrete description of the tactical scenario. This description has the form of a state vector $\overline{Z}^t$. The vector is made up of components each representing the state of some tactical aspect of the situation at a given instant t, thus:

$$\overline{Z}^t = [Z_1^t, Z_2^t, \ldots, Z_n^t] \qquad (1)$$

In the tactical submarine simulation the components of the state vector may be:

$Z_1$ = "How deep is the water" (2)

$Z_2$ = "How far is friend"

$Z_3$ = "How many friend's subs are in the area"

The value of a component of the state vector is one of the possible answers to these questions. Thus $Z_1$ can be, at a given time t either "deep," "medium" or "shallow." The value of $Z_2$ may be either "undetected," "far," "within passive listening range," "within active sonar range" or "within torpedo range." The composition of the state vector is determined by elicitation from experts. The number of discrete values which each component can assume need not be large, it is only determined by what makes a tactical difference. If the tactics of the simulated opponent would be different in "shallow" waters than that in "medium" or in "deep" waters, then only these three discrete values are needed in the tactical simulation. Other components of the state vector may have more or less numerous discrete values, again depending on how many are relevant tactically. These discrete values are used in the intelligent part of the simulation--the part that chooses and changes tactical maneuvers. The part that generates the actual display is incremental and thus can generate continuous motions.

Figure 2 depicts the basic operation and main blocks of the simulation system. The system goes repeatedly through the following cycle; it starts from the current state vector $\overline{Z}^t$ and calculates the state of the world at the next time interval $\overline{Z}^{t+1}$. The calculation is done in two steps. First, a probability matrix is used to determine, from the current state of the world, what are the tactics that should be performed. Then, the tactics chosen are used to transform the current state vector to the vector of the next time interval $\overline{Z}^{t+1}$. This new vector might include an incremental change in location: $\nabla X$, $\nabla Y$, a change in direction: $\nabla\Theta$, or a firing of a torpedo which is another component of the state vector. The same new vector is now used also to generate the new outputs that will produce the new display for the user (interfaces with the current system).

The new value of the state vector $\overline{Z}^{t+1}$ is now fed back to the starting point where it is used as the current state vector for the next time interval. Thus, the total process progresses cyclically through this sequence of steps.

UPDATING THE STATE VECTOR. The actual calculation of the changes of the state vector is somewhat more complex than what was described above. The complexity is necessary to provide some randomness in the simulated behavior to prevent the trainee from learning a prerecorded scenario. The randomness is generated from probability information elicited from experts, and thus the behavior produced would be typical and similar to an opponent commander behavior but would still be unpredictable in its details.

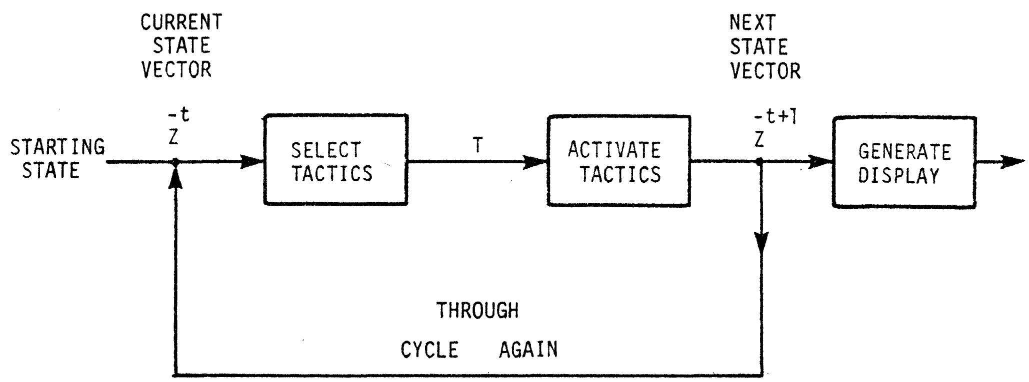



Figure 2. The State Transformation System

Figure 3 shows in more detail the specific steps that are taken in calculating the state vector at time t+1 from $\overline{Z}$ at time t. The current state vector $\overline{Z}^t$ is used to select, by combining conditional probabilities, the tactics to be applied. Let us define the terms more precisely before using them. Let us call the vector of all the tactics that are performed at time t by:

$$\overline{T}^t = (T_1, T_2, \ldots, T_m) \tag{3}$$

$T_1$ might be "turn right 10°," $T_2$ might be "decrease depth to periscope level." More than one activity can take place at the same time, so that the vector format is needed to combine the effect of all in each time interval. To determine which tactics should be selected in the next interval a conditional probability is needed: $P(\overline{T}^t|\overline{Z}^t)$. This conditional probability answers the following question: given the current situation $\overline{Z}^t$ what tactics should be applied--$\overline{T}^t$. $\overline{Z}^t$ and $\overline{T}^t$ are vectors with many elements and that makes the conditional probability a matrix of the form:

$$P(\overline{T}^{t+1}|\overline{Z}^t) = \left\{p(T_i^{t+1}|Z_j^t)\right\}_{i,j} \tag{4}$$

In every row i, which corresponds to a tactics $T_i$, the entries indicate the conditional probability of selecting these tactics given that the $Z_j^t$ component of the state vector is present. For instance, one entry might be the answer to: What is the probability, given that friend is "in torpedo range" that the tactics "shoot a torpedo" be applied. There are two problems with this approach. One is the independence of the state vector elements, i.e., whether the conditional probability of a tactics $T_i^t$ given $Z_j^t$ is independent of the other components of $\overline{Z}^t$. The other problem is meaningfulness to the expert. For example, a question like: What is the probability of choosing a "zigzag maneuver to the right" given "enemy sub is nuclear?" Posing the question the other way around should prove much more meaningful: Given a tactics $T_i$ what set of events would cause you to choose it? The natural question to an expert is the conditional probability matrix:

$$P(\overline{Z}^t|\overline{T}^{t+1}) = \left\{P(Z_j^t|T_i^{t+1})\right\}_{i,j} \tag{5}$$

This matrix of probabilities is obtained from experts in submarine tactics. The expert estimates can be based upon experience, upon real world measurements, upon theoretical models, etc. It is also possible to determine the conditional probabilities by collecting statistics during an actual training session in which the instructors are controlling opponent actions.

To calculate the conditional probability in (4) from the estimated conditional probability given in (5) the following formula has to be used:

$$P(T_j|Z^t) = \frac{P(T_j)P(\overline{Z}^t|T_j)}{P(\overline{Z}^t)} \tag{6}$$

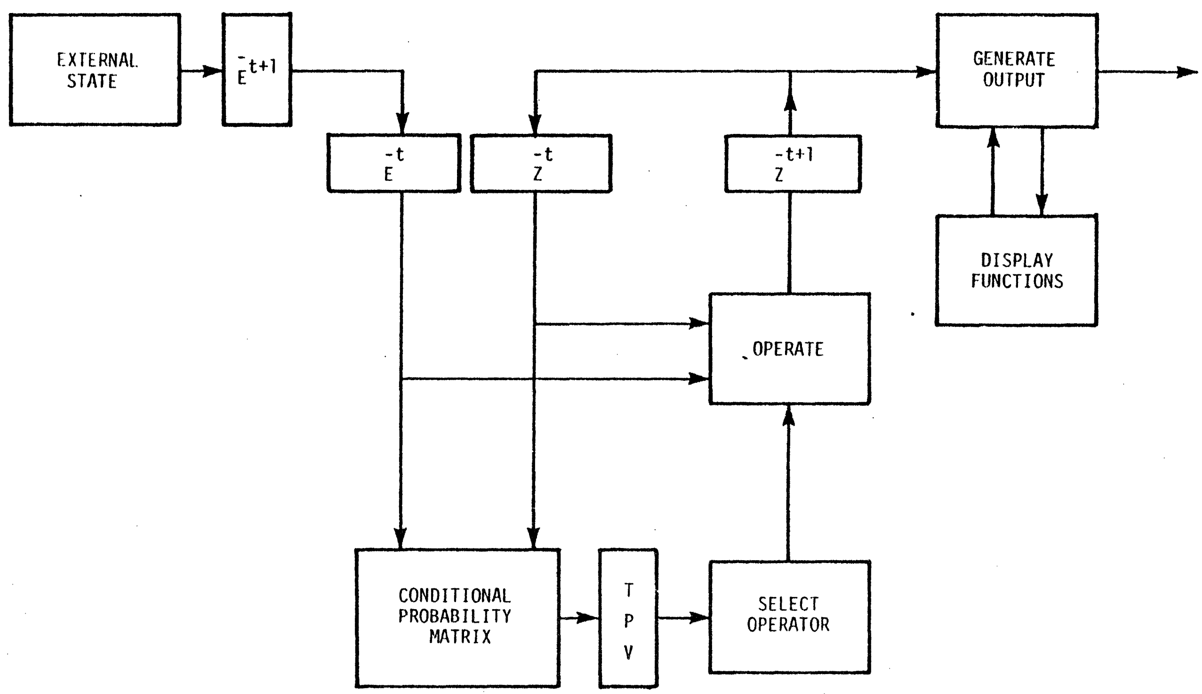



Figure 3. Detailed System Block Diagram

This formula, basic in Bayes probability theory, combines the conditional probabilities $P(Z^t|T_j)$ to give $P(T_j|Z^t)$.

Two additional vectors of _a priori_ probabilities, also estimated by experts, are required. The components of the first vector, $\overline{P}_T$, are the _a priori_ probabilities that each state transformation operator will be selected. They are represented thusly:

$$\overline{P}_{oT} = [p_o(T_1), p_o(T_2), \ldots, p_o(T_m)] \tag{7}$$

The components of the second vector, $\overline{P}_Z$, are the _a priori_ probabilities of the occurrence of each state component of the $\overline{Z}$ vector. They are represented as follows:

$$\overline{P}_{oZ} = [p_o(z_1), p_o(z_2), \ldots, p_o(z_n)] \tag{8}$$

The _a priori_ probabilities don't have to be estimated with great precision because, as the scenario unfolds, they have less and less effect over the behavior of the scenario.

If we assume independence of the impact of the different components of the state vector then:

$$P(\overline{Z}^t|T_j) = \prod_{i=1}^{n} p(z_i^t|T_j) \tag{9}$$

Thus, equation (6) becomes

$$p(T_j|\overline{Z}^t) = \frac{p(T_j) \prod_i p(z_i^t|T_j)}{\prod p(z_i^t)} \tag{10}$$

When equation (10) is implemented, the $p(T_j|\overline{Z}^t)$ are normalized; thus, the denominator in (10) is not needed.

Table 1 is a partial example of the probabilities as they are elicited from the experts and after they are used in formula 10 to obtain the conditional probabilities $P(T_j^{t+1}|Z_i^t)$. The left most column shows the components of the state vector and the values that they can assume. The list of useful tactics are indicated on the top. The first column of numbers and the first row indicate the _a priori_ probability of each state vector component value and each tactics. The body of the table contains the conditional probabilities. Looking at the second row of numbers, the probability of friend being undetected is 0.9 if the tactics is "proceed" but it is 0.0 if the tactics is "run." This makes sense; because if friend is undetected, there is no reason to choose "run." Naturally, each row sums up to 1 because if that particular state variable is present, it must have some succeeding action, even if it is only "proceed."

The assumption that the variables which comprise the state vector are independent is a crucial one. The most practical way to meet this condition is to take care to define the state vector such that it is

## TABLE 1. MANEUVER SELECTION MATRIX

| State | | | Hide | Run | Proceed | Zigzag | Attack |
|---|---|---|---|---|---|---|---|
| | | A Priori Probability | 0.05 | 0.1 | 0.7 | 0.05 | 0.1 |
| How Far Is Friend: | Undetected | 0.60 | 0.05 | 0.0 | 0.9 | 0.05 | 0.0 |
| | Very Near | 0.15 | 0.1 | 0.1 | 0.1 | 0.3 | 0.4 |
| | Near | 0.15 | 0.2 | 0.15 | 0.05 | 0.3 | 0.3 |
| | Medium | 0.05 | 0.3 | 0.20 | 0.15 | 0.15 | 0.2 |
| | Far | 0.05 | 0.25 | 0.20 | 0.25 | 0.25 | 0.1 |
| Has Friend Detected: | Yes | 0.15 | 0.2 | 0.1 | 0.1 | 0.3 | 0.3 |
| | Possible | 0.15 | 0.20 | 0.20 | 0.2 | 0.25 | 0.15 |
| | No | 0.70 | 0.10 | 0.15 | 0.45 | 0.15 | 0.15 |
| War State: | War | 0.01 | 0.1 | 0.1 | 0.5 | 0.2 | 0.1 |
| | Peace | 0.99 | 0.1 | 0.1 | 0.7 | 0.1 | 0.0 |
| Water Depth: | Shallow | 0.1 | 0.3 | 0.1 | 0.3 | 0.1 | 0.2 |
| | Normal | 0.5 | 0.1 | 0.2 | 0.3 | 0.2 | 0.2 |
| | Deep | 0.4 | 0.1 | 0.25 | 0.3 | 0.2 | 0.15 |

independent. If there are dependencies in the state vector, they may not noticeably affect the behavior of the scenario (e.g., environment, opponent's actions). This can be tested by using the model to generate behavior which is viewed by the person from whom the probabilities were elicited. If the behavior is not as desired, the elicited probability values can be fine-tuned until the proper behavior is obtained.

One technique of handling dependencies in the state vector is to also elicit the covariance matrix representing the correlation among state variables. This matrix can then be used in one of two methods:

a. The problem is transformed into a domain where independence holds (by proper selection of independent tactically significant state vector components).

b. The covariance matrices are used to derive weights to compensate for dependence.

Both methods have several disadvantages:

a. The covariance matrices are dependent on the order of processing state variables; a different covariance matrix must be used for each order.

b. The covariance matrices involve either asking people to estimate means and standard deviations, or polling a group of experts and collecting these statistics.

c. When the probabilities are subjectively determined (by elicitation), the precision of the problem is such that the covariance matrices may be meaningless.

In general, the complexity of using the covariance matrices seems to exceed that justified by meaning and relevance.

Another method of handling dependencies in the state vector is to construct a new set of variables based on permutations of some of the dependent variables. This approach is simple, but leads to a rapid increase in the size of the state vector.

Going back to Figure 3, formula 10 is used to obtain, from the current state vector, the tactic's probability vector (TPV):

$$P(\overline{T}^{t+1}) = [P(T_1^{t+1}), P(T_2^{t+1}), \ldots, P(T_m^{t+1})] \qquad (11)$$

This vector indicates the probability of selecting tactics $T_j$ in the current tactical situation. The next step is to select the tactics to be actually applied. This can be done in several ways:

a. Select the tactics with the highest probability.

b. Select all the tactics with probability higher than some threshold level.

c. Select the tactics randomly but in such a way that the probability to select a particular tactics is proportional to its $P(T_j^{t+1})$. (This is the "Monte Carlo" method.)

