GEORGE MASON UNIVERSITY
SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING
DEPARTMENT OF SYSTEMS ENGINEERING AND OPERATIONS RESEARCH

The Fighter Aircraft's Autodefense Management Problem: A Dynamic Decision Network Approach

A Master of Science's Thesis by

Paulo Cesar G. da Costa

Spring 1999

THE FIGHTER AIRCRAFT'S AUTODEFENSE MANAGEMENT PROBLEM: A DYNAMIC DECISION NETWORK APPROACH

By

Paulo Cesar G. da Costa

A Thesis
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
the Requirements for the Degree
of
Master of Science
Systems Engineering

Committee:

___________________________________ Dennis M. Buede, Thesis Director

___________________________________ David A. Schum

___________________________________ Kathryn B. Laskey

___________________________________ Karla Hoffman, Department Chairman

___________________________________ Lloyd Griffiths, Dean, School of Information Technology and Engineering

Date: ____________________________

Spring 1999
George Mason University
Fairfax, Virginia

THE FIGHTER AIRCRAFT'S AUTODEFENSE MANAGEMENT PROBLEM: A DYNAMIC DECISION NETWORK APPROACH

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at George Mason University.

By

Paulo Cesar G. da Costa

Bachelor of Science: Academia da Força Aérea, Brazil, 1986

Director: Professor Dennis M. Buede
Department of Systems Engineering and Operations Research

Spring, 1999
George Mason University
Fairfax, Virginia

DEDICATION

This work is dedicated to my wife, Claudia, without whose unwavering support and encouragement this endeavor would not have been possible.

ACKNOWLEDGMENTS

Any effort with this level of complexity demands much more than a single person for being successful, no matter how hard this person works. Fortunately, I was lucky enough of being able to count on a distinguishable group of people, which devoted a precious amount of their time to this venture. I would like to acknowledge some of them.

I owe a great debt of gratitude to the Brazilian Air Force, particularly to Maj. Brig. Sérgio Candiota da Silva, for offering me the opportunity of pursuing this Master of Science Degree. Yet, a special recognition has to be made to Col. Luiz Antônio Rezende Lima, whose perseverance and idealism were the leading factors for the tremendous support and dedication he devoted to the accomplishment of this mission.

I am extremely grateful to Dr. Dennis M. Buede, a renowned intellectual whose impressive background and academic skills did not prevent him to be one of the most modest and gentle person I have ever met. Having him as a thesis director was not only an honor for any student but also a chance to learn how a great professor has to be.

The thesis committee was also highlighted by the presence of Dr. David A. Schum and Dr. Kathryn B. Laskey, and I would like to thank both for all the encouragement words I received and for their commitment to this thesis work. In addition, it would be an unforgivable distraction if I did not comment the kindness and attention dispensed to me by the chairman of the Center of Excellence in C^3I, Dr. Harry L. Van Trees, and from Sue Bocarosse, whose always-prompt assistance was exemplary.

My colleagues from the Adelfi Squadron deserve acknowledgement for their forgiveness during my preparation phase, I am sure that my many hours of extra study provided some overload to them. Here, it is hard to overemphasize my gratitude to Lt. Col. Ricardo Mendes, whose unconditional endorsement was decisive during that phase.

Finally, I own a great debt to my children, Paulo and Laura, for all their unquestioning love, and to my wife, Claudia, whose tenacity and tenderness kept me motivated even in the hardest times. To them, I am most grateful.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS OF SYMBOLS

AAA .... Anti-Aircraft Artillery. Includes all gun-based weapons employed against airborne targets (aircraft, helicopters, cruise missiles, etc).

BVR..................... Beyond Visual Range. BVR missiles can be launched with no visual contact from the pilot to the target.

EW....................... Electronic Warfare. Is related with all devices that make use of the electromagnetic spectrum.

FL200 .... Flight Level 200. Standard aviation nomenclature, the three digits after the letters "FL" represent altitude in hundred of feet. In the example, FL200 means 20,000 feet.

GPS .... Global Positioning System. Navigation aid that relies on a net of stationary orbit satellites to calculate its present position.

IFF .... Identification Friend of Foe. It's a transceiver that sends encoded signals (interrogations) and receives encoded positive answers from friend aircraft. If an aircraft is not able to answer positively, it will be considered as a foe.

INS: .... Inertial Navigation System. Extremely precise navigation equipment that senses aircraft's most tiny movements in order to calculate its present position.

IR......................... InfraRed. Part of the optical spectrum that is commonly used for missile guidance and detected by sensors in the aircraft.

Kt......................... Knots. Nautical Miles per hour.

LPI....................... Low Probability of Intercept RADAR. A radar that uses a broader frequency band in order to deceive RHAW systems, delaying or even denying its detection.

MAW................... Missile Approach and Warning system. Detects missile's launching by tracking disturbances on the IR spectrum.

nm........................ Nautical Miles.

Pk......................... Probability of Kill. The probability of a missile to destroy its target after being launched.

RHAW................. Radar Homing and Warning. Detects radar emissions and classifies their sources by means of its electronic signature.

SAM .... Surface-to-Air Missile.

ABSTRACT

THE FIGHTER AIRCRAFT'S AUTODEFENSE MANAGEMENT PROBLEM: A DYNAMIC DECISION NETWORK APPROACH

Paulo Cesar G. da Costa, M.S.

George Mason University, 1999

Thesis Director: Professor Dennis M. Buede

In spite of the increasingly powerful computational assets available in state-of-the-art Artificial Inteliggence research, there is still no existing technique capable of successfully cope with real-time, increasingly complex decision problems like the management of sensor resources for multisensor data fusion systems. In essence, previous efforts have consistently failed to provide a real-time running structure complex enough for modeling and solving these type of problems.

This research focuses on introducing and applying a recently developed technique that has an enourmous potential to fulfill that gap: the Dynamic Decision Networks. DDN is a theory that combines decision analysis and Bayesian Networks for real-time, dual control analysis of complex systems, and is also consistent with parallel processing architectures. The technique was conceived specifically to solve the problem of optimizing information gathering processes in complex stochastic scenarios, providing a

consistent graphically-oriented structure supported by both multi-objective decision theory and probability theory.

Dynamic Decision Networks are sets of interconnected influence diagrams and Bayesian Networks. The influence diagrams provide an automated decision facility for the DDN structure, while the Bayesian Networks act as inference engines that model complex interdependent stochastic processes.

As a case study for this thesis, the theory will be applied to the management of sensor and weapon resources allocation of a modern fighter aircraft. In the course of a mission, the fighter pilot has to deal with many uncertainties on the current situation of his aircraft, analyze the available information, options, values, and take various decisions that might compromise the success of the mission and also his survivability to it.

In this work, two automated systems for solving the case study were developed for analysis and comparison purposes, one using an established artificial intelligence technique and the other was based on the DDN approach. Also, a sensitivity analysis is performed in order to illustrate in practive the conclusions made during the development of both systems. As a central objective, the applicability of the DDN technique for solving complex, real-time problems is to be assessed through the findings of this research.

CHAPTER 1

INTRODUCTION

1.1 Background

The so-called Information Revolution is a phenomenon that can be easily observable under the light of most fields in human research. These chances account not only because of technological breakthroughs, such as the Internet, increasing computer capacity, or better communications, but also on sociological specifics, like current world's economic and commercial globalization. Also, never in the history of humankind have we had such a powerful control over increasingly huge amounts of information, and never has this control been so critical to achieve success in virtually every field of knowledge. One of the consequences of this scenario is the escalating interest in the discipline of decision analysis, which has been transformed in one of the fastest growing topics in today's scientific research, within a wide number application fields.

However, demand for better and faster decisions dramatically increased the complexity of decision processes in most applications. As a response to this challenging issue, many advances have been made in the Decision Analysis arena, one of these advances is the main subject of this work: the Dynamic Decision Networks technique.

1.2 Research Methodology

Loosely speaking, Dynamic Decision Networks is a technique that combines influence diagrams with Bayesian networks, reaching a synergy level that has the potential to overcome some of the weakness of those two techniques, while maintaining its powerful features.

The main goal of this work is to show the method's ability to deal with highly complex decision systems that evolve in time. The use of specialized techniques for the optimization process of this type of systems, such as Dynamic Programming, usually derives into rather complex and computationally intensive solutions, while the use of present Artificial Intelligence approaches commonly reach solutions that are only fairly adequate. To achieve that goal, a real life problem was chosen as a study case: the fighter aircraft's sensors and weapons allocation problem.

One of the main characteristics of the case study's domain is the timeliness of its data sources and decisions, which demands powerful capabilities from an automated system in order to deal with constantly evolving variables. As an example, in a static environment such a system would have to decide what action to perform in order to avoid being shot by an enemy interceptor, and among the answers we would expect actions like launching a missile and/or taking an evasive maneuver. However, since it is a dynamic environment, the correct answer will also vary with the time; in other words, if launching a missile is the right answer now, in a few seconds it would be a useless action.

In addition to the timeliness of the decisions, an automated system also has to deal with information sources that evolve with time as well. For instance, a given ground threat which emissions are being detected by the aircraft's Radar Homing And Warning receiver (RHAW) can stop its transmissions (intentionally or not) at any moment. Also, the automated system itself can decide to turn off the aircraft's radar in order to avoid being detected by the enemy, which will cause a decrease in the awareness on the enemy's aircraft movements. Here, the main topic is a tight, timing-oriented control over the value of the information provided to the system by its sensors.

In spite of the highly automated systems that inhabit modern fighter aircraft, the above-cited characteristics of this domain have prevented the construction of a complete automated decision system for the sensor and weapon management. For fulfilling this thesis' main objective, the Dynamic Decision Networks technique will be presented as a technique that provides both a strong, rigorous analytical capacity and the computational appropriateness for complex, time-evolving applications.

Here, the first research effort will be concentrated in analyzing the nuances of the problem, its uncertainties, and its complexity. Two scenarios were created for comparison purposes, exploring two different levels of uncertainty and complexity. Given those scenarios, the next step will be applying a current AI approach for solving the problem, while the main point will be introduced latter, by implementing the use of Dynamic Decision Networks to solve the same problem. Then, a comparison of the two approaches will be made in terms of the quality of the achieved solution and the computational complexity of its implementation.

An immediate problem that arises when following this comparison strategy is how to choose the appropriate approaches for analyzing the case study. This is not an easy issue as there are many optimization techniques available today, some of them using Artificial Intelligence concepts. Ultimately, comparing all major established techniques would be a Herculean and probably inefficient work.

A better way to achieve this goal is electing a taxonomy that covers all approaches, defining a partition of that taxonomy, and choosing representative elements for each partition. After this, the selected techniques would be applied to the scenarios in order to make inferences on their appropriateness for solving the problem.

For this thesis purpose, I choose the taxonomy proposed by Pearl (1988). In his work, the different optimization approaches were divided with respect to the way they deal with uncertainty. From the spectrum of techniques covered by this taxonomy, he perceived the existence of two main groups: Intensional and Extensional. The first treats uncertainty by using a syntax that consists of declarative statements about states of affairs, while the former assigns a generalized truth value for uncertainty to its formulas, in a way that the uncertainty of a formula is measured by the weighted sum of its subformulas. A considerable advantage in using this taxonomy resides in the fact that we can divide the spectrum of techniques into two main groups, and the DDN technique is the natural candidate for representing one of these groups. This will turn our focus to only one competing approach.

Other approaches in the intensional group that might have been considered here were the dynamic programming (Bellman, 1957) and the partially observable Markov decision processes (Lovejoy, 1991). However, the first requires too many computational resources for a real time decision system, while the second does not provide the DDN's compositional representation, requiring the states of word to be represented enumeratively, which is not the ideal solution for highly dynamic decision problems.

In addition, by selecting only two different techniques, each reflecting its respective group characteristics, we will be able to assess the advantages and disadvantages of both groups. After this evaluation, we would have covered practically most of current knowledge on the optimization techniques, since those two groups form a partition of all available techniques.