After the tactic (or combination of tactics) is selected, the next step is to actually perform the tactics. In terms of the model, we will apply a transformation $T^{i+1}$ to the current state vector to obtain the new one:

$$\overline{Z}^{t+1} = \left[T^{t+1}\right] \overline{Z}^{t} \qquad (\left[T\right] \text{ a matrix}) \qquad (12)$$

There are virtually no restrictions on the kinds of state transformation operators which can be defined. A transformation operator may affect a single state variable and generate a constant output. It may also affect a large number of state variables and make use of a complex decision strategy to determine their values. The transformation operator may even determine the value of a variable for several subsequent time cycles.

A transformation operator may make use of subsets of $\overline{Z}^t$ which were not used in selecting the operator. An operator may also make internal use of Bayesian aggregations based upon additional conditional probability matrices and subsets of $\overline{Z}^t$. Thus, hierarchies of transformation operators can be established.

Each transformation operator affects a set of one or more state variables. The operators, in turn, are grouped according to which set of variables they affect. These sets of variables must be disjoint because, after a single operator is selected from each set, the selected operators are assumed to be invoked simultaneously. If the sets of variables are not disjoint, the order in which the selected operators are actually invoked will affect the value of the transformed state vector. However, non-disjoint sets of variables can be handled by establishing a hierarchy of operators within a "higher level" operator.

The selection of one state transformation operator from each operator set is made by means of a Monte Carlo selection procedure. The probabilities of occurrence of each operator in the set are normalized to obtain a discrete cumulative distribution function. A uniformly distributed pseudorandom number in the range [0,1] is then generated and its position in the distribution function is used to select the operator. Alternatively, the operator with the highest probability could be selected.

In some experimental applications, it may be useful or necessary to know the probability that a state variable will have a particular value, $p(z_k^{t+1}|Z^t)$. By restricting the kinds of allowable state transformation operators to those that generate a constant (and unique) result, it is possible to obtain these probabilities directly from the scenario generator. If state transformation operator, $T_j$, outputs the same value for $z_k^{t+1}$ whenever it is invoked, and only $T_j$ outputs that value, then

$$p(z_k^{t+1}|\overline{Z}^t) = p(T_j|\overline{Z}^t) \tag{13}$$

If more complex transformation operators are used, $p(z^{t+1}|\overline{Z}^t)$ becomes more difficult to compute. A value can always be obtained, however, by making statistical measurements of the behavior of the scenario generator.

The current state vector, $\overline{Z}^t$, is transformed into $\overline{Z}^{t+1}$ by the (assumed) simultaneous invocation of all of the selected state transformation operators. If the state vector is properly designed, it is possible to use the Bayesian/Monte Carlo selection mechanism to choose all of these operators. However, in many instances it may be more convenient to use "external" mechanisms to select transformation operators for certain subsets of the state vector. These externally controlled state vector subsets will be collectively referred to as the $\overline{E}^t$ subvector (see Figure 3). Examples of externally controlled state variables would include clock-driven variables such as day and night, high and low tides, and events which occur on a fixed schedule.

PROBABILITY ELICITATION. Previous research has shown that human experts are good at estimating conditional probabilities, but poor at aggregating them (e.g., Edwards, 1962). Accordingly, the present scenario generator uses conditional probabilities elicited from experts and aggregates them automatically. First, expert inputs are used to:

a. Describe the environment to be modeled in terms of relevant state variables.

b. Determine which variables are externally controlled and which are controlled by the Bayesian model.

c. Define all of the transformations which change the state variables.

Then, the expert is queried in detail to:

d. Estimate the <u>a priori</u> probabilities and the individual conditional probability which consitute the entire matrix.

The method of elicitation is simply to interview the expert and ask him the probabilities. Bond and Rigney (1966) were able to elicit almost 650 conditional probabilities associated with electronic troubleshooting in one hour using a simple questionnaire.

The process of probability elicitation is an iterative one which allows the expert to refine his estimates. That is, once the initial estimates are made, test scenarios are generated which allow the expert to see the consequences of his estimates. He is then asked to modify his estimates to make them more consistent with the desired behavior of the scenario generator.

ELICITED PROBABILITY APPROACH - SUMMARY.

Advantages

a. Simplicity; easy to develop, maintain, implement.

b. Generates a probabilistic opponent and environment.

c. Weights representing behavior are easy to elicit and to alter.

d. State oriented; easy to switch between manual and automatic operation.

Disadvantages

a. It is difficult to alter structural aspects due to the need to avoid dependencies in the state vector.

b. Difficult to insert logical statements to control the scenario.

c. The application of state transformation operators may be order dependent.

d. It is difficult to isolate the particular entry in the transformation matrix that caused some behavior and to give it a tactical interpretation.

THE ADAPTIVE DECISION MODELING APPROACH

INTRODUCTION. The adaptive decision approach to generating knowledgeable opponent behavior--which uses pattern recognition--is based on learning opponent decision modeling and utility theory. In the present application, all of the relevant information for selecting the opponent's next action is immediately available at the time it's needed. The model, which is first adapted to choices made by an expert, is then used to calculate the value of each alternative, and the alternative with the highest value is chosen for actual execution by the system.

ADAPTIVE DECISION MODELING. Work on adaptive decision-making is derived from the areas of behavioral decision research and AI experience with learning networks. The unique aspect of this approach is the capability to adjust model parameters on-line and change decision strategy accordingly. In essence, the learning system attempts to identify the decision process of the human operation on-line by (a) successive observation of his actions, and (b) establishment of an interim relationship between the input data set and the output decision (the model). Learning in this context refers to a training process for adjusting model parameters according to a criteria function. The object is to improve model performance as a function of experience, or to match the model characteristics to that of the operator.

Learning techniques have been used to model the decision strategy and to identify the sources of cognitive constraints on the human operator performing a dynamic prediction task (Rouse, 1972). Another example of an adaptive model of the human operator through real time parameter tracing has been reported by Gilstad and Fu (1970). Linear and piecewise-linear discriminant functions were used to classify system gains, errors and error rate. The decision boundaries for classification were determined through a process on on-line learning, observing operator performance and parameter adjustment. The specific model used was applicable only to very limited tasks, and merely illustrated the feasibility of the technique.

A unique advantage of using a learning system lies in its capability to act as a pattern classification mechanism. As such, it can be used to identify biases in operator decision policy as a response to classes or patterns in the input data (Tversky, et al, 1972). In conventional Bayesian technique, the pattern of events is decomposed into elementary data points. With the assumption of independence, the elementary data points are aggregated to revise the hypothesis. Effects of the data pattern do not bear on the decision.

In dynamic decision making, however, the temporal and spatial nature of the data are highly significant. Since decision data appear as a pattern of individual events, it is reasonable to assume that the subject responds to the pattern as well as to the individual value. In fact, the pattern may contain the greater amount of information. Classification of input patterns by the learning mechanism can be accomplished by programmed cognizance of such data features as: data with non-independent events, data with correlated events, data with events which continuously vary with time, the number of elements of decision data and the rate of change in the data points.

THE MAU MODEL. Multi-attribute decision analysis is the most widely used approach for making evaluations involving multiple criteria. MAU methods decompose the complex overall evaluation problem into more manageable sub-problems of scaling, weighting, and combining criteria. In doing so, the MAU methods provide a rich framework for analysis, discussion, and feedback. This "divide and conquer" approach to evaluation involves defining the problem, identifying relevant dimensions of value, scaling and weighting the dimensions, and finally aggregating the dimensions into a single figure of merit for the system.

The power of the multi-attribute approach lies in its level of analysis and flexibility. Sensitivity analyses of the level and weight of each dimension can provide indications of what aspects to concentrate tests on, or what system elements to modify. Flexibility is present, since criteria can be added or deleted as necessary. Also, the weights and levels can be quickly adjusted according to new functional requirements and capabilities.

In the MAU model, the consequences of every action are considered to be decomposable according to a single common set of attributers. The model computes an aggregate multi-attribute utility (MAU) as a weighted sum of each consequence attribute level ($A_i$) multiplied by the importance or utility of the attribute ($W_i$). The calculated MAU of each action is used as the selection criterion:

$$MAU_j = \sum_i W_i A_{ik} \tag{14}$$

where

$MAU_j$ = the aggregate utility of option j

$W_i$ = the importance weight of attribute i, and

$A_{ik}$ = the level of attribute i for action k.

Figure 4 shows the major components of the MAU model in block diagram form. Possible actions are parameterized in terms of attribute levels. The MAU calculator uses as inputs (1) the attribute levels of the given action, and (2) a vector of "attribute weights" which have been dynamically estimated for a given operator by an adaptive model.

Calculation of the multi-attribute utility for each action is central to the operation of the model. The MAU calculation is shown in Figure 5. The dot-product of the attribute level vector and the attribute weight vector provides the aggregate MAU value. The attributes are scaled so that each attribute level ranges from 0 to 1. Further, the orientation is arranged such that each attribute contributes positively to the overall aggregate MAU. That is, holding all other attribute levels constant, an increase in any attribute level increases the MAU.

ATTRIBUTE CHOICE. The determination of attributes to include in the decision model is probably of greater importance than the accurate assessment of the importance weights (Dawes, 1975). The following list of desirable characteristics for the attributes expands on Raiffa's (1969) recommendations of attribute independence, set completeness, and minimum dimensionality:

a. Accessible. The levels of each factor should be easily and accurately measurable.

b. Conditionally Monotonic. The factor level should be monotonic with the criterion (preference) regardless of the constant values of other factors.

c. Value Independent. The level of one attribute should not depend on the levels of the other attributes. This is to some extent a consequence of recommendation b.

d. Complete. The set of attributes should present the operator's behavior as completely as possible.

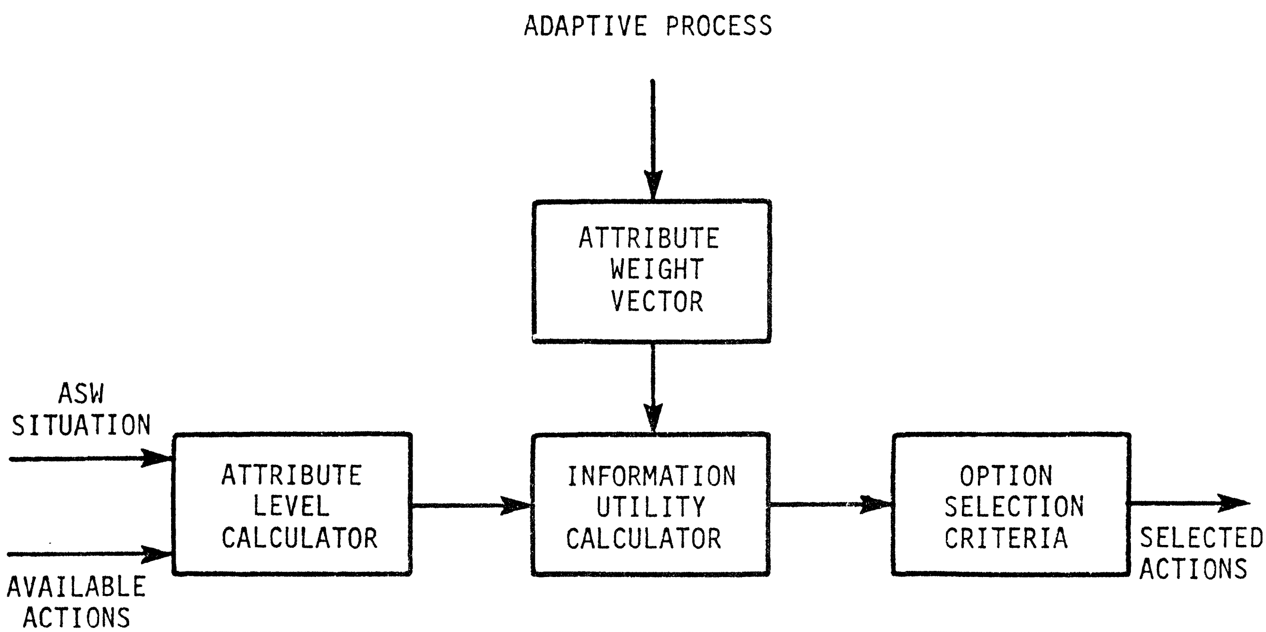



Figure 4. Overview of Action Selection Model

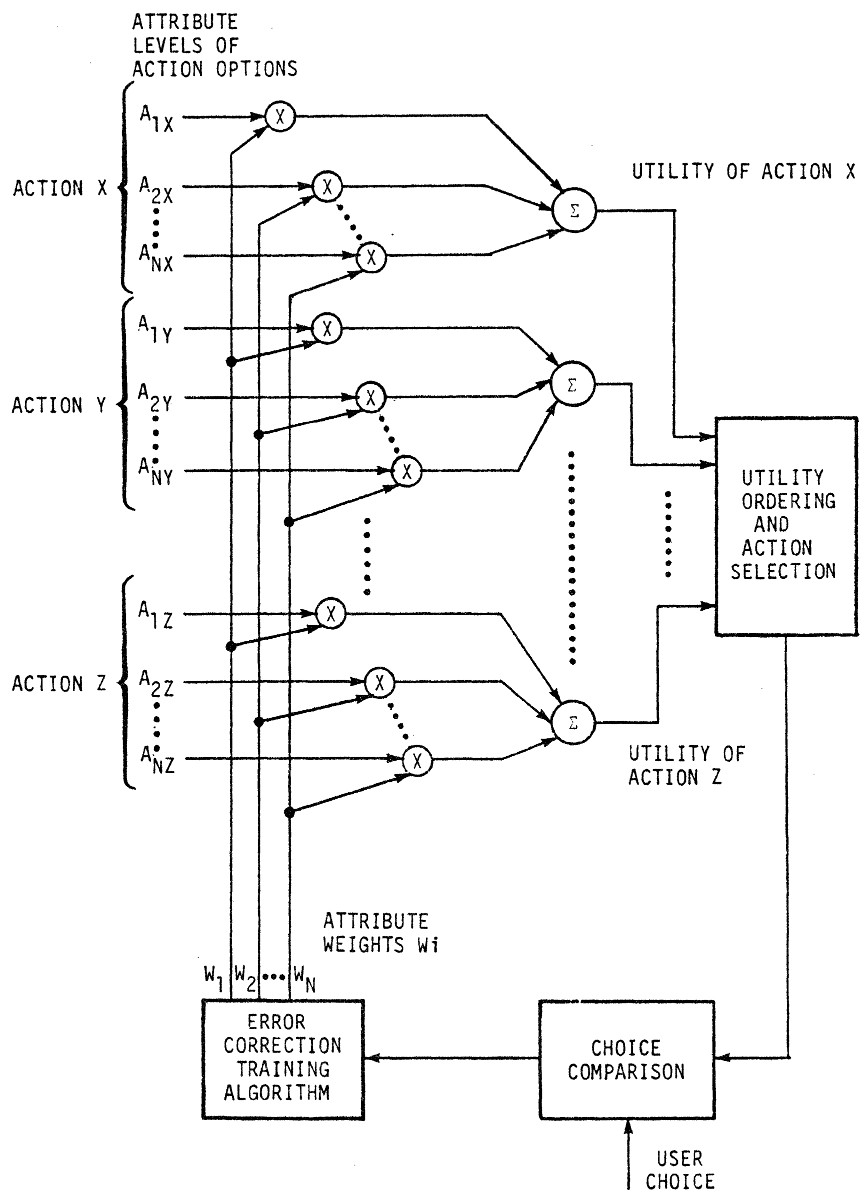



Figure 5. Adaptive Multi-Attribute Decision Mechanism

e. Meaningful. The attributes should be reliable and should demonstrate construct validity. Feedback based on the model attributes should be understandable to the operator.