1.3 Overview of the Thesis

Initially, chapter 2 will describe the Dynamic Decision Networks technique, as developed by Dr. Dennis Buede (Buede, 1999), while also conveying the two case scenarios definition. Although not getting into too much detail, all necessary data one might need for a better understanding of the case study is presented in this chapter.

In the next step of the thesis, Chapter 3 will cover the Knowledge Systems technique, a successfully established method that fully represents the extensional group. In the first sections, a basic introduction on this approach will be conveyed, while some details of the resulting system are presented. The final section is dedicated to the analysis of the advantages and disadvantages perceived through the use of this approach for solving the case scenarios.

Chapter 4 presents the Dynamic Decision Networks technique, which represents the intensional group. The technique will be applied to the same scenarios that we have been analyzing before. The main point here is to infer on the technique's appropriateness in dealing with the problem, in contrast with the previous approach.

Chapter 5 is meant to present the sensitivity analysis that was made with the systems developed in the previous chapters. However, instead of a comprehensive report on all the procedures taken for that analysis, this chapter focuses on the main differences observed in practice. Therefore, only the most insightful observations will be commented and analyzed.

Chapter 6 aggregates the most important points studied in the previous chapters, while providing some comments on the trends of the DDN technique. Also, there are three appendices, which convey additional data regarding the use of both techniques for solving the case scenarios.

CHAPTER 2

UNDERSTANDING DYNAMIC DECISION NETWORKS

2.1 Theoretical Foundations of Decision Theory

Dynamic Decision Networks has its roots in the field of Decision Theory, when the already established concepts of multi-attribute decision analysis were combined with Bayesian Networks in order to achieve a sophisticated architecture that could be used as a powerful decision-making tool for solving complex, real-time decision problems. For a complete understanding of the technique it is necessary a minimum knowledge on the basic principles of modern Decision Theory and its main concepts, which is the main goal of this section.

According to Watson and Buede (1987), the primary modes of a decision are: *choosing* one option from a list, *allocating* a scarce resource(s) amongst competing projects, and *negotiating* an agreement with one or more adversaries. Even though these actions were already taken intuitively by the first socialized human beings, a formal approach to the decision process is an advent of more recent history.

Back to 1738, Daniel Bernoulli did the first documented work relating to what we call today Decision Theory (Watson and Buede, 1987). When explaining the solution of

St. Petersburg paradox, which is related to the proper evaluation of a particular gamble[1], he concluded that money was not an adequate measure of value. Instead, it's utility for each individual is non-linear, with a decreasing slope; marginal utility decreased as wealth increased.

However, it was just two centuries after Bernoulli's idea that modern decision theory was established with the axioms on Normative Decision Theory (Von Newman & Morgenstern, 1944, Savage, 1954, 1961). These axioms were then complemented by the concept of Multi-Attribute Value Analysis, a quantitative method that allowed the application of utility theory for a set of conflicting objectives.

As pointed out by Morgan (1990), although conceptually simple Multi-Attribute Value Analysis can become operationally complex. Given that a comprehensive coverage on it is not in the scope of this work, the subject will be only briefly mentioned in the following paragraphs. Nevertheless, a thorough treatment on it can be found in French (1986), Keeney and Haiffa (1976), and Watson and Buede (1987). For more specific examples of actual applications of the theory, the interested reader should study Bell (1977), Keeney (1980, 1992), Raiffa (1982), Sox (1988), and Corner (1991).

Decision Analysis, in my view, is the scientific approach that applies normative axioms to properly define a problem, establish a set of optimal solutions, and select the

[1] Roughly speaking, the paradox is related to the fact that even in a game with infinite expected value, every one would have a limit on the amount of money to put on risk.

one which gives the higher expected utility. However, many other definitions can be found in the rich literature that exists on this subject, but most of them will describe the process in terms of a collection of tools and techniques that support the decision-making process (Maxwell, 1994).

Most of the philosophical and mathematical background used in modern decision analysis can be found in the axioms of probability theory and utility theory. The first ensures a proper approach for dealing with uncertainty in the decision process. The former presents an effective way of carrying out the difficult task of quantifying the elements that drive the decision-maker(s) behavior (e.g., his preferences, values, and objectives). The following paragraphs are meant to provide a broad view on both fields, while assuring a proper basis for understanding the process of decision analysis.

2.1.1 Probability Theory

As Schum once pointed out (Schum, 1994), probability is one of subjects that have “a very long past but a very short history”. His explanation of that vision regards to the fact that an abstract notion of probability may be tracked back at least to Paleolithic times, in a sense that early cultures are known to have used artifacts for gambling or forecasting the future. In contrast, he adds, the first scientific works on what we now call probability theory have a more recent history, dating back to 400 years ago in the pioneer writings of mathematicians Blaise Pascal (1623-1662) and Pierre de Fermat (1608-1665).

Nevertheless, those 400 years of scientific research were not enough for bringing us with a common agreement on the philosophical foundations of probability theory. Instead, many different theories and axioms arose during this time, all failing to eliminate the discussion on what probability really is. The interest reader will find an excellent account on the historical development of the competing theories in Hacking (1975), while valuable comparative studies can be found in the works of Fine (1973), Weatherford (1982), and Cohen (1989).

In short, probability can be noted as he ratio of favorable cases to total, equipossible cases, the “Classical” approach (Laplace, 1951, Ball, 1960). Also, it may be perceived as a logical relation between statements of evidence and hypothesis, the “Logical” approach (Carnap, 1950, Keynes, 1957). Another view shows probability as the limiting value of a sequential occurrence of an event, the “Frequentist” approach (Von Mises, 1957). Finally, it can be considered as one's degree of belief on an unknown event, which is related to his state of knowledge about it, the “Subjectivist” approach (Ramsey, 1931, Savage, 1974, De Finetti, 1972).

As pointed out by Watson and Buede (1987), the Subjectivist school is the one to be adopted in the decision theory domain, since it provides the framework for helping an individual to model his perceptions of uncertainty, an essential tool for a decision-maker. However, I would add that there are so many approaches under the umbrella of the subjectivist school that we need to go further on our taxonomy effort in order to get our ideas organized.

Schum (1994) and Cohen (1989) pointed out that we could sort most interpretations of probability into two categories: Pascalian and non-Pascalian. The first categorization embodies all probability theories that conform to the following metric, presented by Kolmogorov (1956).

Given a nonempty basic space ***S*** of elementary outcomes, a class ***C*** of events, and a probability measure ***P*** defined for each event ***E*** in class ***C***:

(1) $P(E) \geq 0$ for any event ***E*** in class ***C***.
(2) $P(S) = 1$
(3) $P(E \cup F) = P(E) + P(F) - P(E \cap F)$

Non-Pascalian categories use different systems in order to quantify the degree of belief of a given event. Among this category we could include Shafer's *belief functions* (Shafer, 1976), and the *confidence factors* employed by Buchanan and Shortliffe (1985); both treated with more detail in chapter 3.

Dynamic Decision Networks, as a decision analysis technique, relies on the axioms of the Subjectivist school of probability, which are presented in the following paragraphs, extracted from Watson & Buede (1987):

(1) For any two uncertain events, ***A*** is more likely than ***B***, of ***B*** is more likely than ***A***, or they are equally likely.

(2) If A_1 and A_2 are any two mutually exclusive events, and B_1 and B_2 are any other mutually exclusive events; and if A_1 is not more likely than B_1, and A_2 is not more likely than B_2; then (A_1 and A_2) is not more likely than (B_1 and B_2). Further, if either A_1 is less likely than B_1 or A_2 is less likely than B_2, then (A_1 and A_2) is less likely than (B_1 and B_2).

(3) A possible event cannot be less likely than an impossible event.

(4) Suppose A_1, A_2, … is an infinite decreasing sequence of events; that is , if A_i occurs, then A_{i-1} occurs, for any ***i***. Suppose further that A_i is not less likely than some other event ***B***, again for any ***i***. Then the occurrence of all the infinite set of events A_i, ***I*** = 1,2,…, , is not less likely than ***B***.

(5) There is an experiment, with a numerical outcome, such that each possible value of that outcome, in a given range, is equally likely.

All the properties of the probabilistic system used by Bayesian Networks, Influence Diagrams, and DDN, can be derived from those axioms (i.e., the transitivity implied by axiom 4). As an example of that, an important property to be derived is the concept of conditional probability, which says that the probability of an event ***A*** given the occurrence of another event ***B*** equals the probability of both events occur concurrently divided by the probability of ***B*** occurring alone, that is:

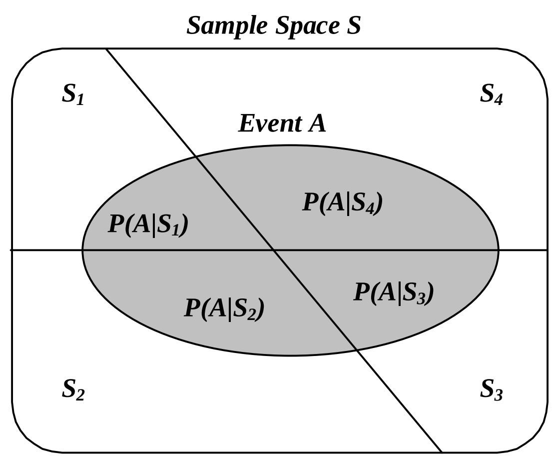
$$P(A \mid B) = \frac{P(A \cup B)}{P(B)}$$

This is a key concept for understanding the powerful capabilities of Bayesian Networks, Influence Diagrams, and Dynamic Decision Networks. However, before leaving this brief refresher on probability theory, it is necessary to go a step further to the concept of conditional probability and present two equations that are crucial for the notion of probabilistic inference: the *Law of Total Probability* and the *Bayes Rule*.

The Law of Total Probability, also known as multiplicative law (Page, 1988), computes the marginal probability distribution of one random variable by summing all possible values of a second random variable that is probabilistic dependent on the first (Buede, Forthcoming). Figure 1 explains the concept.

Law of Total Probability:

$$P(A) = \sum_{i=1}^{n} P(A \mid S_i)P(S_i)$$



In this particular case:
$P(A) = P(A|S_1) + P(A|S_2) + P(A|S_3) + P(A|S_4)$

Figure 1 Law of Total Probability

Bayes rule is the key idea for the rapidly evolving field of Bayesian Statistics, one of the most effervescent areas in current scientific research. Good introductory material on Bayesian Statistics can be found in works of Press (1989), Lee (1992), and Gelman (1995). Basically, the rule provides a mechanism of updating the probability of a random variable with data that is provided by another random variable, which is probabilistic dependent on the first. The standard format of Bayes Rule is:

$$\boldsymbol{P(B \mid A) = \frac{P(A \mid B)P(B)}{P(A)}}$$

P(B) is called prior probability of ***B***, as it reflects our belief in event ***B*** before obtaining information on event ***A***. Likewise, ***P(B|A)*** is the posterior probability of ***B***, and represents our new belief on event ***B*** after applying Bayes rule with the information collected from event ***A***.

2.1.2 Multi-Attribute Utility Theory

Deciding among available options imply in quantifying the uncertainty inherent of each alternative, which is supported by the theory presented in the previous section, and establishing the outcomes of each option, which will depend on how each option is valued by the decision-makers.

After the first ideas of Bernoulli (refer to page 7), it became clear the need of a systematic approach for assigning values for the options of a decision process. As pointed out by Watson and Buede (1984), one can axiomatize expected utility in several ways; but Savage (1954) gave us the most comprehensive account available:

(1) The decision-maker can place all possible lotteries in a preference order. That is, for any two lotteries, $\boldsymbol{A}$ and $\boldsymbol{B}$, he can state which he prefers, or whether he is indifferent between them; furthermore, if he prefers $\boldsymbol{A}$ to $\boldsymbol{B}$ and $\boldsymbol{B}$ to $\boldsymbol{C}$, then he prefers $\boldsymbol{A}$ to $\boldsymbol{C}$.

(2) If the decision-maker prefers $\boldsymbol{P_1}$ to $\boldsymbol{P_2}$ and $\boldsymbol{P_2}$ to $\boldsymbol{P_3}$, where $\boldsymbol{P_1}$, $\boldsymbol{P_2}$, $\boldsymbol{P_3}$ are three prizes, then there is some probability $\boldsymbol{p}$, between 0 and 1, such that he is indifferent between getting $\boldsymbol{P_2}$ for sure and a lottery giving him $\boldsymbol{P_1}$ with probability $\boldsymbol{p}$ and $\boldsymbol{P_3}$ with probability (1-$\boldsymbol{p}$).

(3) If the decision-maker is indifferent between two prizes, $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$, and the decision-maker prefers $\boldsymbol{P_1}$ to $\boldsymbol{P_2}$, then he prefers the lottery with the higher chance of winning $\boldsymbol{P_1}$.

(4) Suppose lottery $\boldsymbol{A_i}$ gives prize $\boldsymbol{P_1}$ with probability $\boldsymbol{P_i}$, and a prize $\boldsymbol{P_2}$ with probability (1-$\boldsymbol{p_i}$), for $\boldsymbol{i}$= 1, 2, 3; and suppose lottery $\boldsymbol{B}$ gives entry to lottery $\boldsymbol{A_2}$ with probability $\boldsymbol{q}$, and entry to lottery $\boldsymbol{A_3}$ with probability (1-$\boldsymbol{q}$); then the decision-maker is indifferent between lottery $\boldsymbol{A_1}$ and lottery $\boldsymbol{B}$ if, and only if;

$$\boldsymbol{p_1} = \boldsymbol{q}\boldsymbol{p_2} + (1-\boldsymbol{q})\boldsymbol{p_3}$$

Baumol (1972) showed that, if an individual can subscribe to these properties of his preferences for lotteries, a utility function must exist and that the rational action is to choose the option with the highest expected utility.

Multi-Attribute Value Analysis, a powerful extension for those axioms, provides a quantitative approach for gathering together the stakeholders' preferences over conflicting objectives to find the option with the highest value when all objectives are considered (Buede, Forthcoming).

That quantification will bring a set of value functions, one for each objective, that are usually monotonic (non-monotonic curves usually indicates two or more objectives being represented by the same function) and commonly represented by an exponential function (Kirkwood, 1996). Eliciting value curves is a science on its own, for a comparison among different elicitation techniques, refer to Buede (Forthcoming).

2.2 The Inference Engine

In the Dynamic Decision Networks scheme, data coming from outside the system are evaluated before feeding the decision nodes. This evaluation is made by a set of Bayesian Networks, which will be briefly introduced in this section. Initially, the necessary basic notions of graph theory will be presented in 2.1.1. Then, an explanation on the most relevant properties of Bayesian Networks is the central subject in 2.1.2.

2.2.1 Graph Theory

The first publication on what is known today as Graph Theory is attributed to the Swiss mathematician Leonard Euler (1707-1783); who wrote a paper in 1736 presenting a solution for the Königsberg Bridge problem[2]. However, it was not until 1930 that we could perceive a sustained and intense interest in graph theory as a mathematical discipline. One reason for such renascence was the realization that many complex and wide-range problems found in diverse fields of the modern society could be easily modeled and solved with graphs (Balakrishnan, 1991).

---

[2] The people from Königsberg, an East Prussian town situated by the Pregel River (now a Russian city called Kaliningrad), asked Euler to determine whether a person would walk through town crossing all its seven bridges without crossing any bridge twice. Euler used graph techniques for proving that such a walk was impossible.

The advantages of graph theory's mathematical elegance and modeling simplicity were also employed in the developing field of Decision Analysis, through techniques like Bayesian Networks and Influence Diagrams. A comprehensive explanation on the graph theory is far beyond the scope of this work, the prospective reader will find many literature on this subject (e.g. Balakrishnan, 1991, Lipschutz, 1976). For this work purposes, the basic set of definitions provided by Maxwell (Maxwell, 1994), which is described below with few modifications, will cover all the background needed for understanding the techniques explained in the remainder of this chapter.

Definition 2.1 **Graph:** A graph is a set of two finite sets, a non empty set $\boldsymbol{N}$ together with a (possibly empty) set $\boldsymbol{A}$, that is disjoint from $\boldsymbol{N}$ and whose elements contain two element subsets of distinct elements of $\boldsymbol{N}$.

The primitive elements of $\boldsymbol{N}$ are referred to as nodes ($\boldsymbol{n}$), and the elements of $\boldsymbol{A}$ as arcs ($\boldsymbol{a}$). Nodes contain data that describe events, or functions that manipulate the data from adjacent nodes. Arcs indicate a relationship between nodes, and two nodes connected by an arc are said to be adjacent. Other literature concerned with graph theory and graph based operations may refer to these components differently; nodes are often referred to as vertices or points, and arcs are sometimes referred to as edges or lines.

Definition 2.2 **Subgraph:** A graph $\boldsymbol{H}$ is a *subgraph* of $\boldsymbol{G}$ if $\boldsymbol{N(H)} \subseteq \boldsymbol{N(G)}$ and $\boldsymbol{A(H)} \subseteq \boldsymbol{A(G)} \cap \boldsymbol{N(H)} \times \boldsymbol{N(H)}$.

Definition 2.3 **Undirected Graph:** An undirected graph is a graph in which the set $\boldsymbol{A}$ consists of a set of pairs of distinct elements of $\boldsymbol{N}$. The set $\boldsymbol{A}$ indicates a reflexive binary relation between $\boldsymbol{n_i}$, $\boldsymbol{n_j}$. These pairs are normally depicted as nodes connected by an undirected edge, or solid line.

Definition 2.4 **Directed Graph:** A directed graph is a graph in which the set $\boldsymbol{A}$ consists of a set of **ordered pairs** of distinct elements of $\boldsymbol{N}$.

These ordered pairs are normally represented in visually depicted graphs as an arrow in which the first node of the pair resides at the tail of the arrow, while the second node is at the head of the arrow. The element of set $\boldsymbol{A}$ denoting the connection between the nodes of an ordered pair ($\boldsymbol{n_i}$, $\boldsymbol{n_j}$) can be displayed as $\boldsymbol{a_{i,j}}$.

Definition 2.5 **Chain:** A chain from nodes $\boldsymbol{n_1}$ to $\boldsymbol{n_m}$ exists if there is a set of nodes $\boldsymbol{n_1}, \boldsymbol{n_2} \dots \boldsymbol{n_m}$ together with arcs (directed of undirected) $\boldsymbol{a_{1,2}}$ or $\boldsymbol{a_{2,1}}$, $\boldsymbol{a_{2,3}}$ or $\boldsymbol{a_{3,2}} \dots \boldsymbol{a_{m-1,m}}$ or $\boldsymbol{a_{m,m-1}}$.

Definition 2.6 **Path:** A path from nodes $\boldsymbol{n_1}$ to $\boldsymbol{n_m}$ exists if there is a set of nodes $\boldsymbol{n_1}, \boldsymbol{n_2} \dots \boldsymbol{n_m}$ together with directed arcs $\boldsymbol{a_{1,2}}, \boldsymbol{a_{2,3}} \dots \boldsymbol{a_{m-1,m}}$.

A node $\boldsymbol{n_m}$ is *reachable* from $\boldsymbol{n_1}$ if a path exists from $\boldsymbol{n_1}$ to $\boldsymbol{n_m}$. A path is said to be *nontrivial* if $\boldsymbol{n_1} \neq \boldsymbol{n_m}$.

Definition 2.7 **Directed Acyclic Graph (DAG):** A directed graph is *acyclic* when there is no path from any node $\boldsymbol{n_i}$ to itself.

Bayesian Networks and Influence Diagrams are directed acyclic graphs.

Definition 2.8 **Singly Connected DAG:** A DAG $\boldsymbol{G} = (\boldsymbol{N},\boldsymbol{A})$ is *singly connected* if for every $\boldsymbol{n_i}, \boldsymbol{n_j} \in \boldsymbol{N}$ there is at most one chain between $\boldsymbol{n_i}$ and $\boldsymbol{n_j}$.

Definition 2.9 **Multiply Connected DAG:** A DAG $\boldsymbol{G} = (\boldsymbol{N},\boldsymbol{A})$ is *multiply connected* if there is at least one pair of nodes $\boldsymbol{n_i}, \boldsymbol{n_j} \in \boldsymbol{N}$ that possess more than one chain between $\boldsymbol{n_i}$ and $\boldsymbol{n_j}$. That is the graph is not singly connected.

Definition 2.10 **Connected DAG:** A graph is said to be connected if all nodes $\boldsymbol{n_i}$ are either singly or multiply connected.

Definition 2.11 **Parents:** Nodes of a set $\boldsymbol{P'}$ are called *parents* of node $\boldsymbol{x}$, and represented as $\boldsymbol{P'(x)}$, if there is an arc from every node in $\boldsymbol{P'}$ to node $\boldsymbol{x}$. Similarly, a set $\boldsymbol{P'(X)}$ are called the parents for a set of nodes $\boldsymbol{X}$ if $\boldsymbol{P'(X)}$ contains all nodes that are parents of any element of $\boldsymbol{X}$. That is $\boldsymbol{P'(X)} = \{\cup_i \boldsymbol{P'(X)} \mid \boldsymbol{x_i} \in \boldsymbol{X}\}$. Parent nodes are sometimes referred to as the conditional predecessors of a node.

Definition 2.12 **Children:** Nodes in a set $\boldsymbol{Q'}$ are called *children* of node $\boldsymbol{x}$, and represented as $\boldsymbol{Q'(x)}$, if there is an arc from $\boldsymbol{x}$ to every node in $\boldsymbol{Q'}$. This can also be extended to include the children of a set of nodes ***Q'(X)***. Children are sometimes referred to as conditional, of immediate, successors.

Definition 2.13 **Predecessors:** Nodes in a set ***P*** are called *predecessors* of node ***x***, and represented as ***P(x)***, if there is a directed path from every node in ***P*** to node ***x***. Similarly, a set ***P(X)*** are called the predecessors for a set of nodes ***X*** if ***P(X)*** contains all nodes that are predecessors of any element of ***X***.

Definition 2.14 **Successors:** Nodes in a set ***Q*** are called *successors* of node ***x***, and represented as ***Q(x)***, if there is a directed path from node ***x*** to every node in ***Q***. Similarly, a set ***Q(X)*** are called the successors for a set of nodes ***X*** if ***Q(X)*** contains all nodes that are successors of any element of ***X***.

Definition 2.15 **Forest:** A DAG is called a *forest* if every node $\boldsymbol{n_i} \in \boldsymbol{N}$ has at most one parent $[\boldsymbol{P'(n_i)} \leq 1]$.

Definition 2.16 **Tree:** A forest is called a *tree* if there is exactly one node $\boldsymbol{n_i} \in \boldsymbol{N}$ that has no parents $[\boldsymbol{P'(n_i)} = \varnothing]$.

2.2.2 Bayesian Networks

Bayesian Networks are direct acyclic graphs (DAG) whose nodes represent random variables, while its edges show the joint probability distribution amongst these random variables. The technique has been successfully used for creating consistent probabilistic representations of uncertain knowledge in diverse fields like medical diagnosis (Spiegelhalter, 1989), image recognition (Booker, 1988), search algorithms (Hansson, 1989), and many others. Wellman (1995) provides a detailed list on current applications of Bayesian Networks.

One of the most important features of Bayesian Networks is the fact that it provides an elegant mathematical structure for modeling complicated relationships among random variables while keeping a relatively simple visualization of these relationships. Figure 2 gives three simple examples of joint probabilities distributions among three random variables being depicted through the Bayesian Network technique.