For the most part, these recommendations result in an attribute set that is measurable, predictive, and in accord with the axioms of utility theory. The recommendations also imply a limitation on the number of possible attributes. The requirements of independence and meaningfulness render any large set of attributes unrealizable, because of the cognitive limitations of the human operator.

ADVANTAGES OF THE MULTI-ATTRIBUTE UTILITY MODEL. The multi-attribute information utility model presented here is characterized by several attractive features. These features, itemized below, offer substantial advantage over the EU decision model. The advantages arise out of the theoretical structure of the model, especially its decomposition property, and have all been empirically demonstrated to some degree in ongoing Perceptronics programs (Samet, Weltman, and Davis, 1976; Steeb, Chen and Freedy, 1977).

a. Generality. The adaptive, multi-attribute model for information selection holds a considerable amount of generality. It can be applied in situations where diagnostic actions can be decomposed into a small set of manageable, quantifiable attributes which have two critical characteristics. First, they must be logically related to the situation-specific demands. That is, their relevance to specific situations must be known. Second, they must directly impact upon a decision maker's choices among competing options. A number of military decision-making environments have already been demonstrated to fit this paradigm (e.g., Coats and McCourt, 1976; Hayes, 1964; McKendry, Enderwick and Harrison, 1971; Samet, 1975).

b. Parsimony. The model is parsimonious; it need only assess an operator's weights for a limited number of information dimensions or attributes. Besides significantly minimizing the model's computational needs and software complexity, this feature reflects findings of psychological experiments (e.g., Hayes, 1964; Slovic, 1975; Wright, 1974) and is in agreement with contemporary decision theory (e.g., Tversky and Kahneman, 1974), all of which suggest that a decision maker can only perform weighting and aggregation on a relatively small number of the important dimensions in the decision task. Also, when decisions are based on a manageable number of information dimensions, they are easier to communicate and rationalize--especially in group decision-making situations (Gardiner and Edwards, 1975). In complex situations, therefore, the reduction in the number of model parameters in the proposed MAU model as compared to the expected utility model are of major importance.

c. Robustness. Like other linear composition models, the multi-attribute decision model is robust; that is, its performance is not significantly degraded by small perturbations in the model's parameters (Dawes and Corrigan, 1974). Such robustness probably contributes to the finding that multi-attribute utility assessment techniques have proven, in certain instances, to be more reliable and valid than direct assessment procedures (Newman, 1975; Samet, 1976).

d. Speed of Adaptation. The adaptive model adjusts all parameters with each incorrectly predicted trainer decision (i.e., action selection). Thus, weights for a specific attribute can be obtained rapidly during sessions in which the trainer performs the simulated CO decisions.

e. Flexibility. The multi-attribute utility model is inherently flexible. If accurate prediction of action selection is not sufficient (i.e., if attribute weights cannot be trained to stable values), additional features or attributes can be added and inappropriate ones deleted. The response to dynamic changes in conditions is similarly flexible. In instances where conditions change rapidly and radically, new sets of weights trained for the new conditions can be substituted. Such weight vectors could be prepared ahead of time by training them either in actual operational situations or in step-through simulations.

UTILITY ESTIMATOR. The dynamic utility estimation technique is based on a trainable pattern classifier. Figure 5 illustrates the mechanism. As the operator performs the task, the on-line utility estimator observes his choice among the available actions at each point in the sequence and views his decision-making as a process of classifying patterns consisting of varying attribute levels. The utility estimator attempts to classify the attribute patterns by means of a linear evaluation (discriminant) function. These classifications are compared with the operator's choices. Whenever they are incorrect, an adaptive, error-correction training algorithm is used to adjust the utilities. A comprehensive discussion of this technique can be found in Freedy, Davis, Steeb, Samet, and Gardiner (1976).

TRAINING ALGORITHM. On each trial, the model uses the previous utility weights ($W_i$) for each attribute (i) to compute the multi-attribute utilities ($MAU_k$) for each action (k). Thus,

$$MAU_k = \sum_{i=1} W_i A_{ik} \tag{15}$$

where

$W_i$ is the weight of the attribute, and

$A_{ik}$ is the level of the $i^{th}$ attribute associated with action k.

The model predicts that the operator will always prefer the action with the maximum MAU value. If the prediction is correct (i.e., the operator chooses the action with the highest MAU), no adjustments are made to the utility weights. However, if the operator chooses an action having a lower MAU value, the algorithm goes into action and applies the error correction training formula. In this manner, the utility estimator "tracks" the operator's decision-making strategy and learns his utilities or weights for the attributes. The training rule used to adjust the weights associated with each of the attributes is illustrated in Figure 5.

Actual in-task training appears feasible using pattern recognition techniques. Instead of batch processing, the pattern recognition methods refine the model decision-by-decision. Briefly, the technique considers the decision maker to respond to the characteristics of the various alternatives as patterns, classifying them according to preference. A linear discriminant function is used to predict this ordinal response behavior, and when amiss, is adjusted using error correcting procedures. This use of pattern recognition as a method for estimation of decision model parameters was apparently first suggested by Slagle (1971). He made the key observation that the process of expected utility maximization involved a linear evaluation function that could be learned from a person's choices.

The suggested technique was soon applied by Freedy, Weisbrod, and Weltman (1973) to the modeling of decision behavior in a simulated intelligence gathering context. Freedy and his associates assumed the decision maker to maximize expected utility on each decision. They assigned a distinct utility, $U(x_{jk})$, to each possible combination of action and outcome, as shown in the decision tree in Figure 6. The probabilities of occurrence of each outcome j given each action k were determined using Bayesian techniques. These patterns of probability were used as inputs to the estimation program (Figure 7). The expected utility of each action $A_k$ was then calculated by forming the dot product of the input probability vector and the respective utility vector. This operation is equivalent to the expected utility calculation:

$$EU(A_k) = \sum_j P(x_{jk}) \cdot U(x_{jk}) \qquad (16)$$

The classification weight vector $W_{jk}$ in the pattern recognition program acts as the utility $U(x_{jk})$. The alternative $A_k$ having the maximum expected utility is selected by the model and compared with the decision maker's choice. If a discrepancy is observed an adjustment is made, as shown in Figure 5. The adjustment moves the utility vectors of the chosen, and predicted, actions ($W_c$ and $W_p$, respectively) in the direction minimizing the prediction error. The adjustment consists of the following:

$$W_c' = W_c - d \cdot P_p \qquad (17)$$

$$W_p' = W_p + d \cdot P_c \qquad (18)$$

where

$W_c'$ is the new vector of weights $[W(x_{1c}), W(x_{2c})]$ for action c

$W_c'$ is the previous weight vector for action c

d is the correction increment

$p_i$ is the probability vector describing the distribution of outcomes

$[P_{1k}, P_{2k}, \ldots, P_{nk}]$ resulting from action k

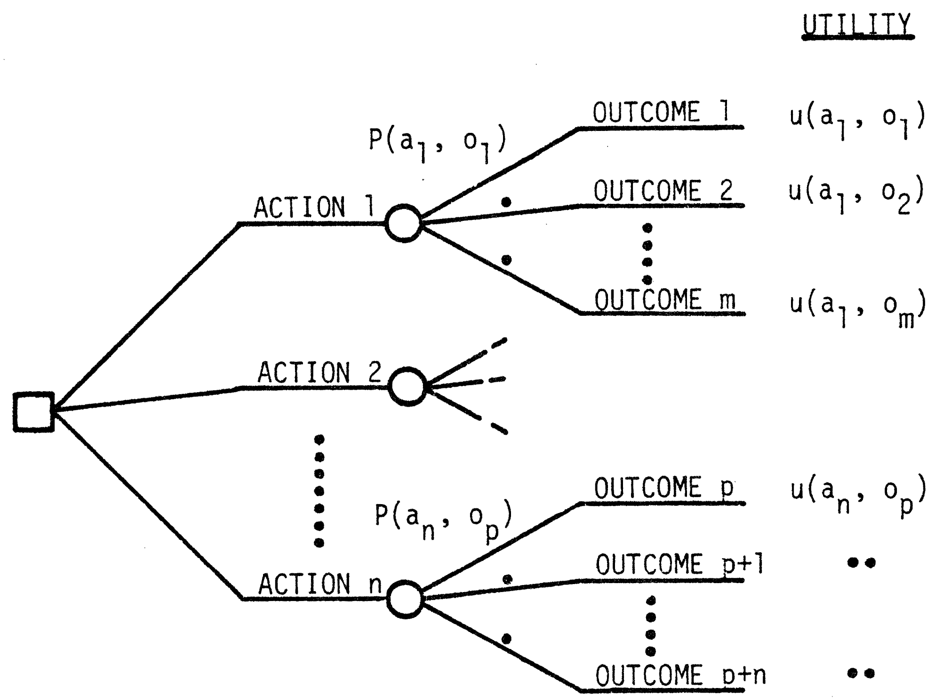



Figure 6. Decision Tree of Utility Estimation Developed by Freedy et al.

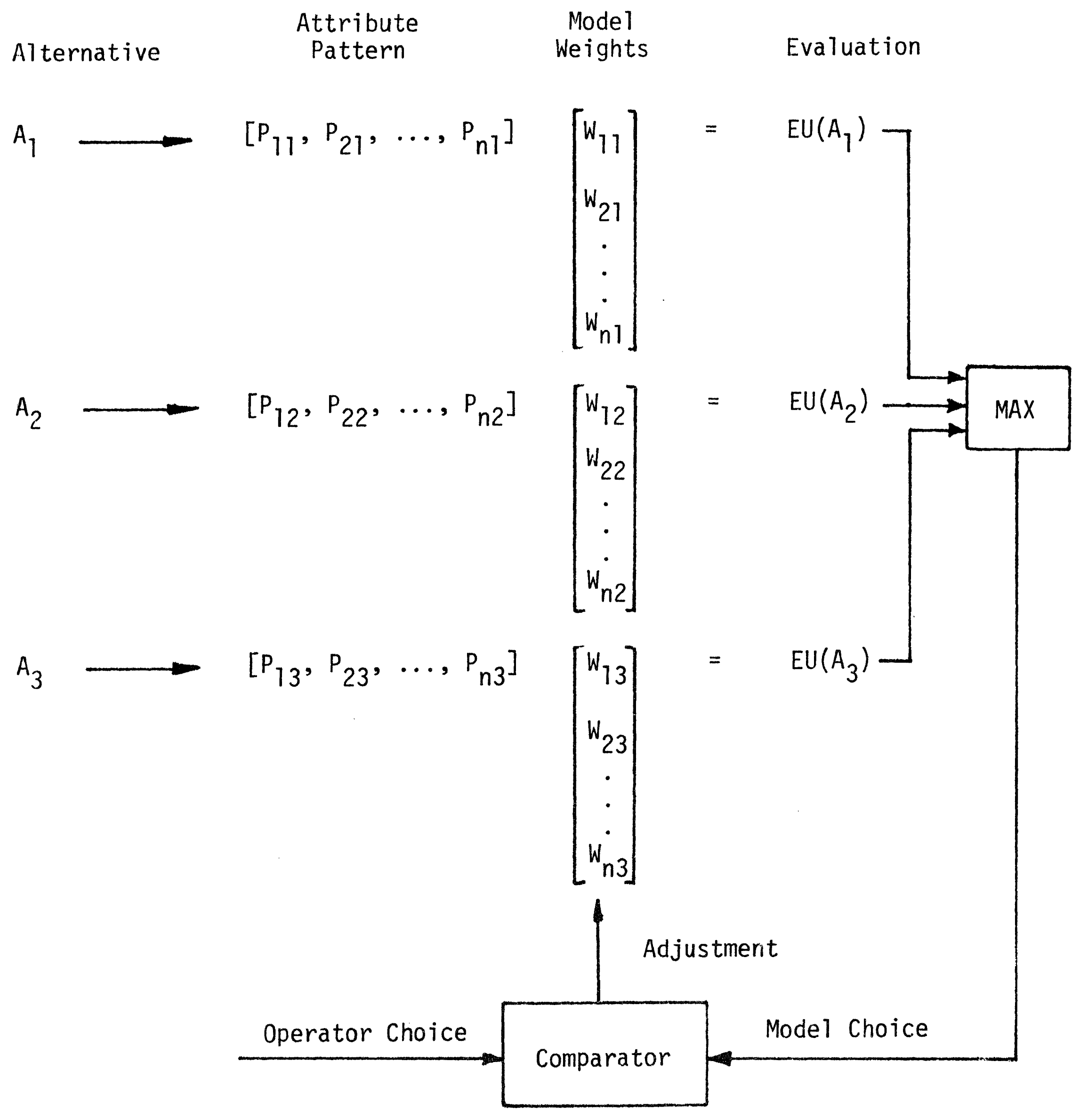



Figure 7. Structure of Utility Estimation Program of Freedy et al. (1973)

The model is an adaptation of the R-category linear machine (Nilsson, 1965). The pattern classifier receives patterns of descriptive data (outcome probabilities) and responds with a decision to classify each of the patterns in one of R categories (actions). The classification is made on the basis of R linear discriminant functions, each of which corresponds to one of the R categories. The discriminant functions are of the form:

$$g_i(x) = W_i \cdot x \text{ for } i=1, 2, \ldots, R \tag{19}$$

where x is the pattern vector and $W_i$ is the weight vector. The pattern classifier computes the value of each discriminant function and selects the category i such that

$$g_i(x) > g_j(x) \tag{20}$$

for all j=1, 2, ..., R; i≠j

A geometric interpretation of the R-category linear machine is shown in Figure 8 (Nilsson, 1965). Decisions involving two possible consequences, $x_1$ and $x_2$, are evaluated according to three discriminant functions $G_1$ ($\underline{x}$), $G_2$ ($\underline{x}$), and $G_3$ ($\underline{x}$). The lines of intersection between the discriminant hyperplanes are the points of indifference between actions. Mappings of these lines of intersection to the attribute plane are shown in the figure. The resulting regions $R_1$, $R_2$, and $R_3$ correspond to the actions maximizing the (expected utility) evaluation function.

The R-category technique becomes somewhat cumbersome if a large number of actions are possible or if the decision circumstances change rapidly. This problem is a result of the assignment of a distinct, holistic utility to each tip of the decision tree. The number of model parameters thus increases rapidly with an increase in the number of actions possible. Also, the only weight vectors adjusted in a given decision are those corresponding to the model-predicted and the actually chosen actions. This partial adjustment makes the system somewhat unresponsive to change.

A natural extension of Freedy's approach is to adapt the single discriminant, multi-attribute approach to the modeling of objective choice behavior. Each possible outcome of a decision can be associated with a set of attributes or objectives of the decision maker. An importance weight vector defined over the various attributes can then be adjusted to predict behavior. The mechanism is simply that of a threshold. The adjustment rule following an incorrect prediction is given in equation 21 with the parameter d controlling the sensitivity of the correction. A large d will cause a fast adjustment but may result in overshoot and oscilitions and a small d will cause slow adaption.

$$W' = W + d(x_c - x_p) \tag{21}$$

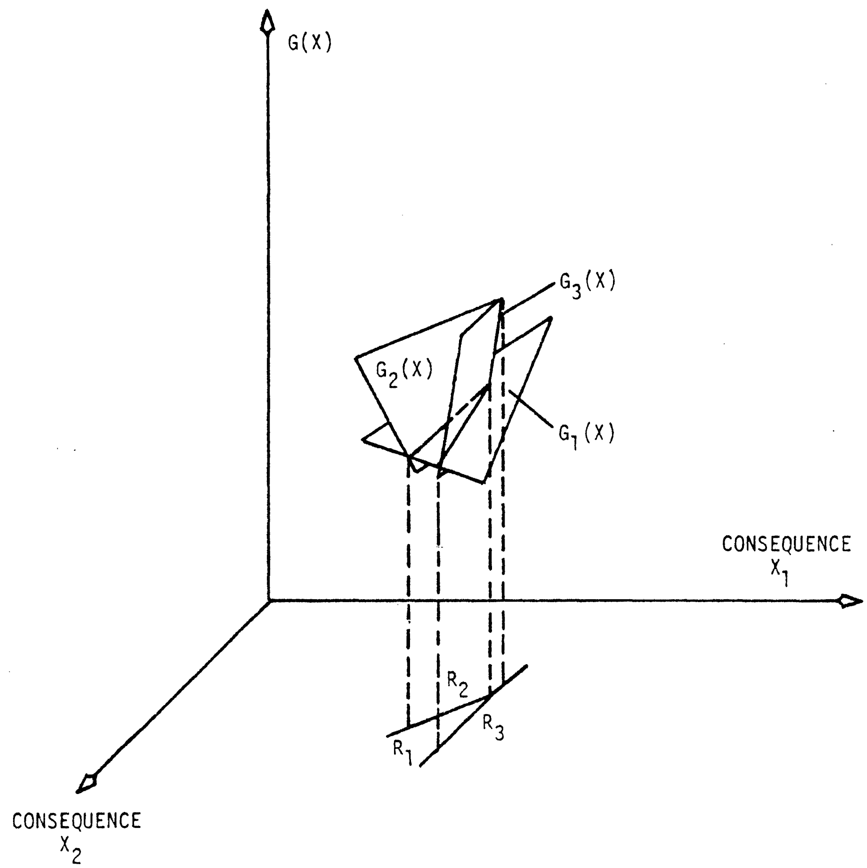



Figure 8. Geometric Interpretation of R-Category Linear Machine (Adapted from Nilsson, 1965)

where

$W'$ is the updated weighting vector

$W$ is the previous weighting vector

$x_p$ is the attribute pattern of the model-predicted choice

$x_c$ is the attribute pattern of the decision maker's choice

$d$ is the adjustment factor.

A possible advantage of the pattern recognition technique over many of the other forms of estimation is its flexibility of adjustment. Several types of error correction are possible for the adjustment rule, each with a different combination of speed, stability, and complexity. The three principle forms are the fixed increment rule, the absolute correction rule, and the fractional correction rule. These differ solely in their formulation of the adjustment factor d in Equation 21.

The fixed increment rule simply assigns a non-zero constant to d. Thus the movement of the weight vector is a constant proportion of the difference in the predicted and chosen patterns. The correction may not be sufficient to avoid subsequent errors with the same pattern, but the process is eventually convergent (Duda and Hart, 1973). The fixed increment rule has the advantages of simplicity and relative insensitivity to inconsistent behavior.

A more rapid but also more potentially unstable rule is the absolute correction rule. This method sets d to be the smallest integer at which the error of the pattern is corrected. In the decision modeling situation, this becomes:

$$d - \text{smallest integer} > \frac{|k \cdot (x_c - x_p)|}{(x_c - x_p) \cdot x_c - x_p)} \qquad (22)$$

in which

$x_c$ is the attribute level vector of the operator selected choice

$x_p$ is the attribute vector of the predicted choice

The fractional correction rule is similar to the absolute rule but is typically less extreme. The fractional rule moves the weight point some fraction of the above distance:

$$d = \frac{\lambda|k \cdot (x_c - x_p)|}{(x_c - x_p)(x_c - x_p)} \qquad (23)$$

where $\lambda$ is a constant $0 < \lambda > 2$.

All three of the adjustment rules have been proven convergent with linearly separable patterns (Nilsson, 1965). The speed of convergence is normally fastest with the absolute rule. This is illustrated for an example series of adjustments in Figure 9. The set of four numbered lines in the figure are a sequence of patterns. These patterns are shown as hyperplanes in a 2-dimensional weight space. Each hyperplane represents the difference between two multi-attribute vectors. The operator choice is shown by the direction of the arrow at each pattern. The absolute rule, (the triangles in the figure) achieves correct prediction after four observations, while the fixed rule (the circles) requires five. Unfortunately, the absolute rule is expected to be less forgiving of inconsistent behavior than the fixed or fractional rules. This is because of the large responses the absolute rule makes to operator inconsistencies. The fixed and fractional rules may exhibit a greater tendency to smooth or average the behavior.

AN EXAMPLE. For an example of how the adaptive decision analysis approach is applied, consider the select maneuver decision. Assume it has already been decided that the goal of the maneuver should be to evade.

Assume that the following alternative evasive maneuvers are available:

a. Sink to the bottom and hide.

b. Run (full speed in straight line).

c. Sink to bottom and deploy decoy.

d. Run in a zigzag pattern.

e. Run and deploy decoy.

The following attributes could be used:

Information Gain. This represents the expected information gained by friend about the opponent as a result of the action being considered. This is dependent on the probability (assessed by opponent) that friend has already detected him. Thus, if friend already has a lot of information there's not much information left to be gained.

Deception. This is the expected amount of false information gained as a result of decoying. This may be situation dependent. In the example, releasing a decoy would have greater deception value if the sub is resting on the bottom, than if it is going full speed ahead. Also, if you haven't yet been detected, deploying a decoy will give away the fact that you are in the area.

Vulnerability. This attribute represents your vulnerability to being hit if you are detected. The attribute levels for vulnerability should be subjectively estimated and defined in advance for each alternative.

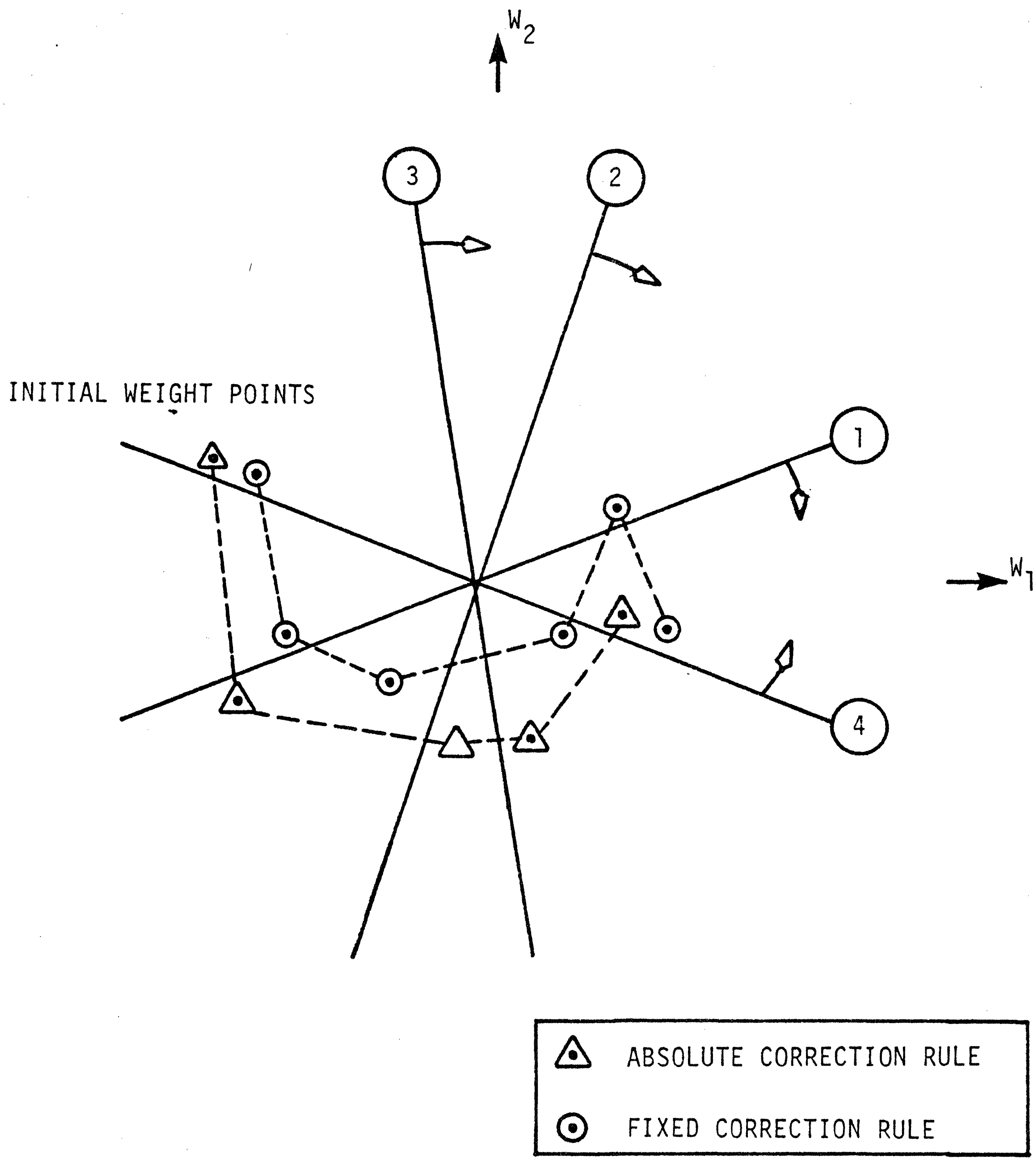



Figure 9. Comparison of Behavior of Convergence Rules

Cost. This is the direct cost of the alternative. Cost may also be used as a gross resource depletion attribute.

Effect on Mission Objective. This attribute should be redefined with subjective weights.

Table 2 gives an example of the MAU approach to optimal action selection. Each column represents one alternative action that the CO can choose in the tactical situation. Each row represents one of the key attributes by which each action is evaluated. The values in the table are predetermined or calculated from the features of the tactical situation. In the example given, the tactical situation is the following:

a. The opponent is 80 percent sure friend has not detected him (thus the "run" alternatives may cause high information gain).

b. The deception effect of a decoy is higher if chosen with a "sink" rather than "run" alternative.

c. Cost includes fuel, weapons and decoy expenses, and are .1, .4, .6, .5 and .8, respectively.

The first column gives the utility value associated with each attribute by a trainer. These values are the result of the adaptive training algorithm. They are positive for good attributes (for the opponent objectives) and negative for bad ones. The MAU processor will select the alternative action that would have the highest combined value. This is done by a weighted sum of utilities times the attribute level. These values are calculated and rank ordered at the bottom of the table. Alternative #1 turns out to have the highest value and it is the one the system will select. In a different tactical situation the attribute levels of the various options may be different (e.g., friend has detected the opponent) causing another action option to come up on top and that action would be the one the opponent model would select to activate the simulated opponent on the screen.

## THE HEURISTIC SEARCH APPROACH

STATE SPACE MODEL. The overall objective of knowledgeable opponent scenario generation is to provide a realistic simulation of an active enemy. The enemy would react to events and actions taken by the friendly forces and choose a course of action that would lead to the achievement of some enemy goal, which usually means a bad outcome for the friendly forces. The heuristic search approach provides such a mechanism.

In the underlying model, which is called the "state space" model, the problem domain (such as underwater warfare) is expressed in terms of "states," which are complete descriptions of the tactical situations as they exist at some particular instant of time (Nilsson, 1971). An "action" is a transformation which, when applicable, converts one state into another. Thus, a sequence of actions ("plan" or "allocation") converts some initial state into a final, or goal, state. The enemy submarine

TABLE 2. ATTRIBUTE LEVELS, VALUES, AND EXPECTED VALUES FOR EXAMPLE SCENARIO

| Attribute | Utility | Sink to bottom & hide | Run | Sink and deploy decoy | Run in zig zag pattern | Run and deploy decoy |
|---|---|---|---|---|---|---|
| Information Gain | -1.0 | 0.0 | 0.7 | 0.5 | 0.8 | 1.0 |
| Deception | +0.5 | 0.3 | 0.0 | 0.8 | 0.0 | 0.5 |
| Vulnerability | -0.8 | 1.0 | 0.5 | 1.0 | 0.2 | 0.2 |
| Cost | -0.2 | 0.0 | 0.5 | 0.9 | 0.6 | 1.0 |
| Effect on Mission Objective | +0.2 | -0.9 | 1.0 | -1.0 | 0.7 | 0.6 |
| MAU Value of Choice | 0.0 | -0.47 | -1.16 | -0.98 | -1.18 | -0.77 |
| Rank Order | | 1 Best | 4 | 3 | 5 worst | 2 |

commander asks the questions, "What sequence of actions can transform the current state into a goal state which satisfies my overall objectives?" In other words, "How do I get from where I am to where I want to go?" Before a system can perform properly, it must know what actions are available, under what circumstances they can be applied, what their effects are, and what possible states can arise from their use.

BASIC SEARCH TECHNIQUES. The most basic search techniques are systematic expansions of the state space. Starting from the start node (labeled 1 in Figure 10--the current state), the search algorithm expands all its possible successive nodes. When a goal node is encountered, the path from the initial node to that goal node is the solution sought. In the ASW case, it is the strategy, or sequence of actions, the commander has to take to reach his objective.

Figure 10 shows the most elementary algorithms--the "breadth-first" and the "depth-first" algorithms, respectively. In the "breadth-first" algorithm, each node is expanded completely--all its "sons" identified--before the next is started. This method is guaranteed to find the shortest path from the start to the goal nodes. The numbers in Figure 10 indicate the order of node expansion.

In the "depth-first" algorithm, each alternative line of inquiry is sought to the fullest depth before other alternatives are evaluated. When such a search fails, the algorithm tries the next deepest possibility. Figure 10 also shows the order of node expansion in this algorithm. The depth first algorithm does not guarantee the shortest path to a goal if more than one goal node exists.