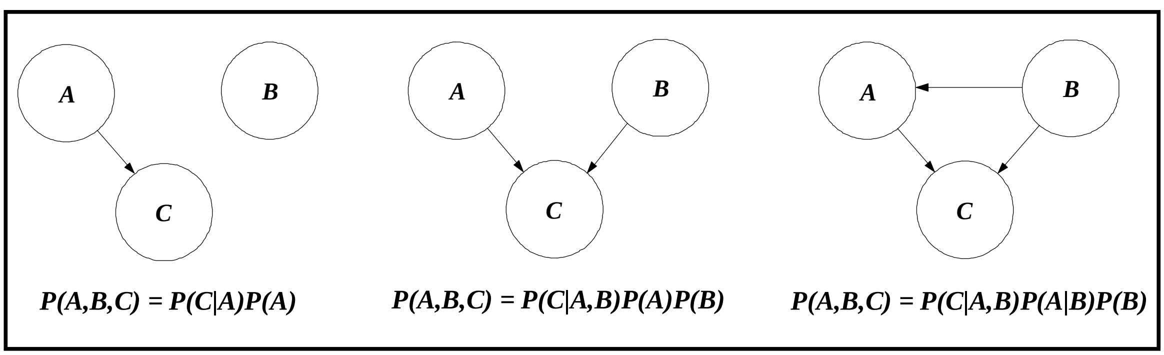


Figure 2 Sample Relationships Among Three Random Variables

To emphasize the communication power of this representation, a probabilistic unsophisticated expert looking only to the written equations below the pictures will have to think twice before making any conclusion on the relationships among events ***A***, ***B***, and ***C***. On the other way, looking only to the pictures the same expert will instantaneously perceive that in the leftmost picture, for example, event ***B*** is independent of events ***A*** and ***C***, and event ***C*** is dependent on event ***A***. It is pretty straightforward to perceive that increasing the number of nodes, in other words increasing the model's complexity, this communication power will become an invaluable characteristic.

Another important issue on Bayesian Networks, is the ability of updating the beliefs of each random variable by a bi-directional propagation of new information through the whole structure. This can be achieved by an algorithm proposed by Pearl (1988) that fuses and propagates the impact of new evidence providing each node with a belief vector consistent with the axioms of probability theory. Figure 3 depicts the bi-directional propagation scheme.

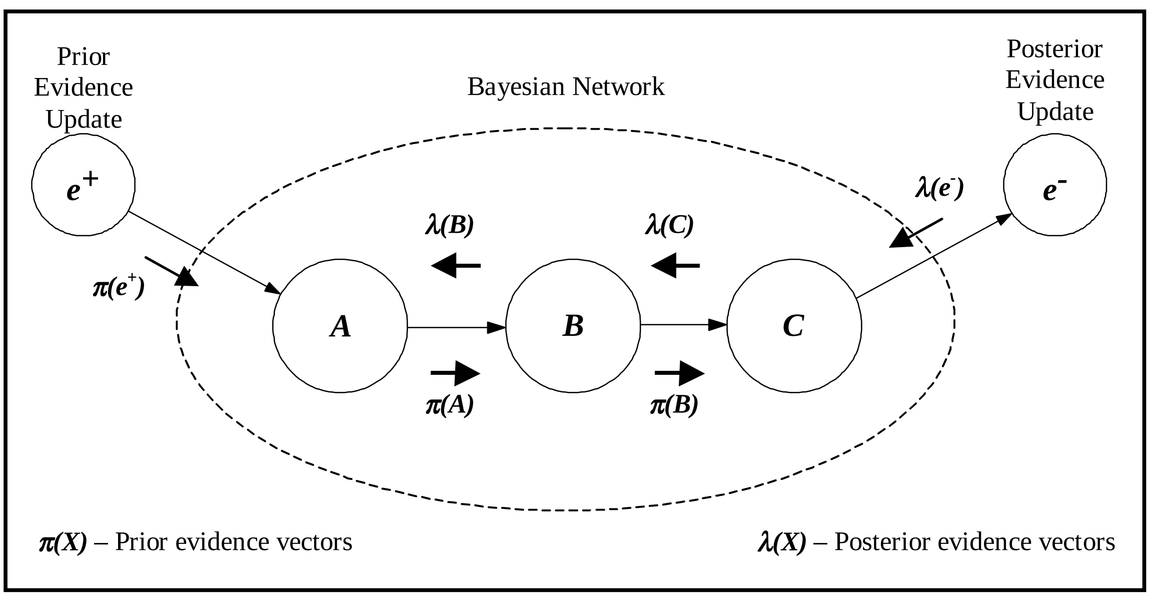


Figure 3 Bi-Directional Propagation in a Bayesian Network

Roughly speaking, information can be inserted in Bayesian Networks through a data updating in the prior probabilities or in the posterior probabilities. In the first case, the new data will flow via a $\boldsymbol{\pi}$ row vector (prior evidence vector), while in the former case data will flow via a $\boldsymbol{\lambda}$ column vector (posterior evidence vector). Both vectors update the node belief (say node ***B***) by the equation

$$\boldsymbol{Bel(B)} = \boldsymbol{P(B|e^{+},e^{-})} = \alpha\,\pi(\boldsymbol{B})^{\boldsymbol{T}} \bullet \lambda(\boldsymbol{B})$$

where "$\boldsymbol{\alpha}$" is a normalizing constant, and " • " means term by term multiplication (inner or dot product). The resulting column vector is the new belief of node ***B***, clearly, vector ***Bel(B)*** will have as many elements as the number of states of the random variable depicted by node ***B***.

Nodes of a Bayesian network have different number of states, which will reflect in the number of elements each $\boldsymbol{\pi}$ or $\boldsymbol{\lambda}$ vectors will have. After receiving a $\boldsymbol{\pi}$ vector with updated information from a parent node (say ***A***), node ***B*** will send its own $\boldsymbol{\pi}$ vector to its children nodes. The equation used in node ***B*** for creating its $\boldsymbol{\pi}$ vector is

$$\pi(\boldsymbol{B}) = \sum_{\boldsymbol{A}} \boldsymbol{P}(\boldsymbol{B} \mid \boldsymbol{A}, \boldsymbol{e}^{+}) \bullet \boldsymbol{P}(\boldsymbol{A} \mid \boldsymbol{e}^{+}) = \sum_{\boldsymbol{A}} \boldsymbol{P}(\boldsymbol{B} \mid \boldsymbol{A}) \bullet \pi(\boldsymbol{B}) = \pi(\boldsymbol{B}) \otimes \boldsymbol{M}_{\boldsymbol{B|A}}$$

where " ⊗ " means vector multiplication (or congruent product), and $\boldsymbol{M_{B|A}}$ is the likelihood matrix, or conditional probability distribution matrix between nodes ***B*** and ***A***.

When receiving a $\boldsymbol{\lambda}$ vector with updated information from a child node (say node ***C***), node ***B*** will send its own $\boldsymbol{\lambda}$ vector to its parent nodes. The formula used in node ***B*** for creating its $\boldsymbol{\lambda}$ vector is

$$\lambda(\boldsymbol{B}) = \sum_{C} \boldsymbol{P}(\boldsymbol{e}^{-} \mid \boldsymbol{B}, \boldsymbol{C}) \bullet \boldsymbol{P}(\boldsymbol{C} \mid \boldsymbol{B}) = \sum_{C} \boldsymbol{P}(\boldsymbol{e}^{-} \mid \boldsymbol{C}) \bullet \boldsymbol{P}(\boldsymbol{C} \mid \boldsymbol{B}) \Rightarrow$$

$$\Rightarrow \sum_{C} \lambda(\boldsymbol{C}) \bullet \boldsymbol{P}(\boldsymbol{C} \mid \boldsymbol{B}) = \boldsymbol{M}_{C|B} \otimes \lambda(\boldsymbol{C})$$

where the resulting column vector $\boldsymbol{\lambda(B)}$ is then transmitted to parent nodes.

However, a node usually has multiple children, which means it may receive different λ vectors. The node internal algorithm must be able to deal with these vectors concurrently, as more than one node can send λ vectors at the same time. This problem was already solved by Pearl's algorithm, Figure 4 is an adaptation of the one he used for explaining his algorithm in page 168 (Pearl, 1988).

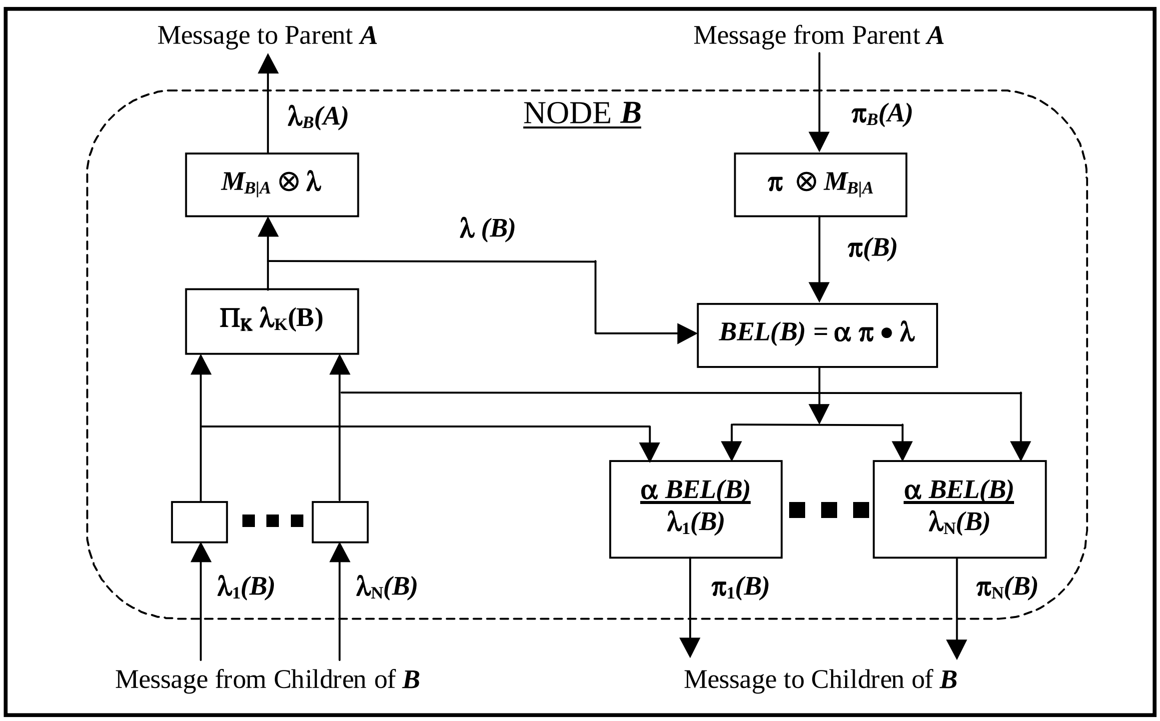


Figure 4 Internal Structure of a Single Node Processor

As an example illustrating the effectiveness of the algorithm, let's imagine the case in which node ***B*** has two children (say nodes ***C*** and ***D***). When a λ vector is received from node ***C***, that information will update node ***B***'s belief vector and this new belief vector will be sent to parent nodes (as λ vectors), and to children nodes (as π vectors). However, sending a π vector back to node ***C*** would generate a new update in node ***C*** with the same data it sent before, thus creating a loop. The division that happens in the lower left part of the diagram prevents this unwanted characteristic. The p message that is sent to children nodes is ***BEL(B)*** divided by the respective children node λ vector, eliminating the possibility of double counting the information. In our example, node ***D*** will receive a π vector from node ***B*** that has the information sent by node ***C*** (which means that node ***C***'s new information is propagated to ***D***). In contrast, node ***C*** will receive a π vector that is divided by λ_C so the information already sent will not be double counted.

In the present work, the BN's inherent ability as inference engines is explored in Chapter 4's DDN system for analyzing the aircraft sensors' information and assessing each eventual perceived track's relevant probability. For more information on Bayesian Networks the prospective reader will find other comprehensive coverage in many current literature, such as Pearl (1988), Neapolitan (1990), Oliver (1990), Charniak (1991), or Jensen (1996).

2.3 Modeling Decisions with Graphs

The idea behind the representation of hard problems through graphical tools is to provide an intuitive approach for the formulation of the underlying probabilistic model, while also providing a formal mathematical and statistical for analyzing that representation.

Initially, decision trees were the most popular technique for modeling decisions, given the relative simplicity in which a hard decision could be divided in a set of smaller, easier-to-solve problems; most literature covering decision analysis will have a topic on decision trees (e.g. Winston, 1993, Clemen, 1996). Influence diagrams became to overcome this preference, since it avoids some of the shortcomings of decision trees (see Matheson, 1988, for a good explanation on that). Nevertheless, these techniques can be considered isomorphic, since a properly constructed decision tree can be converted to an influence diagram and vice versa.

Dynamic Decision Networks extends the decision analysis capabilities of influence diagrams, overcoming some of its limitations. Section 2.3.1 conveys a brief presentation of influence diagrams, while section 2.3.2 carries an explanation of the ideas developed by Dr. Dennis Buede on Dynamic Decision Networks (Buede, 1999).

2.3.1 Influence Diagrams

The initial works on influence diagrams were focused in developing a computer-aided modeling tool for representing decision analysis problems (Miller, 1976, Howard, 1981). Olmsted (1983) proposed the idea of manipulating that tool in order to not only represent, but also solving decision analysis problems, which could be achieved by the mathematical background developed by Shachter (1986, 1988, and 1990).

Since these pioneering works, much have been written on influence diagrams. Buede (Forthcoming) synthesizes the characteristics of a well-formed influence diagram:

(1) The influence diagram is a DAG.
(2) Each decision of chance node is defined in terms of mutually exclusive and collectively exhaustive states.
(3) There is a joint probability distribution that is defined over the chance nodes in the diagram that is consistent with the probabilistic dependence defined by the arcs.
(4) There is at least one directed path that begins at the originating or initial decision node, passes through all the other decision nodes, and ends at the value node.
(5) There is a proper value function defined at the value node.
(6) There are proper functions defined for each deterministic node.

Once well elaborated, an influence diagram can be analyzed by a combination of graph and mathematical operations for determining the optimal decision strategy to be taken. The operations that allow a complete analysis of any influence diagram are evidence absorption, deterministic absorption, null reversal, arc reversal and deterministic propagation. More detailed information on these operations can be found in Shachter (1986, 1988, and 1990).

Basically, influence diagrams share the same graph theory principles as Bayesian networks; that is, the same representation of a set of random variables and its joint probability distribution. However, unlike the unique set of random variables found in Bayesian networks, influence diagrams contains four different sets of random variables, one for each type of node that can be present in the model: value nodes, decision nodes, chance nodes, and deterministic nodes.

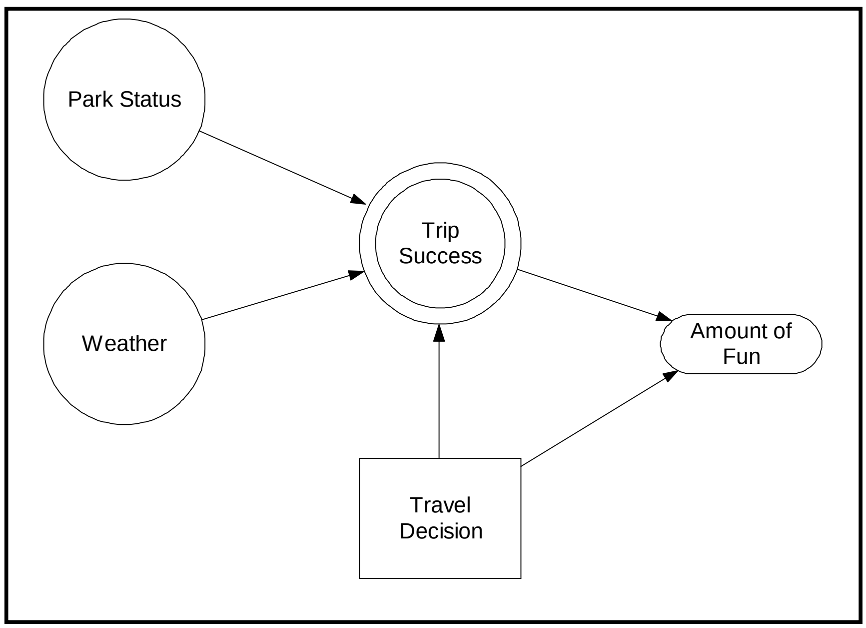


Figure 5 Weekend Trip Influence Diagram

Figure 5 shows a simple influence diagram that contains all four types of nodes. It depicts a simple decision of whether to travel to a given park or to stay at home in a hypothetical weekend. The value node (rounded rectangle) represents the family goal of having fun during the weekend, which will depend on the decision of either going to the park or staying at home (the rectangle) and on the success of an eventual trip to the park.

The success of a trip to the park is a deterministic node (a double oval), since it is a function of the weather's behavior and of the park status (open or closed, assuming as unknown to the decision-maker) in that weekend. These last two nodes are value nodes (ovals), which means that its outcomes are unknown at least until the decision is taken. Finally, the decision will also have an impact on the trip success, since a “no-go” decision will bring the deterministic node's value to zero.

In addition to the different types of nodes, influence diagrams also have inherent characteristics with respect to its arcs. Depending upon which nodes they connect, arcs will have different meanings, some are showed in Figure 6.

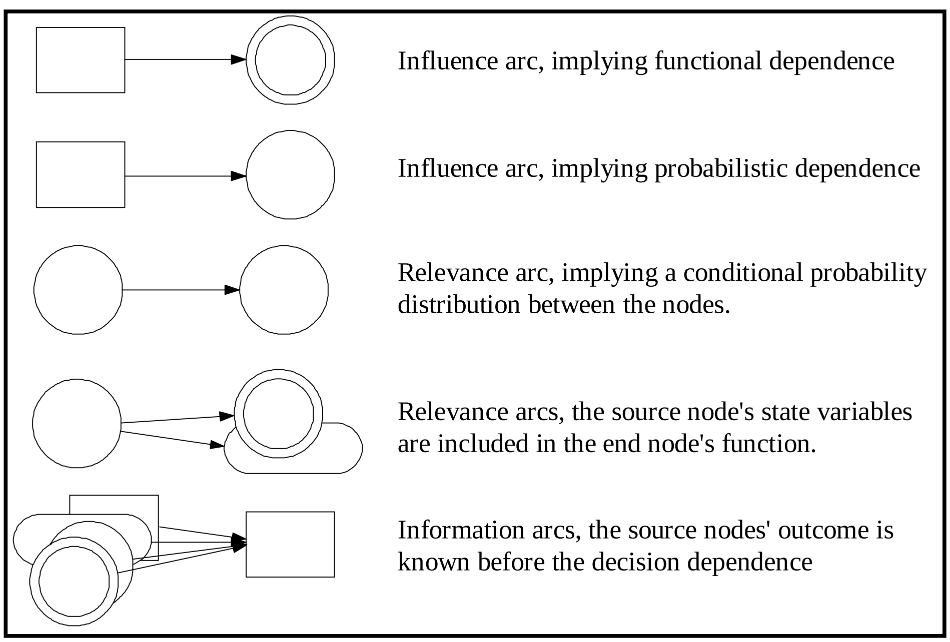


Figure 6 Meaning of Arcs in an Influence Diagram

Influence and relevance arcs have a straightforward applicability for elaborating the decision model, as they represent directly the complex intricacies of the interrelationship among the model's nodes. In contrast, informational arcs have a more subtle function, as they are used for determining the value of information, a key factor in the decision analysis process. Insightful discussions about the value of information in uncertain decision problems can be found in Ziemba (1975) and Hilton (1981), while Matheson (1990) shows the advantages of using influence diagrams as modeling tools, with respect to its ability of capturing the value of information in the modeling process.

2.3.2 Dynamic Decision Networks

The main idea under the concept of dynamic decision networks is to optimize the synergy between Bayesian networks and influence diagrams, focusing in the advantages of both techniques while eliminating their weaknesses. Bayesian networks have already proved to be a strong inference engine, and influence diagrams is a growing subject in the decision analysis theory. Dynamic decision networks uses the first to analyze the uncertainty level of separate elements of the problem, and the second to proceed the data fusion of the information gathered by the separate Bayesian networks and to define an optimal set of actions which will maximize the end objective.

Influence diagrams were already successfully applied to real-time applications, as in the milling machine monitoring and control system described by Agogino and Ramamurthi (Agogino, 1990). However, these applications did not cover multiple elements in quantities that varies with time, like in the pilot's decision problem; also, most did not consist of applications controlling more than one system simultaneously, where the value of information plays a vital role.

Another interesting related real-time application is provided by the multi-target tracking system presented by Kenley and Casaletto (Kenley, 1990), which uses Bayesian networks (named there as Gaussian influence diagrams) for identifying different tracks in a dense environment. According to that paper, this technique proved to be superior to previous approaches, which uses independent entity modeling. Yet, that application is focused in the probabilistic inference provided by the Bayesian networks, and there is no dynamic decision components modeled. In other words, there are no decisions taken by the model in time step “t” that could change the way the system behaves in time “t+1”.

When envisioned by Dennis Buede in the late eighties, DDNs were intended to solve complex, real-time problems with a non-constant number of time-evolving variables (Buede, 1999). C^3I applications presented a clear candidate for taking advantage of this new technology. Actually, systems employing Bayesian networks as inference engines where not new in this domain, an example of that is provided by the ship imaging recognition system showed by Booker and Hota (Booker, 1988). However, problems where each decision may have its expected value varying drastically with time, like the in pilot's sensor and weapon allocation case, still felt a lack of an extensive modeling technique, rule-based systems being the only option for solving that problem.

A DDN can be seen as having three different parts, the decision sub-net, the data fusion sub-net, and the inference sub-nets. These sub-nets are not constant with respect to time, and the modifications between two subsequent time steps can be caused by external changes, like a new track from a sensor in a tracking control application, or by internal decisions, as the decision sub-net may turn a sensor off, depending on the value of its information.

One or more influence diagrams, whose main objective is to define an optimal set of actions to be taken during time step “t”, comprise the decision sub-net. These influence diagrams also calculate the value of information provided to the inference sub-nets, which will be taken in account when defining the optimal set of actions for that time step. In the present application, just one influence diagram will perform those activities, as it is shown in Chapter 4.

One or more separate influence diagrams may be used in the data fusion sub-net, which collects the results from the many inference sub-nets and distribute it to the decision sub-net in a normalized way. In the case study's application, the data fusion sub-net have no specific values to be considered or decisions to be made so a simple set of chance and deterministic nodes are used for fusing the data from the inference sub-net.

Finally, the inference sub-net is comprised by a set of Bayesian networks which main purpose is to analyze the data provided to the system and to make probability inferences on that data. Depending upon the application, these Bayesian networks can be interconnected either directly or only by the data fusion sub-net. In most applications, each Bayesian Network will be linked to a separate object of that application (e.g. in a tracking system, each track will have a separate BN updating its uncertainty), as it happens in this work's case study. Figure 7 brings a sample use of the DDN technique.