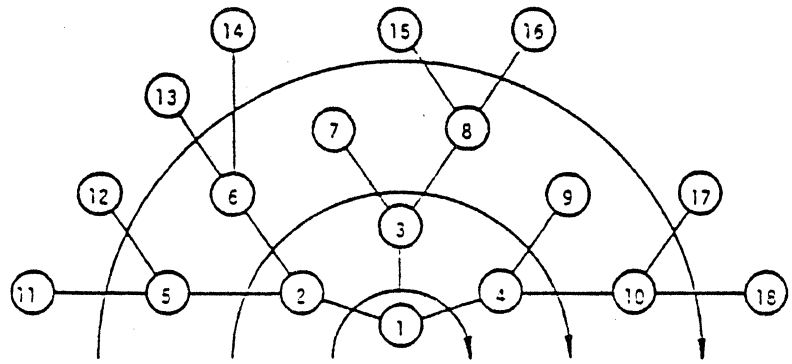
These search methods are "blind" methods because they develop systematically every node in the state space without using any information which may be known in advance about the particular problem domain or the particular knowledge found in the nodes that has already been expanded to guide the search process. The heuristic search approach is the class of algorithms that uses such domain specific knowledge to guide the search.

HEURISTIC SEARCH METHODS. Heuristic search methods try to utilize any information known about the problem domain to guide the search for a solution in the state space. The added information helps avoid the combinatorial explosion of computer resources (time and memory) needed for the basic search techniques. Figure 11 illustrates the basic idea of the heuristic search approach by comparing it to depth first and breadth first searches. The contours of node expansion are directed toward the goals G1 and G2, in contrast to the blind search algorithm. Applying a heuristic search usually leads to the discovery of optimal or suboptimal solutions in cases that would be too big to handle by standard techniques. Many achievements of heuristic search are known. For example,

a. Computer Aided Design (Powers, 1973; Hagendorf et al, 1975).

b. Test Sequence Generation for Detection of Failures in Clockmode Sequential Circuits (Hill and Huey), 1977.

Breadth
First




Depth
First

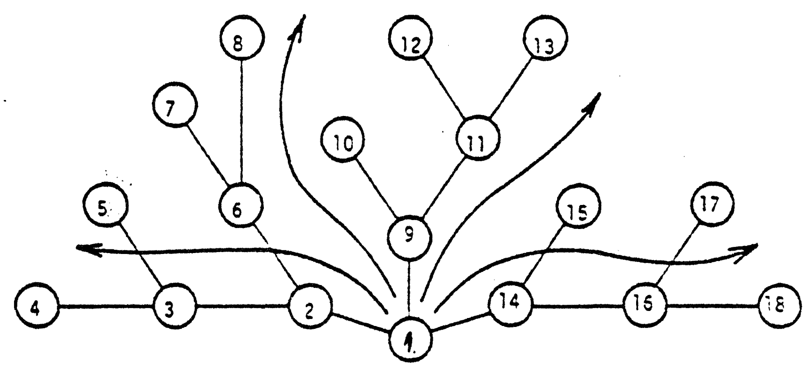



Figure 10. Breadth and Depth-First Expansion Order

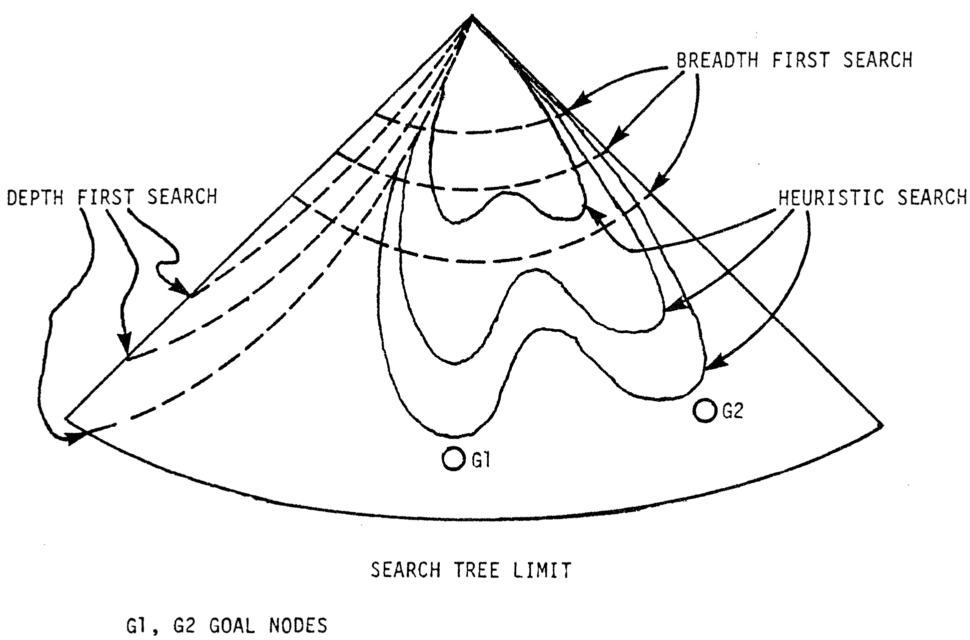



Figure 11. Expansion Contours of Depth-First Breadth-First and Heuristic Search Methods

c. Edge and Contour Detection (Martelli, 1976).

d. Chromosome Matching (Montanari, 1970).

e. Organic Chemical Synthesis (Sridharan, 1973).

f. Ballistic Missile Defense (Leal, 1977).

g. Discovery of Mathematical Concepts (Lenat, 1978).

The heuristic information can be contained in different parts of the search algorithm. If $\Gamma$ is the function that generates node successors and f $(\eta)$ is an estimate of the promise of node $\eta$ to be on the path to a goal node, then the heuristic information may be contained in either of them. Using knowledge in $\Gamma$, the search algorithm would generate first the more probable successors of a node. On the other hand, using knowledge in f $(\eta)$ the most promising nodes would be selected for subsequent development in the face of less promising ones.

THE MINIMAX AND $\alpha\beta$ ALGORITHMS. Two algorithms which have particular applicability to the case of military confrontation are the minimax and the $\alpha\beta$ algorithms. The minimax is applicable in zero-sum adversary confrontations where what is good for one side is bad for the other. When developing the state space of such a problem, the prudent decision maker has to assume that, when given the choice, the enemy would select the alternative which is the most damaging to the decision maker"s own objectives. When expanding the search space for this problem, as shown in Figure 12, the commander first determines all the alternatives available to him. This is the maximizing level because at this level the commander has the choice, and he will obviously choose the alternative that maximizes his measure of success. The next level is the set of responses available to the enemy for each of the commander's choices. Here the enemy will make the choice, and he will choose the worst alternative (from the commander's point of view). Thus, this layer is called the minimizing level. The maximizing and minimizing of layers continues downward in the tree until the allocated computing resources are used up. At that point, the static value of each tip node is evaluated. The value of a tip node is a measure of how "good" is the state represented by the node from the commander point of view. If the layer of nodes just above the tip nodes is a "maximizing" layer, each node in it assumes the maximal value of its "children" nodes (and vice versa for a minimizing layer). These "backed-up" values propagate upward in the state space tree until they reach the top layer. The minimaxed values that reached the layer just under the current state (the root of the tree) are the basis of the commander's choice among the alternative actions available to him. This "minimaxing" algorithm is repeated for every decision the simulated commander has to make; thus, it takes into account the dynamics of the situation, and it finds the best tactical move foreseeing the best choice of the enemy. In this algorithm, the heuristic information is contained in the tip node evaluation function f (n) in the previous section.

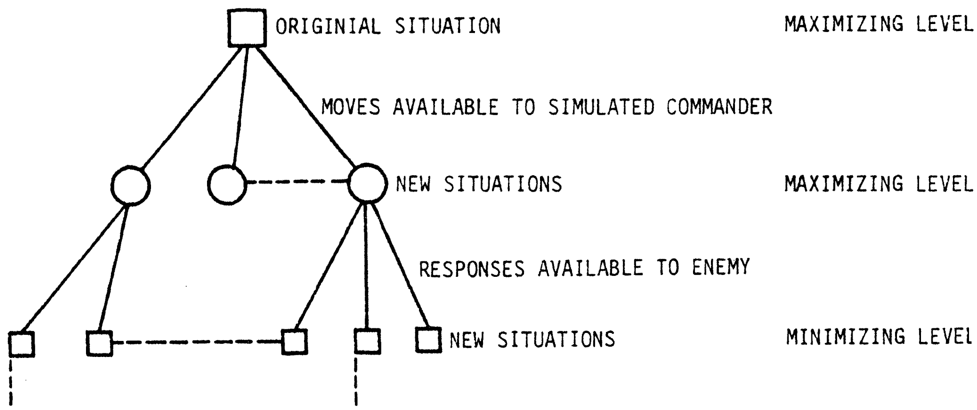



Figure 12. The Minimax Algorithm Tree Development

The alpha-beta algorithm is an improved version of the basic minimax algorithm. The alpha-beta algorithm is a systematic method to reduce the number of nodes that have to be evaluated and even makes it unnecessary to expand complete branches of the state space tree. It can be shown that although the algorithm allows a large part of the search tree to be completely ignored, it will not lose any solution that the basic minimax algorithm would find.

The alpha-beta algorithm starts with a depth-first expansion of the tree down to some level n (see Figure 13). When the depth limit is reached, the tip nodes are evaluated and temporary values are backed-up in the tree. The alpha-beta technique takes advantage of these preliminary values. Consider, in Figure 13, the maximizing node A in the tree after nodes 4-9 have been developed below it. A has been assigned a temporary value of 0.2 (propagated from node 5). B, which is a minimizing node, has been assigned a temporary value of 0.1 (propagated from node 9).

At this time, there is no point developing any other successor to the node B (such as C) because, since it is a minimizing node, the best value B can get is 0.1 or lower, and node A, being a maximizing node, will always select 0.2 over 0.1. This argument is the "alpha" half of the alpha-beta pruning. The empty nodes in Figure 13 show all the subtrees that will be pruned off and the order of node generation. In fact, the empty nodes need not be generated at all.

The "beta" half operates in precisely the reverse for nodes in the minimum layers. By using the alpha-beta algorithm, the tree can be explored approximately twice as deep as a simple minimax algorithm, while expanding the same number of nodes. The algorithm is somewhat slower, inasmuch as it has to do the bookkeeping for the temporary alpha and beta values. The alpha-beta algorithm is a very promising potential opponent model.

## ADVANTAGES

a. Heuristic search techniques have a wide range applicability, as can be seen from the examples mentioned above.

b. The underlying structure (state-space, AND/OR graphs) is very general and fits naturally all problems of a combinational nature and all hierarchical problems which can be decomposed into goals and subgoals (this includes decision trees).

c. General theoretical results are available.

d. It is universally accepted that heuristics are crucial to cope with intractable problems.

## SCOPE AND LIMITATIONS

a. Heuristic search techniques are designed for problems of a particular nature only, with well-defined states, subgoals or subproblems. Problems with a continuous nature, for instance planning in a continuum, cannot be solved via heuristic search.

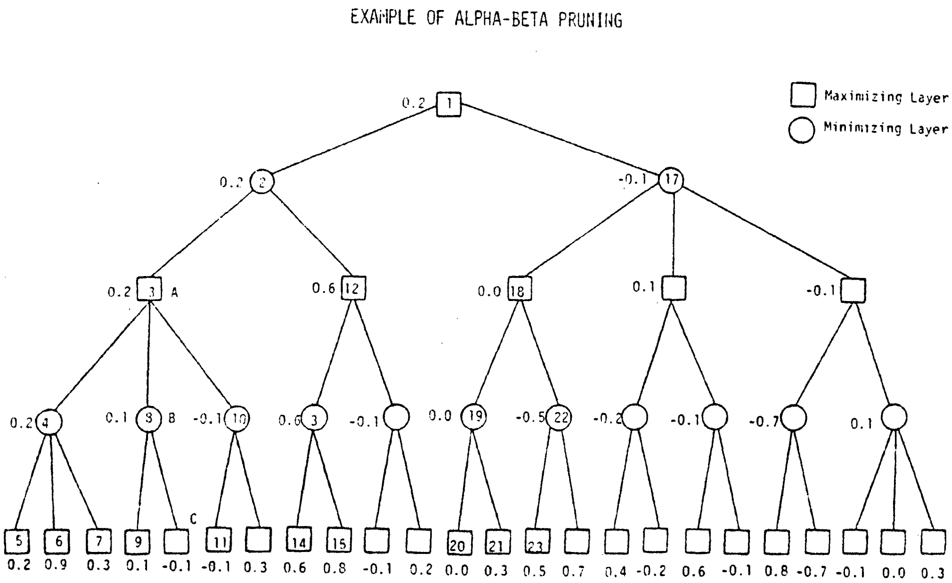



Figure 13. Example of Alpha-Beta Pruning

b. The use of heuristic search itself poses a problem. The more specific a heuristic function, the more efficient it is in guiding the search. How well designed and problem-specific heuristics are will therefore determine their efficiency.

c. Heuristic search might be subject to catastrophes (if no solution is found after the computational resources are exhausted or an insufficiently good solution is found).

## PRODUCTION RULES APPROACH

OVERVIEW. Production rule systems represent another successful approach for knowledge representation and deductive mechanisms. This approach is similar to the heuristic search approach in that it uses a modification of the state space model as the underlying conceptualization (see definition in the section on heuristic search). The technique of representing the knowledge is different, however, and so is the mechanism which finds the path from the current state to the goal state. The problem specific knowledge (heuristics) is packaged in production-rule systems as small modular "chunks" called productions.

A production is a rule which consists of a situation-recognition part and an action part. Thus a production is a "situation--action" pair in which the left side is a list of things to watch for in the description of the current state of the world, and the right side is the list of things to do in that case.

In the case of submarine warfare, a production that guides the commander's actions may be something like:

If

AND

Enemy dominates area

Enemy has not yet detected you

You are out of his torpedo range

You are in very shallow water

Then

Escape by sinking to bottom in silence

The effect of such a production is to respond to the situation when all the aspects combined by the AND are present and change the current action from whatever it was before to ESCAPE.

In addition to the large set of such productions, the production rule system contains a triggering mechanism that uniformly checks all the productions that apply in a given situation (by testing for truth of the left hand side of each production) and applies those that are applicable--causing the situation to change.

The main advantages of the production rule approach are the ease and modularity of the knowledge representation. Consequently, it is easy to elicit information from experts without requiring that they be programmers. In fact, many training manuals are written already in "production rule style." Furthermore, the information is incremental; thus it is easily modified, updated and expanded into new areas of expertise. It is also usually argued by production rule proponents that this form of knowledge representation is highly compatible with human cognition, making it a very useful and powerful training tool. For example, suppose an opponent commander model is built as a production rule system. It becomes very easy to communicate with the system and ask "Why have you done that?" meaning what aspects of the situation or what actions of the trainee caused some unexpected response of the simulated enemy commander.

The trainee can discover specifically where he went wrong, and he can start in mid action and try other alternatives. At the same time, this is also a powerful debugging tool allowing experts to tune the system by following its reasoning process and identifying the specific cause for a mistaken conclusion which led to an unreasonable response.