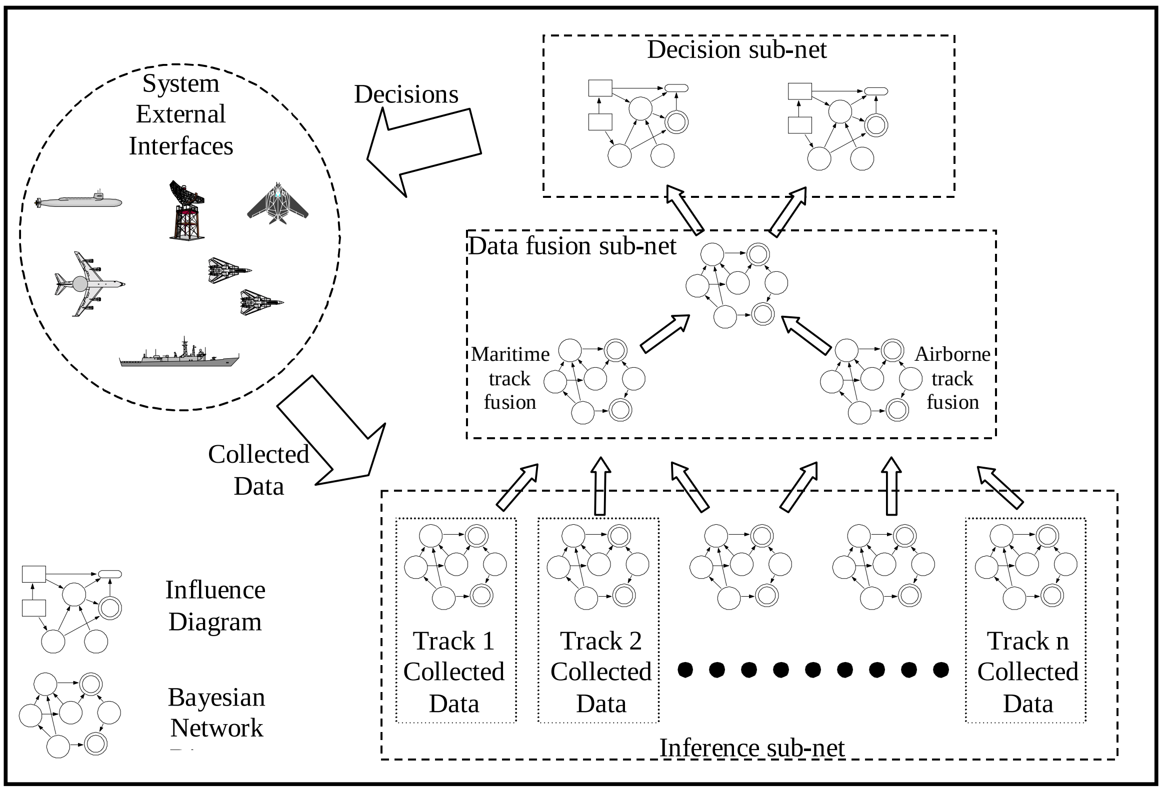


Figure 7 DDN Integrated Defense System

In this sample application, the number of Bayesian Networks in the inference sub-net will be a function of the number of tracks / intruders in the controlled space, and the number of nodes in each data fusion BN will depend upon the number and type of tracks in the inference sub-net. In addition, the number of nodes in each Bayesian network of the inference sub-net at time step "t" is a function of the decisions taken in time "t-1". As an example, if a track is an enemy wild weasel aircraft, depending upon the value of information versus the risk of keeping a radar activated, the system may decide to turn a given radar off, which will decrease the number of input nodes of each inference sub-net's BN.

A DDN system like the one portrayed in Figure 7 may become pretty complex, and its creation must be a system engineering oriented effort. This is the only way of dealing with the plethora of information that will arise from the many domain experts involved in the probability elicitation process. In addition, the wide spectrum of sensors, weapon systems and other external interfaces of the system will also make an optimized approach to the system's life cycle a key issue.

DDNs can be also used in smaller, yet complex applications like the explained in this work; and since it is a relatively new technique, much research is still to be done in order to exploit completely the technique's potential. Nevertheless, it is clear that the advantages already envisioned largely pay the efforts of its implementation.

2.4 Case Study: The Fighter Pilot’s Problem

2.4.1 A Typical Sortie

In a typical mission, here called a sortie, a fighter aircraft has to take-off from a friendly aerodrome, perform a high-altitude flight over friendly territory, descend to a lower altitude preferably before the enemy’s radar coverage, attack the mission's target(s) and egress home safely. Enemy’s role is to detect the incoming fighter and deny its attack, using weapons like interceptors, AAA[3], or missiles.

Behind the scenes lies a high-tech contest between the intruder fighter and enemy’s forces, usually called electronic warfare. This contest can be compared to a hide-and-seek game, where the intruder fighter tries to stay out of enemies' electronic eyes (i.e. early warning radars, interceptor sensors, AAA radars, etc.) as long as he can. The ability of the intruder to hide from these hostile sensors will depend on tactics like low-level flight and reduction (or elimination) of communication and radar emissions. The first is intended to use the terrain as a mask against enemy’s radar, while the latter avoids being discovered by the enemy's passive detectors.

However, flying into enemy territory means to be vulnerable to a wide array of threats, and for most of them awareness is the first requisite to improve the chances of surviving. To be aware, the pilot counts on the information provided by it’s own sensors, which can usually be grouped in two distinct types: passive and active. Sensors in the first group detect all transmissions and classify its respective source; thus they do not need to make any transmissions by themselves. Sensors in the second group are those which transmit for a period of time and wait for a reply in order to obtain information.

---

[3] AAA is the acronym for Anti-Aircraft Artillery, which includes all gun-based weapons employed against airborne targets (aircraft, helicopters, cruise missiles, etc).

Although passive sensors are a stealthy way of gathering information about the enemy, an obvious drawback is in the fact that their efficacy depends on whether the enemy is emitting or not. In addition, passive sensors like the RHAW do not provide a reliable measure of distance.

Active sensors, on the other hand, usually provide more accurate measurement. As a consequence, the decision to decrease uncertainty or detectability is a hard conflict to be solved, mainly during a high attention-demanding situation as a flight sortie. However, there are other issues on the use of active sensors, among them is the management of the sensor's power, that is how to direct it (allocate it) for the many surrounding enemy’s targets/aggressors. The pilot shall perform this allocation wisely, in order to achieve an optimal use of the aircraft's weapon systems (offensively and/or defensively). Here, the level of uncertainty will also gear the pilot’s decisions.

Those decisions are not restrained to electronic warfare considerations. The pilot also has to deal with navigation issues, complex aircraft systems' monitoring, damage control, fuel consumption, and ultimately he still has to pilot his aircraft in a 540 knots near-the-ground flight. In addition, modern aircraft and sophisticated defense systems have dramatically increased pilot’s workload, particularly in the most critical phase of the mission, the attack.

These aspects more than justify the efforts that have being made to reduce the number of decisions a pilot has to take during flight. Systems are being engineered to function automatically most of the time, while warning the pilot when there is a problem. Yet, sensor allocation is a task that is still much more manual than automatic, thanks to the great amount of uncertainty faced by the pilot.

One of the most common sorties is the HI-LO-LO-HI profile. In USAF terminology (and also in many other Air Forces), this means that a fighter will go at high altitude (usually over 15,000 ft), change to a low altitude flight (normally between 200 – 500 ft.) to avoid enemy's radar, attack a ground objective, egress at low altitude until flying over a safe area and return home at high altitude.

This thesis research is based in two particular scenarios taken from a typical real-life sortie; those scenarios were designed to track different situations that might occur in a usual sortie, each with its particular level of complexity. In order to achieve a better understanding on the scenarios, a brief description of its components is needed. Initially, we make the customary assumption of calling friend forces as Blue forces and enemy forces as Red forces.

For the sake of simplicity, we will assume that only three types of Red Air Force's aircraft can provide danger to the Blue fighter: X fighters, Y fighters, and Red Bombers. They can all be found in both scenarios, as well as Blue Bombers, other Blue fighters and non-dangerous aircraft (here named as unarmed aircraft). Table 1 conveys the information on these aircraft that is relevant to the scenarios.

Table 1 Aircraft in the Scenarios.

AIRCRAFT TYPE	BVR Missile Range	IR Missile Range	Radar	Blue Country	Enemy Country	Neutral Country
X Fighter	27 nm	15 nm	EM001	- x -	81	27
Y Fighter	27 nm	15 nm	EM002	- x -	45	15
Bomber	- x -	15 nm	EM002	- x -	72	48
Blue Fighter	40 nm	10 nm	EM003	112	- x -	- x -
Blue Bomber	- x -	10 nm	EM003	80	- x -	- x -
Unarmed Aircraft	-x-	-x-	Other	840	625	535

Besides the airborne threats, ground risks have also to be considered. Again, the number of different types of threats will be restricted, since assigning a greater number would not give much insight and might only decrease the clarity of the conclusions. Two classes of ground threats were considered: surface-to-air missiles (SAM) and anti-aircraft artillery (AAA).

There are five possible types of SAM, four of them have their guidance made by a ground radar while the other uses the target aircraft's IR[4] emissions for homing purposes. The latter is a small portable missile that is particularly dangerous, since it can be launched from an infantry troop's shoulder and does not provide any electromagnetic emission as the previous four SAM.

[4] The main source of IR (infrared) emissions from an aircraft is the exhaust of gases from the engines. However, up-to-date missile seekers can detect IR emissions from the heat caused by attrition between aircraft edges and the air.

Table 2 brings all relevant information about the ground threats that is found in the scenarios.

Table 2 Weapon Systems in the Scenarios.

Ground Threats	Horizontal Range	Vertical Range	Radar	CCME[5] capability
Portable SAM	4 nm	Up to 8,000 ft	No emitter needed	Weak
SAM1	13 nm	Up to 36,000 ft	EM004/EM005	Fair
SAM2	15 nm	Up to 36,000 ft	EM004/EM006	Strong
SAM3	8 nm	Up to 33,000 ft	EM007	Fair
SAM4	5 nm	Up to 33,000 ft	EM008	Weak
AAA1	2 nm	Up to 5,000 ft	EM009	Fair
AAA2	2 nm each, spread over a 4 nm area	Up to 5,000 ft	EM009	Fair
AAA3	2 nm	Up to 10,000 ft	EM010	Weak

As it is in any present-day conflict, electromagnetic emissions play an important role in both case studies' scenarios. Although one can find profuse literature on this subject, sometimes different or even inconsistent meanings might be found for the same words. To prevent confusion, this work will treat sensor terminology as follows:

- Acquisition – mode or phase in which the sensor is looking for probable targets, it is possible to acquire more than one target at a time.

[5] CCME is the acronym for counter-counter electronic measures, which includes all activities taken in the magnetic spectrum in order to overcome a target electronic defense. As an example, if an aircraft employs jamming signals to avoid detection by a radar, the radar's actions taken to overcome the jamming signal are considered CCME.

- Tracking – mode or phase in which the sensor is keeping track of the position of acquired targets, it is also possible to track more than one target at a time.

- Engaged / Lock-on – part of tracking mode/phase, in which the sensor is locked-on a specific target, engaging the beyond-visual-range (BVR) missile to that target and providing guidance information for it. Only one target can be engaged at a time.

Table 3 brings a complete list of all emitters that can be found in both scenarios.

Table 3 Emitters Present in the Scenarios.

Emitter	**Class**	**Type**[6]	**Range**
EM001	Airborne	Acquisition/Tracking	More than 80 nm
EM002	Airborne	Acquisition/Tracking	40 nm
EM003	Airborne	Acquisition/Tracking	More than 80 nm
EM004	Land based	Acquisition	30 to 40 nm
EM005	Land based	Tracking	15 nm
EM006	Land based	Tracking	24 nm
EM007	Land based	Acquisition/Tracking	19/14 nm
EM008	Land based	Acquisition/Tracking	10/9 nm
EM009	Land based	Acquisition/Tracking	10/9 nm

[6] Tracking "type" emitters have the lock-on capability; in other words, they are responsible for tracking and lock-on phases.

Data on Tables 1, 2, and 3 were adapted from current real-life system's characteristics, collected from well establish military publications (e.g. Blake, 1998, Cullen, 1998, Foss, 1998, Jackson, 1997). However, as stated earlier in this work, the main purpose here is only to provide simple scenarios for showing how the different approaches will address them. Thus, a greater number of elements or more detailed information would not add much insight to this thesis purpose.

Another relevant issue is related with the Blue fighter characteristics. A multi-role aircraft is assumed, so the equipment shall reflect the capabilities of current state-of-the-art fighters. Since this is a sensitive type of information, the data being used has been adjusted to provide a case study's fighter with at least similar capabilities as we would find in real-life.

One of the most important sensors in a fighter is the radar. We assume that Blue fighter is equipped with a low probability of intercept (LPI) radar, which uses a spread bandwidth in order to confuse enemy's radar warning receivers. Although a modern radar is fairly complex and might have many different modes, for decision making we would be interested only in the capacity of a mode to engage a specific track, so we will consider that the Blue fighter's radar has two (classes of) modes: Low and High. The first emits less energy, which makes it less detectable, while the other emits more energy but is capable of engaging a target.

Along with the radar, the aircraft is equipped with two passive sensors: RHAW (Radar Homing And Warning), and MAW (Missile Approach and Warning). The first senses any radar activity, tracks it's azimuthal movement, and classifies its source. The latter relies on IR spectrum disturbances for plotting the launching of a missile

For self-defense, the aircraft is equipped with 6 active-radar missiles, 2 infrared missiles, flares, chaff, and a radar jammer. The jammer actively interferes with the enemy’s radar, while chaffs do the same passively, but with limited success; flares are used to deceive IR controlled devices, like the missiles launched by the enemy.

Usually, an aircraft's self defense system is an integrated suit of highly complex devices like the ones described in the previous paragraph. In addition, much of the actions to be taken by those devices must be executed in a synchronized fashion. As an example, in order to escape from an enemy IR missile one must execute a break maneuver (a very tight curve) and launch a programmed sequence of flares. Each of those actions would achieve no result if executed alone.

It is interesting to observe that even when executed together in a synchronized way, these actions will have distinct results for each different type of missiles. In short, the effectiveness of performing a break and launching a programmed sequence of flares is a function of the missile’s characteristics, like aerodynamic performance, seeker complexity, anti-flare capabilities, etc. Data concerning these issues is classified, and a more profound analysis on it is beyond the objective of this work. Yet, the break-flare maneuver will be considered here as an effective “11th hour” measure (e.g. an IR missile is launched against the Blue fighter for sure), while in less dramatic situations this action will be regarded as pure wasting of resources.

Considering the high level of integration among the self-defense systems and in order to limit the complexities of the decision making process, it will be assumed that the pilot has four different defensive actions with respect to the integrated self defense suit (chaff, flares, and active jammer): Break and ECM, deviate and ECM, ECM only, and the trivial action (do nothing). The first means that a break (regarded as a “last resource” action, since it deteriorates the aircraft energy) is to be executed in conjunction with an ECM device (flare, chaff, or active jammer), to be defined and activated by the automated suit depending upon the kind of threat. In the second case, no “last resource” maneuver is needed, and an intermediate action (deviate from the threat) is taken in conjunction to the automated ECM action. The third action is to perform a deviation in the aircraft's course in order to avoid a given threat, this action may be taken either by the pilot or by the autopilot system (if engaged).

Active-radar missiles shall be used in mid-range BVR (beyond visual range) air combat, with an ideal range of 45 miles. Infrared missiles shall be used in visual range combat (also called dogfight), with an ideal range of 10 miles. It is interesting to note that a BVR missile would be used only if the radar is set to a “High” mode.

We also assume that enemy fighters will be using a combination of missiles: Infrared guided missiles with a 15 miles range, and semi-active guided missiles for BVR combat, with an ideal range of 27 miles. As a final general assumption, all missiles pK (probability of kill) will be assumed as .90 (90%) when launched within their operational range. That is, once fired, it is considered that they will destroy the target with a probability of 90%.

Before going further in the case, it is necessary to explore the meaning of the missile-related assumptions; in other words, to assess what they imply to the situation. It is clear that in visual range the enemy fighters will be a step ahead, as they are able to shoot at 15 miles, while the Blue fighter is limited to a 10 miles range. So, visual-range combat is to be avoided by the Blue fighter. Nevertheless, it is also clear that the Blue fighter will have an advantage on the BVR combat, since he may launch its missile before the enemy fighters become able to shoot at him. However, to guarantee that advantage he has to comply with some arithmetic inferences for effectively denying the enemy's ability to shoot.

Once his objective is to hit the target before it could fire his missile, Blue fighter's pilot has the range of 27 miles as his first limit. However, it is necessary to find out when he will have to pull the trigger in order to achieve this limit. Figure 8 depicts our problem.

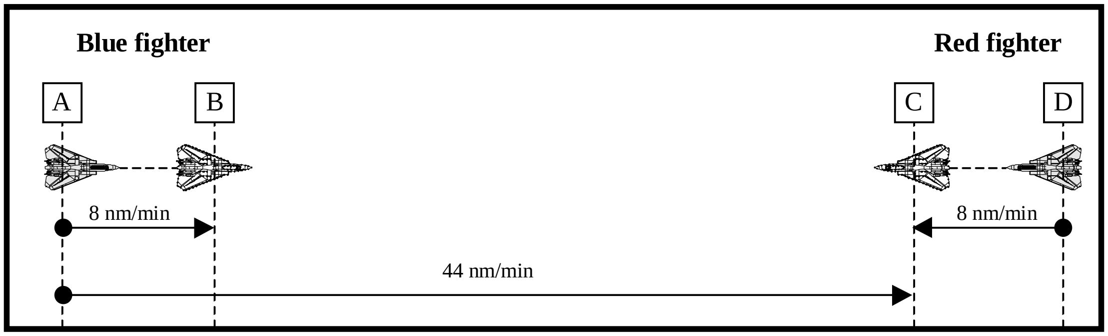


Figure 8   Missile Launch Point Assessment

The arrows show the velocity of both aircraft (8 nm/min, or approximately 480 kt) and the missile (44 nm/min). In point “A” the pilot pulls the trigger to shoot at enemy fighter in point “D”. During the missile’s time of flight both aircraft will move, so when the missile hits the target, blue fighter will be at point “B” while the target will be at point “C” (the impact point). From the assumptions we know that distance BC must be at least 27 miles in order to prevent the enemy aircraft’s ability to shoot. What we want to know is distance AD, that is the limit blue fighter’s pilot have to shoot his own missile.

Distance AD can be calculated by a simple algebraic calculation:

$$\frac{(\overline{\boldsymbol{AB}}+\overline{\boldsymbol{BC}})}{44}=\frac{\overline{\boldsymbol{AB}}}{8}\Rightarrow 8\times(\overline{\boldsymbol{AB}}+27)=44\times\overline{\boldsymbol{AB}}\Rightarrow \overline{\boldsymbol{AB}}=\frac{8\times 27}{44-8}=6$$

After the launching, both aircraft will move approximately 6 nm during the missile's time of flight. Thus, AB = CD = 6 nm and AD = AB + BC + CD = 6 nm + 27 nm + 6 nm = 39 nm. That is, pilot’s limit for the “pull the trigger” decision is 39 nm. If we do the same calculations for the enemy's IR missile (15 nm range), we will find a limit of approximately 17 nm.

It should be clear that some simplistic assumptions are being made here, like the same velocity for both missiles (BVR and IR). In a real case, more complicated calculations are made for establishing the actual limit. Nevertheless, for this model's purpose the preciseness of that limit is much less important than the very awareness of its existence and the considerations that will be made in the model as a consequence of it. Thus, I am considering the above mentioned assumptions as being fair in the light of our present goals.

However, for the sake of simplicity, our coarse calculations did not include some more sophisticated (but not negligible) issues like the time between pilot’s decision and the missile’s departure, that is the time for the human body’s action of pulling the trigger and for hardware’s action of igniting missile’s propellant. In order to compensate those issues, this work will consider 40 nm as a limit against BVR missiles and 20 nm as a limit against IR missiles.

It is interesting to note that those limits would be optimized with the use of an automated self-defense system, which is able to perform instant calculations with real-time data on relative velocity between aircraft and target. Nonetheless, a pilot by himself is not able to perform those calculations in real time, so the worst case was considered as a matter of prevention.

2.4.2 First Scenario

The first scenario covers the initial part of a typical sortie, where the intruder aircraft (here called blue fighter) is flying at high altitude towards the target. In order to focus in the air-to-air scenario over a neutral territory, it is assumed that the flight is taking place over international waters. In addition, the following assumptions were made:

- Blue aircraft's radar is on and set to a “Low” mode.

- The flight is taking place over international air space (i.e. over the ocean).

- The pilot’s main objective at this phase of flight is not to shoot at enemy aircraft, but to reach the descending point and proceed with the mission, preferably with full self-defense capacity (no weapons deployed) and without being contacted by the enemy.

- There is no escort formation. That is, all defense procedures are to be made by the aircraft itself.

- Although an attack formation has a minimum of two aircraft, a mandatory requisite for operational missions in most air forces, we will consider only the decisions that have to be made by one of them. That is, synergy in the formation can be considered as an additional help for the pilot, but we assume the worst case (no synergy) for this work's purpose.

- In addition to the above assumption, no external help to the pilot will be considered. Again, the worst case is understood and a possible extra help is not modeled in order to avoid adding unnecessary complexity to the model. An example of external help would be the information about tracks received by a data link between the Blue aircraft and an Airborne Warning And Control System (AWACS) aircraft.

Figures 9, 10, and 11 are spaced by a 20-second interval (corresponding to 3 miles at high subsonic high-altitude flight) and define the first scenario.

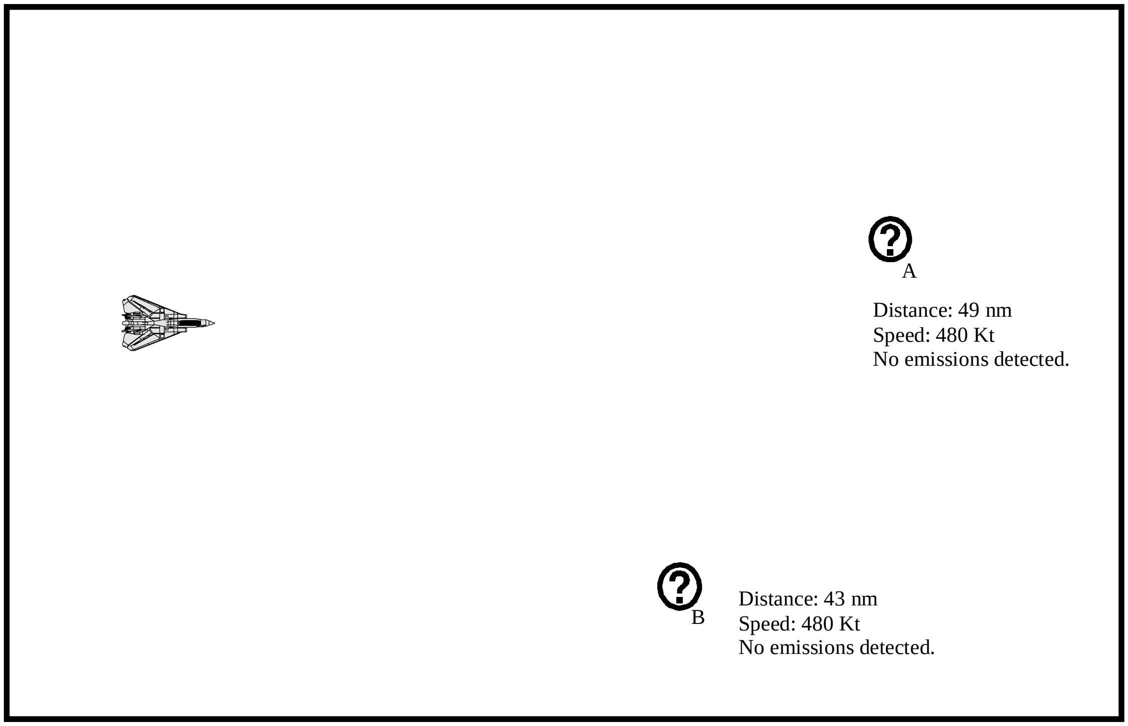


Figure 9 First Scenario Initial Position

In Figure 9, Blue aircraft is still out of enemy's radar range and remaining undiscovered is a major issue at this part of mission. The aircraft's LPI radar has detected two possible threats, displayed here as question marks. However, no further information is available other than distance and bearing, since both tracks are not emitting (at least no emissions were detected and identified by Blue fighter's RHAW at this time).

A great deal of uncertainty is involved in this situation. Both tracks could be enemy interceptors or not. Even if they are identified as fighters, there will still be no clues whether they know or not about blue aircraft's existence and position. It would be fair to say that at this point the pilot does not have sufficient information to shoot at either aircraft.