THE PRODUCTIONS. As AND/OR graphs (a graph with nodes combined by logical AND or OR functions), production systems are composed of two parts: the set of productions and a mechanism to find a solution in a given situation. We will discuss first a graphic representation of the productions themselves. A simple production specifies a single conclusion which follows from the simultaneous satisfaction of the situation recognition conditions. Any particular conclusion may spring from any production. The conclusion specified in a production follows from the AND or "conjunction" of the facts specified in the premise recognition part. A conclusion reached by more than one production is said to be the OR or "disjunction" of those productions. Depicting these relationships graphically produces an AND/OR graph. Figure 14 shows an AND/OR graph which reaches from base tactical facts ($F_i$) on the left, through the different productions ($P_i$), to a conclusion or an act to be taken, on the right side of the figure. Any collection of productions implies such a graph. In Figure 14 we used the set of submarine warfare productions given in Figure 15. These productions should be taken as an example of the capabilities of this approach.

The arrangement of nodes in this graph focuses on how the conclusion can be reached by various combinations of basic facts. As with ordinary AND/OR trees, a conclusion is verified if it is possible to connect it with basic facts through a set of satisfied AND/OR nodes. Different sets of facts can be used to reach a given conclusion by selecting different branches at OR nodes.

Sometimes it is useful to look at the implied graph to get a better feel for the problem space, noting whether the reasoning is likely to be broad and shallow, narrow and deep, or broad and deep. Again, however, caution is in order. When used prominently in discussions of goals and

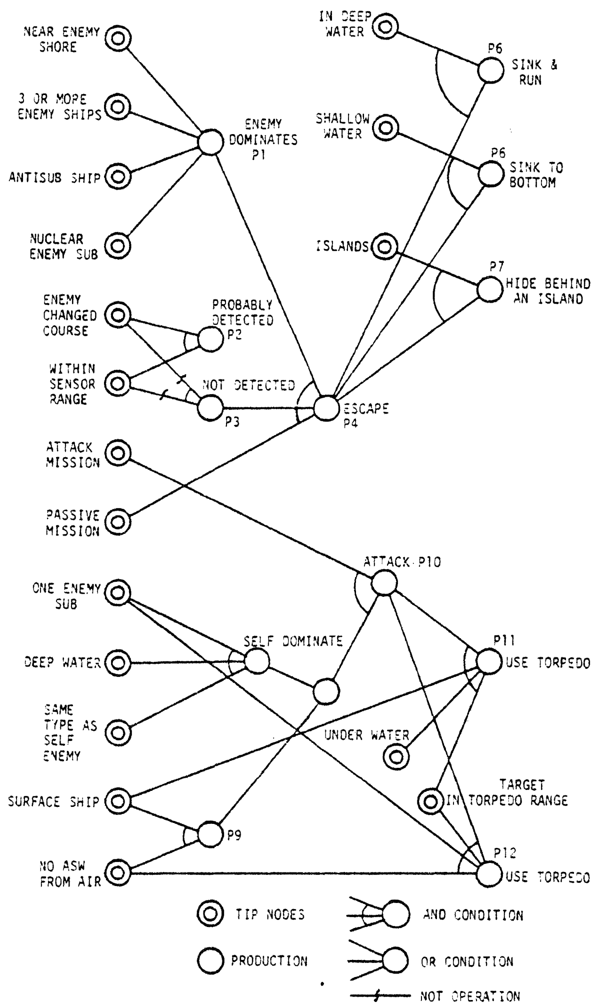



Figure 14. And/Or Graph

<u>P1</u>
IF
  OR
    Location near enemy shore
    3 or more enemy ships in the area
    Anti sub ship in area
    Nuclear enemy sub
THEN
    ENEMY DOMINATES SCENE

<u>P2, P3</u>
IF
  AND
    SELF WITHIN SENSOR RANGE
    ENEMY CHANGED COURSE
  THEN
    SELF DETECTED
  ELSE
    NOT DETECTED

<u>P4</u>
IF
  AND
    SELF on passive mission
    SELF not detected
    Enemy dominates area
  THEN
    ESCAPE

<u>P5</u>
IF
  AND
    ESCAPE
    SELF in deep water
  THEN
    Sink deep and run

<u>P6</u>
IF
  AND
    ESCAPE
    Self in shallow water
  THEN
    Sink to bottom in silence

<u>P7</u>
IF
  AND
   ESCAPE
   SELF in Islandic area
  THEN
    Hide behind and island

<u>P8, P9</u>
IF
  OR
   AND
     One enemy sub in area
     Self in deep water
     Enemy sub of same type
   AND
     Enemy surface ship alone
     NO ASW in air
THEN
   SELF dominate

Figure 15. Production Rule Example

subgoals, and/or graph representations tend to make control look like a search problem with the various search ideas becoming applicable. This position has its good and bad features. One bad feature is that it can create a tendency to waste time with an existing problem space rather than to make a better space, where less search, if any, would be needed.

THE CONTROL MECHANISM. The control mechanism which utilizes the set of productions takes a collection of known facts about the situation and makes new conclusions according to productions that are satisfied by the initial facts. In operation, the user would first gather up all facts available and present them to the system. The control mechanism will then scan the production list for a production which has a matching situation part, i.e., all the premises in the left hand side are satisfied. This production will be activated and its action side will change the facts known about the situation. In the example given, if P1 was activated, it adds the conclusion that the "enemy dominates the area" to the situation description.

Reasoning from base facts to a conclusion rarely entails using only a single step, however. More often, intermediate facts are generated and used, making the reasoning process more complicated and powerful. One consequence is that the individual productions involved can be small, easily understood, easily used, and easily created. Also notice that the intermediate facts added by the lower level productions are tactical facts meaningful to the military users of the system, resulting in many benefits. Using this approach, a simulated submarine commander can produce a chain of conclusions leading to intelligent tactical actions, even as a trainee commander makes his actions dynamically.

In the event many productions have premises or situation specifications that are satisfied simultaneously, there must be some way of deciding among them. Here are some of the popular methods:

a. All productions are arranged in one long list. The first matching production is the one used. The others are ignored.

b. The matching production with the toughest requirements is the one used, where "toughest" means the longest list of constraining premises or situation elements.

c. The matching production most recently used is used again.

d. Some aspects of the total situation are considered more important. Productions matching high priority situation elements are privileged.

So far, the deduction oriented production system is assumed to work from known facts to new, deduced facts. Running this way, a system is said to exhibit "forward chaining." But "backward chaining" is also possible, for the production system user can hypothesize a conclusion or a desired final state and use the productions to work backward toward an enumeration of the facts that would support the hypothesis. For example, (see Figure 14) in the case of a submarine commander, the system can start from the mission, e.g., attack enemy sub. Then chaining backward from

(P10), it will conclude that it has to achieve self-dominance. This can be achieved by confronting an enemy surface ship (P9) or an enemy sub of the same type in deep water (P8). Thus, by a small change of orientation, the same set of productions was used backwards. Knowing that a deduction-oriented production system can run forward or backward, which is better? The question is decided by the purpose of the reasoning and by the shape of the problem space. Certainly, if the goal is to discover all that can be deduced from a given set of facts, then the production system must run forward. The production system can run forward from all premise elements as long as suitable productions exist. Using sensory systems to supply more facts is necessary only when no productions apply, and no conclusion has been reached. On the other hand, if the purpose is to verify or deny a particular conclusion, or reach a desired situation through a sequence of actions, then the production system is probably best run backward from that conclusion. Avoiding needless fact accumulation is one result obtained; indeed, no irrelevant facts need be checked at all.

Deciding whether forward chaining or backward chaining is better depends, in part, on the shape of the space. Figure 16 illustrates this by way of two symmetric situations. All possible states are represented along with the operations that can change on e state into a neighbor. In the first situation shown, forward chaining is better because there is a general fan-in from the typical initial states toward the typical goal states. It is hard to get into a dead end. In the second situation, the shape favors backward chaining since there is fan out.

ADVANTAGES. Proponents of production rule systems usually cite one or more of the following advantages:

a. Production systems provide a powerful model of the basic human problem solving mechanisms. This results in easy expert elicitation, user communication at the comfortable level of military tactical concepts and terms, easy trouble-shooting, and good training capability.

b. System states are meaningful to users, debuggers, etc.; thus an evaluation can be made on the tactical level rather than in the computer implementation level.

c. Production systems enforce a homogeneous representation of knowledge, effectively separating the static data representation from the uniformly applied evaluation mechanism.

d. The control mechanism is simple and explicit on what to do next, is clear from the current state what productions are available.

e. Production systems allow incremental growth through the addition of individual productions and without changes necessary to any others.

f. Production systems allow unplanned, but useful, interactions which are not possible with control structures in which all procedural interactions are determined beforehand. A piece of knowledge, or a combination of such, can be applied whenever appropriate, not just whenever a

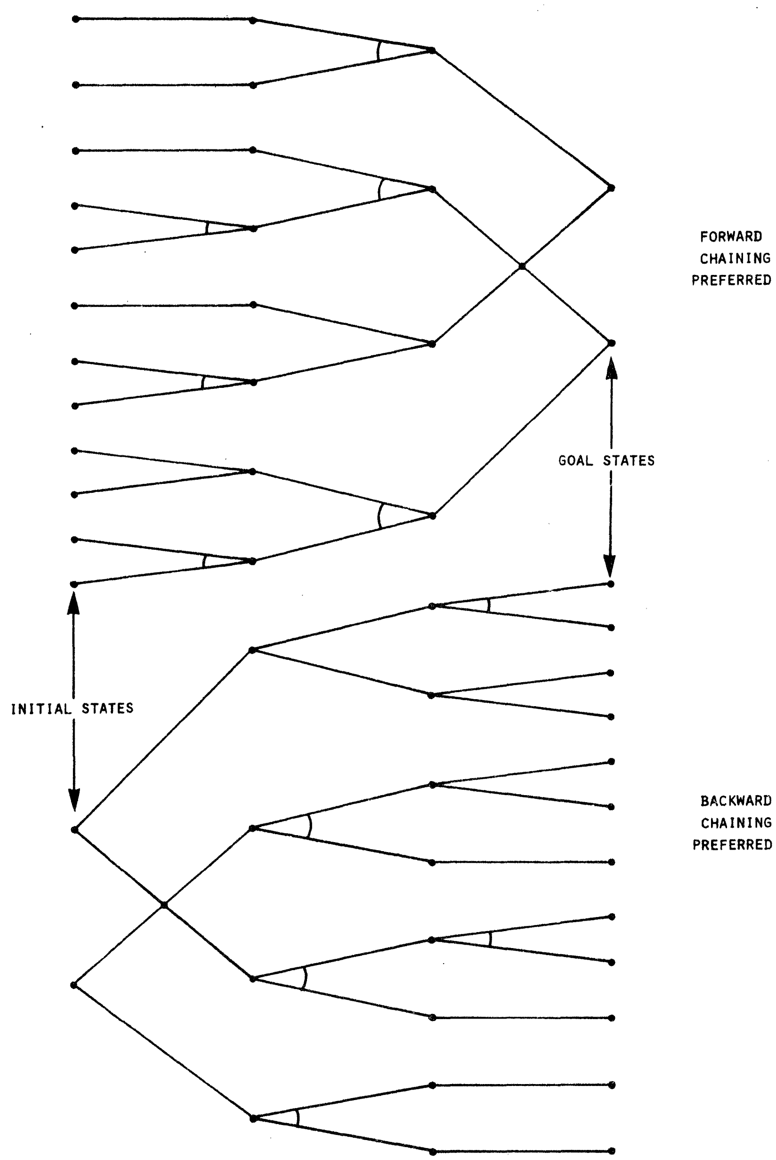



Figure 16. And/Or Graph Shapes for Forward or Backward Chaining

programmer predicts it can be appropriate. This can lead to highly intelligent performance by systems with a surprisingly small (several hundreds) set of productions.

g. Providing explanation capability to the system is natural to implement. When some decision is made, the system can present the sequence of productions that led to that conclusion, thus affording its "reasoning" about the situation.

h. The production rule approach is as general as any other method based on the state space model.

i. Productions can be quantified with probability information leading to applicability in decision making and risk evaluation.

DISADVANTAGES. Some of the advantages of the production rule approach can become disadvantages if care is not exercized in the design process:

a. Maintaining focus of attention: It would seem that PR systems allow knowledge to be tossed into the system homogeneously and incrementally without worry about relating new knowledge quanta to old. Thus, by relinquishing control, such system allow unimportant productions to usurp center stage from more important productions, leading the process astray.

b. Size problems: One particular problem is that production systems may break down in the amount of knowledge is too large, or when the number of productions grows beyond reasonable bounds. The advantage of not needing to worry about the interactions among the productions can become the disadvantage of not being able to influence the interactions among the larger number of productions.

The possible solution, of course, is to partition the facts and the productions into subsystems such that at any time only a manageable number are under consideration. Within each subsystem, some productions may be devoted to arranging transfer of information or attention to another subsystem. Curiously, some users of Hewitt's ACTORS language produce programs that have a strong resemblance to systems of communicating production subsystems.

This solution, however, goes against one of the main advantages of production rule systems, namely, modularity and independent control. If control guiding productions are added, we again have the problem of explicitly directing where control should go.

c. Global effects: It is awkward to represent global effects using PR approach. Here, again, the modularity of the productions requires that if some global effects (such as weather in ASW) take part in many productions, it is necessary to duplicate the whole set of productions which behave differently for each different weather state.

# SECTION V

# MODEL EVALUATION

## EVALUATION ATTRIBUTES

The attributes for evaluating different opponent models are described below. These attributes are divided into three categories:

a. Modeling Attributes.

b. Development Attributes.

c. Performance Attributes.

## MODELING ATTRIBUTES

a. Flexibility for Modeling Different Opponents. How easy it is to change the opponent's appearance of tactical behavior such as smart/dumb, aggressive/defensive, cautious/risky, type of simulated sub, and mission type.

b. Ability to Model Subjective Operator Decision Criteria. How well the model deals with subjectivity. Can the model make use of the operator's internal preference--value structure?

c. Modeling Continuous Behavior. Continuous behavior means that the parameters representing the behavior (sub x, y location) can vary in infinitesimal increments rather than between a few discrete alternatives.

d. Modeling the Flow of Control. (Representing in a flexible manner the sequence of processing.) Processing may be a decision--selecting among alternatives or assessing a situation, or it may be an action. The flow of control may further be parallel or sequential, instantaneous or protracted, synchronous or a synchronous, and event driven versus schedule driven.

e. Modeling AND and OR Conditions. Can the model represent complicated, logically structured criteria (i.e., a set of conditions linked by AND's and OR's) for making a decision.

f. Modeling Probabilities. The capability of the model to respond to probabilistic inputs and to give probabilistic outputs (or make a Monte Carlo selection of outputs).

g. Conciseness of Representation. The quantity of parameters, data, or code needed to represent a particular behavior.

h. Adaptiveness. Can the model modify automatically its own parameters in response to external events. The training is done on-line, in task and in real time.

i. Dependencies Among Input Variables. The difficulty of applying the model when dependent relationships among the input variables exist.

j. Auxilliary Payoffs. This represents extra features available with the particular modeling approach. Examples are: ability to explain decision selections, ability to output relative desirability of the alternatives, performance measures, etc.

DEVELOPMENT ATTRIBUTES

a. Scenario Set-Up Time. This is the time and effort required to specify a new scenario or enemy behavior. This function is done by the instructor ahead of the training session.

b. Required Development and Implementation Time and Cost. This includes the time spent by analysts, the amount of research required, the required size and complexity of the software, ease of debugging, computer resources required, etc.

c. Required Integration Time and Cost. The difficulty of integrating the new software into the current SCST software systems.

d. Vulnerability to Increase in the Size of the State Space. This represents the degree to which development, implementation, and modeling difficulty increases with the size of the state space. More vulnerability means that the complexity increases more rapidly than the increase in state space size. Vulnerability carries the risk of the problem "blowing up" or becoming intractable.

PERFORMANCE ATTRIBUTES

a. Instructor Time Needed for Operation. The amount of effort and interaction required of the instructor during operation. Hopefully, the instructor's burden would be decreased rather than increased.

b. Instructor Control. This represents problems of synchronizing the model to allow smooth transitions from instructor control to model control and vice versa.

c. Required Computer Resources. Run time and memory requirements during model operation.

d. Trainee Evaluation and Performance Measurement. This represents the degree to which trainee performance measures are naturally and readily available from the model.

e. Real World Fidelity. The degree to which the model reflects real world behavior patterns.

EVALUATION BY MODELING ATTRIBUTES

a. Flexibility of Modeling Different Opponents. In evaluating flexibility we are not considering the number of parameters that have to be adjusted to bring about a particular behavior--because any pre-defined set can be brought in from back-up memory in essentially the same speed. Rather, we are concerned with how easy it is to obtain the parameters and identify the parameters that have to be replaced. This related to the consideration of how transparent the representation is with respect to knowing what behavior a particular parameter creates and vice versa. The Adaptive Decision approach is the easiest in that a particular behavior can be generated automatically by training the system on samples of the desired behavior. However, this approach is not transparent unless all the attributes used are explicitly meaningful to the decision maker. The production rules approach offers the greatest transparency and clarity becuase particular behaviors are generated in a few localized productions and they are stated there in (almost) plain language rather than a collection of numbers. The Elicited Probability approach is non-automatic (the conditional probabilities, etc., have to be elicited explicitly from experts) and it is also less transparent than the Adaptive Decision Analysis approach because more parameters are needed to represent a given behavior. With the Heuristic Search approach, the heuristic, pruning and generating functions can be changed, but the changes necessary to obtain a particular behavior are not immediately drivable from it. The rank order (starting with the most flexible and transparent approach) is:

(1) Production Rules.

(2) Adaptive Decision Analysis.

(3) Elicited Probability.

(4) Heuristic Search.

b. Ability to Model Subjective Decision Criteria. The Adaptive Decision Analysis Model was developed specifically to handle subjective criteria and even can capture them automatically through training. With the Elicited Probability approach, subjective weights could be applied to the output but more research would have to be done to find a way to obtain them by automatic training. The Production Rules approach can capture subjective decision criteria of experts by embedding them in the productions themselves, but as with the Elicited Probability approach it takes a deliberate effort. The Heuristic Search approach cannot represent subjective criteria directly. The rank order, starting with the approach with the greatest ability for modeling subjective criteria is:

(1) Adaptive Decision Analysis.

(2) Elicited Probability.

(3) Production Rules.

(4) Heuristic Search.

c. Modeling Continuous Behavior. All of the approaches select discrete alternatives as their output; however, this decision making function can be separated from the actual calculation of the continuous variables. Thus, the decision model will select among several functions that will perform the actual trajectory calculation. Adaptive Decision modeling is the only approach which accepts continuous criteria as an input. The Elicited Probability and Adaptive Decision approaches give a value associated with the output which is continuously variable. Heuristic Search involves a traverse through a tree of discrete nodes. The criteria for selecting a node may be continuous but based on the state at the parent node which is a unique node. Production Rules combine discretely defined logical statements to select discrete outcomes. The ranking of the four approaches (best first) for this attribute are as follows:

(1) Adaptive Decision Analysis.

(2) Elicited Probability.

(3) Heuristic Search.

(4) Production Rule.

d. Modeling the Flow of Control. Traditionally, the flow of control in a simulation program was imbedded in the control structure of the implementation language. This method is always available as a last resort. By including a network of states in the production rule system the control flow can be made explicit. This avoids dependency on hard coded logic and makes the flow of control flexible, visible, and easy to modify. In the Heuristic Search approach the flow of control is rigidly built into the state space and the evaluation function, making changes more awkward. The Elicited Probability approach represents flow of control indirectly in that the behavior created has an orderly sequence. The Adaptive Decision Analysis addresses mainly the actual decision points and the flow of control has to be provided by external mechanisms. In rank order, starting from the most explicit and flexible flow of control is:

(1) Production Rules.

(2) Heuristic Search.

(3) Elicited Probability.

(4) Adaptive Decision Analysis.

e. Modeling AND and OR Conditions. Only the Production Rules approach explicitly models AND and OR input conditions. In order to model AND and OR conditions with the Elicited Probability approach, it is necessary to define an input state which is determined from logical conditions. Thus the AND's and OR's tend to be hard coded into the program which generates the input state. This may complicate the dependency problem. The Adaptive Decision Analysis approach has similar but more severe problems in dealing with AND and OR conditions. With heuristic search there would be a separate node for every possible combination of AND and OR conditions. One way to include AND and OR conditions would be to use a Production Rule approach to select from the other three approaches as sub-models (e.g., combine approaches). The rank order of the approaches is:

(1) Production Rules.

(2) Elicited Probabilities (distand second).

(3) Adaptive Decision Analysis.

(4) Heuristic Search.

f. Modeling Probabilities. The Elicited Probability approach generates probabilistic outputs and considers the probabilities of the input states, but explicit probabilities as input state variables are not modeled. With the Adaptive Decision Analysis approach, explicit probabilities as inputs can be handled, but the outputs are not probabilistic. With Production Rules, a probability may be associated with the output, input probabilities can be handled as with the Elicited Probability approach described above. Heuristic Search cannot handle probabilities directly. With the approaches which do not explicitly use probabilistic inputs, it is still possible to implicitly represent probabilistic inputs by expanding states into sub-states which have a probability as part of the state definition or breaking the probabilistic variables into several discrete ranges. This is clumsy, however, because it increases the size of the state space. The rank order of how well the four approaches model probabilities is:

(1) Adaptive Decision Analysis.

(2) Elicited Probability.

(3) Production Rules.

(4) Heuristic Search.

g. Conciseness of Representation. In a sense this is relative to the application. Each model could be the most concise for modeling a problem ideally suited for that approach. As a general measure of conciseness we can consider the number of parameters needed to represent behavior. Here, conciseness should not be confused with precision. We assume the more concise model has fewer parameters. The Adaptive Decision Analysis model represents behavior with only four to seven attribute weights, and it is necessary to calculate the same number of attribute levels for each action alternative. The Elicited Probability approach has a column of elicited probabilities for each alternative. The number of states considered in making a decsion. The Production Rule approach uses one or more logical structures for each action alternative. The truth or falsity of each operand must be evaluated. Heuristic Search has nodes corresponding to the number of possible combinations of input states. A Heuristic function and a pruning function must also be evaluated. The rank order of the approaches (most concise first) are as follows:

(1) Adaptive Decision Analysis.

(2) Elicited Probability.

(3) Production Rules.

(4) Heuristic Search.

h. Adaptiveness. Only the Adaptive Decision Analysis approach is adaptive in real time.

i. Dependencies of Input States. The Elicited Probability and Adaptive Decision Analysis approaches both assume independent input states. In both cases it is common practice to assume independence as a working assumption even when it is not strictly true. The methods of overcoming this problem are basically the same in both cases. The Production Rule and Heuristic Search techniques don't make an independent assumption and are therefore not affected by this problem. The rank order (most favorable first) of this attribute is:

(1) Production Rules and Heuristic Search.

(2) Elicited Probability and Adaptive Decision Analysis.

j. Auxiliary Payoffs. The auxiliary payoffs for each approach are as follows:

(1) Production Rules. Ability to explain reasoning leading to the selected action alternatives. Similarity of the representation to the human thought process.

(2) Adaptive Decision Analysis. Relative desirability of alternatives is available. A good collection of performance measures have been developed to go with this approach.

(3) Elicited Probabilities. A simulated intelligence expert can readily be made.

(4) Heuristic Search. This approach most directly simulates the process of "thinking ahead" or contemplating a sequence of possible moves and counter moves.

The rank order depends upon what auxiliary payoffs are appropriate for the particular application of the number of auxiliary payoffs available (largest number first):

(1) Adaptive Decision Modeling.

(2) Elicited Probability.

(3) Production Rules.

(4) Heuristic Search.

EVALUATION BY DEVELOPMENT ATTRIBUTES

a. Scenario Set-Up Time. With the Adaptive Decision Analysis approach, the instructor would act out the desired scenario in an operational setting and the behavior would be learned by the model. It may take a while for the model to converge, and consistent behavior is required for the model to train. Compared to other methods the time would be spent doing the normal task rather than struggeling with concepts which may be unnatural. The Elicited Probability approach requires that the instructor estimate a number of probabilities, view the resultant behavior, and make fine tuning changes. The Production Rules approach requires the specification of new or modified production relevant to the new behavior. The Heuristic Search approach requires changes to the heuristic function and possibly the node definition. This may be very difficult. The rank order (starting with the shortest time) is:

(1) Adaptive Decision Analysis.

(2) Production Rules.

(3) Elicited Probability.

(4) Heuristic Search.

b. Required Development and Implementation Time and Cost. This is a very difficult attribute to estimate. Each approach has aspects which are easy and those which are hard. The following rank order (quickest and cheapest first) is biased by previous experience Perceptronics has had with these models:

(1) Elicited Probability.

(2) Production Rules.

(3) Adaptive Decision Analysis.

(4) Heuristic Search.

c. Required Integration Time and Cost. Since integration difficulty is dependent on the amount of interfacing with the existing system, and the amount of interfacing is dependent on inputs, outputs, and data areas needed (which are roughly the same for all approaches), there is no basis at present for rating one approach above any other.

d. Vulnerability to Increase in Size of the State Space. The Adaptive Decision modeling approach is the least vulnerable to increase in the size of the state space. This is because a small number of attributes are used and their number does not increase. The only effect an increase in the size of the state space has is to make it more involved to calculate the attribute levels.

The Elicited Probability approach could also stay the same size as the state space size increases; however, it would probably be a practical necessity to increase the number of parameters or to put more model levels in the hierarchy.

The Production Rules and Heuristic Search approaches are potentially extremely vulnerable to increase in the size of the state space. In the case of the Production Rules approach, the number of additional Production Rules needed is likely to increase faster than the size of the state space. Heuristic Search is the most vulnerable, since its complexity increases as a combinatorial function of the size of the state space.

Here is the rank order (best first):

(1) Adaptive Decision Modeling.

(2) Elicited Probability.

(3) Production Rules.

(4) Heuristic Search.

EVALUATION BY PERFORMANCE ATTRIBUTES

a. Instructor Time Needed for Operation. Most of the factors affecting this are probably independent of the model itself except for those things discussed earlier under "Instructor time needed to set up problem scenario." There should probably be some interface programs which help transfer information and control from the instructor to the models, and information back to the instructor from the models.

b. Instructor Control. When the instructor assumes control from the model and vice versa, steps must be taken to insure smooth transitions. This means that all of the state variables needed by the models must be maintained. Also, the state changes created by the models must be updated in the existing software. Furthermore, when control is returned from the instructor to the automatic opponent, the specifics of the opponent state must be provided. This attribute is nearly independent of the model approach; however, in general there is greater difficulty with a more complicated model.

c. Required Computer Resources. Computer resources are a function of how detailed each decision is modeled. In general, the rank order (best first) is as follows:

(1) Adaptive Decision Analysis.

(2) Elicited Probability.

(3) Production Rules.

(4) Heuristic Search.

d. Capability for Including Performance Measures and Evaluation. A lot of development has gone into performance measures with the Adaptive Decision Analysis approach. Performance measures haven't been developed with the other approaches.

In the applications where performance measures have been developed the adaptive model was used to model the trainee, whereas, in the present application it is the instructor who is adaptively modeled. The power of the performance measures is derived from the adaptive model of the trainee. The reason for this is that the model of the trainee represents the current state of knowledge and skill of the trainee and performance measures are based on an analysis of model parameters. The performance measures made possible by modeling the trainee include the following:

(1) Decision consistency.

(2) Comparision of trainee values with expert values.

(3) Use of the trainee values to drive a simulation to compare the behavior created by the trainee's values to behavior created by other sets of values.

(4) Use the trainee values as they are to characterize the trainee.

In addition to performance measures based on adaptively modeling the trainee, the following measures have been developed:

(1) Evaluate trainee's skill at purchasing information.

(2) Compare the trainee's decision with the decision the expert would make (as indicated by an expert model with corresponding values).

(3) Measure decision time.

(4) Define a way to score the task elements such that a score results from each session (this measure is more powerful when used with an adaptive trainee model).

(5) Compile statistics on the trainee's frequency of making various decisions and compare these with expert statistics.

As envisioned previously, the adaptivity is used to model the instructor acting as the opponent--the trainee was not modeled. However, if good performance measures are important, it would be good to model the trainee as well. The algorithms to do this would be available in the software since they would have been developed to model the instructor. Much of the interfacing to model the traineee must also be done anyway. The main complication in adding the vapability to also model the trainee is the fact that to be valid the attribute levels should be displayed to the trainee. This changes the task as it appears to the trainee.

e. Real World Fidelity. Each model has the highest real world fidelity when applied in an area most suited for it.

Table 3 summarizes all the conclusions of this chapter in table form.

TABLE 3. MODEL EVALUATION BY DIFFERENT CRITERIA

| Modeling Attributes / Approach | Flexibility for modeling different opponent | Subjective decision criteria | Continuous behavior | Flow of control | AND and OR conditions | Probabilities | Conciseness | Adaptiveness | Dependencies | Auxiliary payoff | TOTAL |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Elicited Probability | 3 | 2 | 2 | 3 | 2 | 2 | 2 | 4 | 2 | 2 | 24 |
| Production Rules | 1 | 3 | 4 | 1 | 1 | 3 | 3 | 4 | 1 | 3 | 24 |
| Adaptive Decision Model | 2 | 1 | 1 | 4 | 3 | 1 | 1 | 1 | 2 | 1 | 17 |
| Heuristic Search | 4 | 4 | 3 | 2 | 4 | 4 | 4 | 4 | 1 | 4 | 34 |

TABLE 3 (CONTINUED). MODEL EVALUATION BY DIFFERENT CRITERIA

| Attributes | Scenario set-up | Development time and cost | Integration time and cost | Vulnerability to size increase | Instructor operation time | Instructor control | Computer resources | Performance measures | Real-world fidelity | TOTAL |
|---|---|---|---|---|---|---|---|---|---|---|
| Elicited Probability | 3 | 1 | — | 2 | — | — | 2 | 4 | — | 12 |
| Production Rules | 2 | 2 | — | 3 | — | — | 3 | 4 | — | 14 |
| Adaptive Decision Model | 1 | 3 | — | 1 | — | — | 1 | 1 | — | 7 |
| Heuristic Search | 4 | 4 | — | 4 | — | — | 4 | 4 | — | 20 |

# SECTION VI

# MODEL EVALUATION FOR SPECIFIC DECISIONS

## GENERAL

In the preceeding section each of the models were evaluated by a list of general attributes. In this section, we will present several specific decisions that a submarine CO has to perform and discuss the applicability of each model. It has to be kept in mind, however, that each decision does not stand alone and the control process that determines what has to be considered next, and what are the action options available there, is as important as the making of the decision itself.

For each of the decisions described below a simple description of the decision is given and then the various approaches are rank ordered according to their suitability.

## CONTACT DECISION

This is a protracted decision which dramatically influences the CO behavior. It has to be continued even after a positive contact is made to maintain the contact and to retract the "contact made" decision if new evidence indicate that the intitial decision was erroneous. Time enters the decision in that the probability of positive contact increases if a noise is repeated or is detected over a longer period. Additional considerations are the level of background sea noises at the given weather, the closeness to enemy sea operations, previous intelligence information, etc.

Some of these decision variables are intended to the model and some are inputs generated by the friend or the sea. The external signals have to be preprocessed and transformed into a variable acceptable by the decision model. A probabilistic output is desirable. A recommended rank order of the approaches is the following:

a. <u>Elicited Probability</u>. This model takes the available apriori probabilities and can update them incrementally as new evidence comes in. The output is compared to a threshold to decide whether to declare "contact" or not. The conditional probabilities in the transformation matrix represent an opponent's ability to diagnose noises and aggregate clues. These probabilities can be changed to simulate different opponent skill levels and even level of conservatism. Furthermore, a threshold change can be a simple mechanism to adjust the opponent's conservatism.

b. <u>Adaptive Decision Analysis</u>. The input consists of attributes of the noise state scaled such that a high attribute level means "contact." An expert's weights for each attribute are learned. An expected value is computed which represents the likelihood of contact. Contact is declared when this value exceeds a pre-set threshold.

c. Production Rules. The various considerations suggesting a contact can be incorporated into ascending states. Productions triggered by noise type and level can "vote" to move the state to one of increased probability of contact.

d. Heuristic Search. The only way heuristic would be appropriate is if the order of different noises was the predominate identifying characteristic.

THREAT DECISION

The threat decision is more an interpretation of external events than a classification of fixed patterns. It considers the mission, state of war, location relative to enemy, noises detected and number location and motions of potential threat. A simple breakdown of the different considerations follows:

| Type | Nationality | Location | Maneuver | Etc. |
|---|---|---|---|---|
| Nothing | Friendly | Near home | Indifferent | |
| Whale | Neutral | Open sea | Moving away | |
| Decoy | Unfriendly/peace | Near enemy | Moving toward | |
| Surface ship | Unfriendly/war | | Positioning for attack | |
| Nuclear sub | | | etc. | |

a. Elicited Probabilities. This approach has the flexibility to include all of the above factors. The apriori probabilities of the various output conditions (e.g., nature of the threat) can be biased according to the intelligence information which exists. The monitor's probability information is discretized and made part of the input state.

b. Production Rules. Because of the large number of contributing factors involved in this decision the Production Rules can be used to make an orderly decision. Each production handles a set of factors which lead to a meaningful conclusion, the conclusion can make other factors more relevant and new productions are triggered, etc. In general, the Production Rule approach is advantageous for formulating tactical assessment when interpretive consideration is dominate.

c. Adaptive Decision Analysis. With this approach a discriminant function is used for each possible interpretation. The model can handle naturally more than one plausible interpretation concurrently. The continuous time effect is awkward to represent as are apriori probabilities such as those derived from intelligence information.

d. Heuristic Search. In a situation where it is necessary to evaluate a sequence of moves and counter moves in order to determine whether a threat exists the Heuristic Search approach can be used. In this case a threat is a state that can lead to a set of terminal nodes which include some that are detrimental to the opponent. In other cases where "look ahead" is not relevant to the threat evaluation, the method would not be appropriate.

## MANEUVER SELECTION DECISION

The select maneuver decision is made under several different circumstances such as evade, attack, track, approach, etc. Each of these circumstances has a set of relevant maneuvers, one of which has to be selected. The selecting mechansim can be similar but with a different set of parameters. The details of the trajectory implementing the maneuver is performed by a lower level subroutine that is separate from the select decision. Such a subroutine can use a Monte Carlo method to specify the parameters of the trajectory guided by the intended objective of the maneuver.

a. Adaptive Decision Analysis. With this approach the relative desirability of each possible maneuver is computed. There is one discriminant function for each maneuver and a set of attributes across all maneuvers. This decision was used in Section IV to illustrate the Adaptive Decision approach.

b. Production Rules. Production Rules are excellent for imposing logical criteria on the maneuver selection decision. Probabilities can be attached to the Production Rules, but this increases their number.

c. Elicited Probabilities. By interpreting probabilities as relative desirability this model can be used to select maneuvers. Each contributing factor considered increases or decreases the desirability of the candidate maneuvers. The algorithm aggregates the individual desirabilities and the highest one is selected. The particulars of the trajectory are then calculated. This approach is able to handle situations where there may be a large number of possible maneuvers and many decision criteria.

d. Heuristic Search. This approach is not of use unless maneuver selection appears in the context of a series of maneuvers alternately selected by both sides.

## REFERENCES

Ahlers, R.H. "Intelligent Opponent Simulation." Personel communications. 1978.

Bond, N.A. and Rigney, J.W. "Bayesian Aspects of Troubleshooting Behavior." Human Factors, 1966, 8, 377-383.

Buchanan, B.G., and Smith, D.H. Computer Assisted Chemical Reasoning. Proc. III International Conference on Computers in Chemical Research, Education and Technology, 1976.

Coates, E.N. and McCourt, A.W. A Questionnaire-Based Analysis of Order-of-Battle Elements. U.S. Army Research Institute for the Behavioral and Social Sciences (Arlington, VA) Technical Paper 271, January, 1976.

Dawes, R.M. Shallow Psychology. Paper presented at the 11th Annual Carnegie Symposium on Cognition, Carnegie-Mellon University, April 24-26, 1975.

Dawes, R.M. and Corrigan, B. Linear Models in Decision Making. Psychological Bulletin, 1974, 81:95-106.

Duda, R.O. and Hart, P.E. Pattern Classification and Scene Analysis, Wiley-Interscience: New York, 1973.

Duda, R.O., Hart, P.E., Nilsson, N.J., and Sutherland, G.L. Semantic Network Representations in Rule-Based Inference Systems, to appear in Proc. Workshop on Pattern-Directed Inference Systems, 1977.

Edwards, W.D. "Dynamic Decision Theory and Probabilistic Information Processing." Human Factors, 4:59-73, 1962.

Freedy, A., Davis, K.B., Steeb, R., Samet, M.G. and Gardiner, P.C. Adaptive Computer Aiding in Dynamic Decision Processes: Methodology, Evaluation, and Application. Perceptronics, Inc. (Woodland Hills, CA) Final Technical Report PFTR-1016-76-8/30, Contract N00014-73-C-0286/NR 196-128, ARPA Order No. 2347, Code 455, Office of Naval Research, August, 1976.

Freedy, A., Weisbrod, R., and Weltman, G. Self-Optimization of Task Allocation in Shared Man/Computer Control. Proc. of the IEEE Conference on Decision and Control, Paper #FP5-3, San Diego, California, December 5-7, 1973.

Gardiner, P.C., and Edwards, W. Public Values: Multi-Attribute Utility Measurement for Social Decision Making. Proceedings of the 1975 IEEE International Conference on Systems, Man and Cybernetics, Paper No. THAM4-2.

Gilstad, D.W. and Fu, K.S. A Two-Dimensional Patterns Recognizing Adaptive Model of the Human Operator, Proceedings, Sixth Annual Conf. Manual Control, April, 1970.

Hagendorf, H., Kolbe W. and Sommerfeld, E. About the Solution of Combinatorial Problems with Problem Solving Methods, Proc. Fourth International Joint Conference on Artificial Intelligence, 1975.

Hayes, J.R.M. Human Data Processing Limits in Decision Making. In E. Bennett (Ed.), Information System Science and Engineering. Proceedings of the First Congress on the Information Systems Sciences, New York: McGraw-Hill, 1964.

Hill, F. and Huey, B. A Design Language Based Approach to Test Sequence Generation, Computer, June, 1977.

Kowalski, R. AND/OR Graphs, Theorem-Proving Graphs and Bi-Directional Search. Machine Intelligence, 6, 1971.

Leal, A. Adaptive Decisions in Ballistic Missile Defense, IEEE Trans., Sys., Man, Cybernetics. Vol. SMC-7, No. 6, May, 1977.

Lenat, D. The Ubiquity of Discovering. Artificial Intelligence, pp. 257-285, 1978.

Martelli, A. An Application of Heuristic Search Methods to Edge and Contour Detection. Comm. ACM Vol. 19, No. 2:73-83, February, 1976.

McKenry, J.M., Enderwick, T.P. and Harrison, P.C. A Subjective Value Approach to Information Utility. Human Factors, 1971, 13(6): 503-509.

Montanari, U. Heuristically Guided Search and Chromosome Matching. Artificial Intelligence, Vol. 1, No. 4, 1970.

Newman, J.R. Assessing the Reliability and Validity of Multi-Attribute Utility Procedures: An Application of the Theory of Generalizability. Social Science Research Institute, University of Southern California, Technical Report 001497-2-T, SSRI Research Report 75-7, Contract N00014-75-C-0487, ARPA Order No. 2105, Code 455, Office of Naval Research, July, 1975.

Nilsson, N.J. Learning Machines, New York: McGraw-Hill, 1965.

Nilsson, N.J. Problem-Solving Methods in Artificial Intelligence. McGraw-Hill: New York, 1971.

Powers, G. Non Numerical Problem-Solving in Computer-Aided Design in Computer-Aided Design. Vliestra, T. and Wiengla, R.T., (Eds.), 1973.

Raiffa, H. Preferences for Multiattributed Alternatives. Rand Corporation (Santa Monica, CA) Memorandum RM-5868-DOT/RC, April, 1969.

Rouse, W.B. Cognititive Sources of Suboptimal Human Prediction, Ph.D. Dissertation, MIT, Report DSR 70283-19, September, 1972.

Samet, M.G. Quantitiative Interpretation of Two Qualitative Scales Used to Rate Military Intelligence. Human Factors, 1975, 17(2):192-202.

Samet, M.G. Utility Assessment by "Interpolation Between Corners": An Application to the Evaluation of Drug Efficacy. U.S. Army Research Institute for the Behavioral and Social Sciences (Arlington, VA) Unpublished Manuscript, December, 1976.

Samet, M.G., Weltman, G., and Davis, K.B. Application of Adaptive Models to Information Selection in C3 Systems. Perceptronics (Woodland Hills, CA) Technical Report PTR-1033-76-12, Contract MDA 903-76-C-0241, Defense Advanced Research Projects Agency, December, 1976.

Shortliffe, E.H. Computer-Based Medical Consultations: MYCIN. New York, N.Y.: American Elsevier, 1976.

Slagle, J.R. Artificial Intelligence: The Heuristic Programming Approach, New York: McGraw-Hill, 1971.

Slovic, P. Consistency of Choice Between Equally-Valued Alternatives. Oregon Research Institute, ORI Research Bulletin No. 1, 14(11), 1975.

Sridharam, N. Search Strategies for the Task of Organic Chemical Synthesis. Third International Joint Conference on Artificial Intelligence, 1973.

Steeb, R., Chen, K. and Freedy, A. Adaptive Estimation of Information Values in Continuous Decision Making and Control of Remotely Piloted Vehicles. Perceptronics Technical Report PATR-1037-77-8. Woodland Hills, California, August, 1977.

Tversky, A. and Kahneman, D. Judgment Uncer Uncertainty: Heuristics and Biases. Science, 1974, 185:1124-1131.

Tversky, A., Slovic, P. and Lichtenstein, S. Subjective Optimality: Conceptual Issues, Empirical Results and Decision Aids, in Application of Research on Cognition to Man-Machine System Design, (Ed.) W. Edwards, Engineering Psychology Laboratory, University of Michigan, Report 010342-1-F, March, 1972.

## DISTRIBUTION LIST

Commanding Officer
Naval Training Equipment Center
Orlando, FL 32813 53

Defense Documentation Center
Cameron Station
Alexandria, VA 22314 12

All other addressees receive one copy.

Chief of Naval Operations
OP-987H
Attn: Dr. R. G. Smith
Washington, DC 20350

Chief of Naval Operations
OP-102X
Washington, DC 20350

Commander
Naval Air Systems Command
AIR-4135F
Washington, DC 20361

Chief of Naval Research
Code 455
800 N. Quincy Street
Arlington, VA 22217

Chief of Naval Research
Code 458
800 N. Quincy Street
Arlington, VA 22217

Commander
Training Command
Attn: Educational Advisor
US Pacific Fleet
San Diego, CA 92147

Chief of Naval Reserve
Code 332
New Orleans, LA 70146

CDR Charles Theisen
Commanding Officer
Naval Hospital Corps School
Great Lakes, IL 60088

Commander
Naval Weapons Center
Code 3154
Attn: Mr. Bob Curtis
China Lake, CA 93555

Director Educational Development
Academic Computing Center
US Naval Academy
Annapolis, MD 71402

Library, Code P201L
Navy Personnel Research and Development Center
Attn: M. McDowell
San Diego, CA 92152

Dr. Edward M. Huff
Deputy Chief, Man-Vehicle Systems Research Center
NASA Ames Research Center
Moffett Field, CA 94035

ASD SD24E
Attn: Mr. Harold Kottmann
Wright-Patterson AFB, OH 45433

US Air Force Human Resources Lab
AFHRL-TT (Dr. Rockway)
Technical Training Division
Lowry AFB, CO 80230

US Air Force Human Resources Lab
DOJZ
Brooks AFB, TX 78235

US Air Force Human Resources Lab
AFHRL-AD
Advance Systems Division
Wright-Patterson AFB, OH 45433

Commander
Navy Air Force, US Pacific Fleet
NAS North Island (Code 316)
San Diego, CA 92135

Chief, Research Office
Office Deputy Chief of Staff for Personnel
Dept of Army
Washington, DC 20310

DISTRIBUTION (Continued)

Commandant
US Army Field Artillery School
ATSF-TD-TS (Mr. Inman)
Ft. Sill, OK 73503

Dr. Jesse Orlansky
Institute for Defense Analyses
Science and Technology Division
400 Army-Navy Drive
Arlington, VA 22202

Commandant
USA Field Artillery School
Target Acquisition Dept.
Attn: Eugene C. Rogers
Ft. Sill, OK 73503

Seville Research Corp.
Suite 400 Plaza Building
Pace Boulevard at Fairfield
Pensacola, FL 32505

Dr. Edward A. Stark
Link Division
The Singer Co.
Binghamton, NY 13902