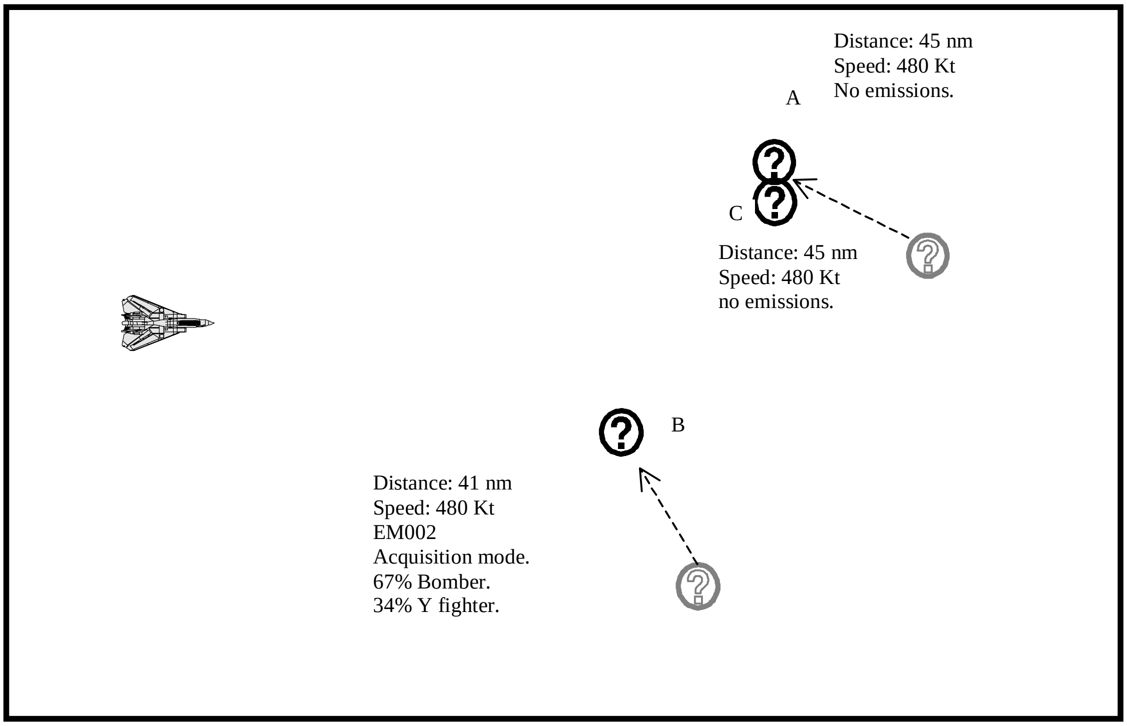


Figure 10 First Scenario, Second Snapshot

Figure 10 is another snapshot of the scenario, but taken 20 seconds after the one in Figure 9. Now track "A" became closer with a slight change on bearing, while track "B" practically stayed at the same distance but with a different bearing. Furthermore, track "B" is now emitting in tracking mode and the RHAW has identified those emissions as being from EM002, a radar that is used in two different aircraft types: a bomber and a "Y" fighter. The former does not carry BVR missiles, but only two wing tip mounted IR missiles; thus it would threaten the Blue fighter only at distances within visual range. The latter is more dangerous, since it has four BVR missiles under its fuselage and two wing tip mounted IR missiles.

Previous intelligence reports have shown that enemy's air force has 45 type Y air dominance fighters and 72 bombers, while the neighbor neutral country uses 15 type Y air dominance fighters and 48 bombers. Assessing the probabilities, we will discover that:

- P(Y|EM002,&)[7]=.33, that is probability of aircraft type Y given EM002 and previous information is (45+15)/(45+15+72+48) = 33%.

- P(B|EM002,&)=.67, that is probability of bomber aircraft given EM002 and previous information is 1 -.33 = 67%.

- P(E|EM002,&)=.65, that is probability of enemy aircraft given EM002 and previous information is (45+72)/(45+72+15+48) = 65%.

- P(N|EM002,&)=.35, that is probability of neutral aircraft given EM002 and previous information is 1 - .65 = 35%.

- P(E|Y,EM002,&) =.75, that is probability of being enemy given Y aircraft, EM002, and previous information is 45/(45+15) = 75%.

---

7 The symbol "&" is used to denote prior knowledge (Howard, 1989), meaning that previous information about that situation was considered in the assessment of the probabilities.

- P(Y,E|EM002,&) = .25, that is probability of being enemy and Y aircraft given EM002 and previous information is P(Y|&) * P(E|Y)= 25%. We can check this by 45/(45+15+72+48) = 25%.

It is important to note that these are “static” probabilities, that is track B’s emission alone can be considered as .25 chance of being from an enemy’s Y aircraft. However, in a real situation much more than just this mere quantitative assessment will be made. Issues related to the enemy doctrine (the way aircraft are usually employed), like types of formation, altitude, number of aircraft, period of the day, and other characteristics usually employed by enemy’s fighters will be also considered by blue intelligence personnel to evaluate the static probabilities.

These considerations are not used here for they would add a great deal of complexity that is not related to this work’s purposes; instead, only the quantitative assessment will form the static (prior) probabilities in our scenarios. Nevertheless, issues like aircraft’s behavior (e.g., approaching fast and at acquisition mode) certainly will increase (or decrease) the given static probabilities of a dynamic assessment.

In addition to all those changes, a new track, “C”, has been perceived at the same point of track “A”, which usually indicates an aircraft formation. Now, the pilot has more information but a high level of uncertainty is also involved. Yet, no emission has been detected from track “C”, which means that he either is not emitting (thus can not launch a

BVR missile) or the RHAW has missed track's “C” emission (e.g., a very week signal). In both cases, it is very unlike that track “C” represents an imminent threat.

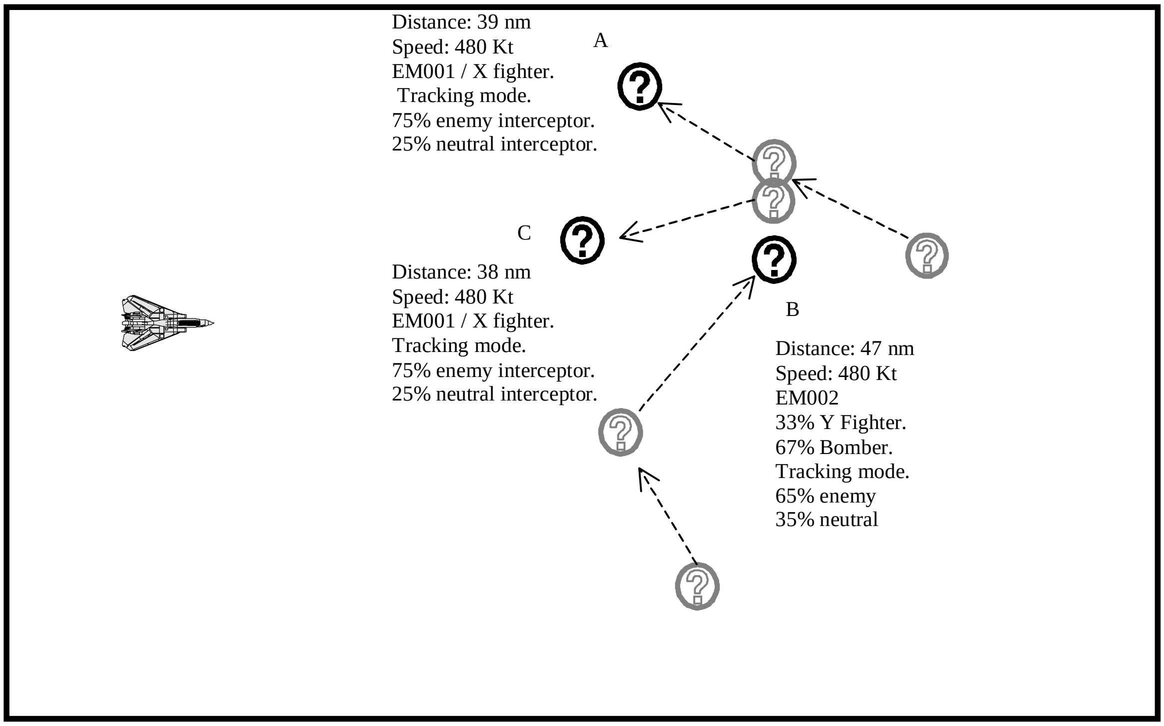


Figure 11 First Scenario, Third Snapshot

The picture in Figure 11 is also a snapshot taken 20 seconds after the one in Figure 10, and it brings new data to our problem. Tracks “A” and “C” are now emitting and both have been identified it as Type X air dominance fighter, which is the top-of-line air dominance fighter of Red Air Force and also of neutral country's Air Force. It is a supersonic fighter that carries six BVR missiles and two IR missiles.

Previous intelligence reports assured that enemy’s air force has a total of 81 type X aircraft operating at this theater, while neutral country’s air force has 27. A pretty straightforward assessment of the probabilities will show us that P(E|X,EM001,&) = .75, that is probability of being enemy given aircraft type X, EM001, and previous knowledge is 75%.

At the same time, track “B” has already crossed our aircraft’s path and its distance is increasing. Taken alone, this would diminish track “B” threat assigned level, but we also have to consider the possibility that it is performing an engagement maneuver, which obviously would not diminish the assigned threat level. In addition, it is still in tracking mode, a clue that goes against a lower threat level assignment.

Track “C” is emitting in Tracking mode, but at a greater distance than track “A”. Although it’s path is not directly threatening our aircraft, the fact that it may be working with track “A” indicates that he could be trying to disengage from Blue fighter’s radar and going for a better position, while his wingman (track “A”) acts like a bait. Hypothesis of teamwork attack procedures like that would certainly increase considerably the threat level assigned for track “C”, even in a greater degree than the one assigned for track “A”, since a hidden enemy may be much more dangerous than a closer (but still visible) one.

Given this developing situation, the pilot has to decide whether or not to turn to a radar's high mode, get the lock-on and shoot a missile at aircraft A, B and/or C no later than they become able to attack. However, if he pulls the trigger too early, he would be shooting at a neutral aircraft. In addition, he will be announcing his position, nature, and intentions to all other aircraft, as they also might be in doubt about blue fighter's track on their radar or would even have not detected him yet.

What kind of questions should be answered by a pilot or by an automated system in this situation? The question "what is the most threatening emitter?" is a good candidate among many uncertainties to be solved. However, the most urgent questions to be answered are those related to which specific actions must be taken and their respective timings (i.e. shall the semi-active missile be fired? If so, when?).

It is interesting to note that at any time neither pilot's nor automated system's judgment are free from uncertainties. Furthermore, since those uncertainties would change dramatically in a short period of time, the same applies to the answers they have to provide.

2.4.3 Second Scenario

This time, the aircraft is at low altitude flight and inside the 200 miles range to the main objective. This is a riskier scenario, as the flight is over enemy's territory and usually a greater number of threats are found near the objective. First, we have to make the following assumptions:

- Since the flight is taking place over enemy territory and at low altitude, any perceived weapon system (aircraft, AAA, missile, etc) will be considered an enemy device, thus a threat to blue fighter.

- For the sake of stealthiness, radar is initially off. This means all data about the enemy threats will be their emissions' bearing and strength, provided by the RWAS, and an eventual missile launching, provided by the MAW system.

- The pilot's main objective at this phase is not to shoot at enemy defenses, but to reach the target, preferably not being contacted by the enemy.

- Once again, there is no escort formation. That is, all defense procedures are to be made by the aircraft itself.

- Like in the first case, only the decisions that have to be made by one aircraft will be considered. That is, synergy into the formation or between the Blue fighter and external sources through a data link can be viewed as an additional help for the pilot, but will not be modeled here.

The two snapshots of the second scenario, portrayed by Figures 12 and 13, were taken with a 30-seconds interval (corresponding to 4 miles at high subsonic low-level flight).

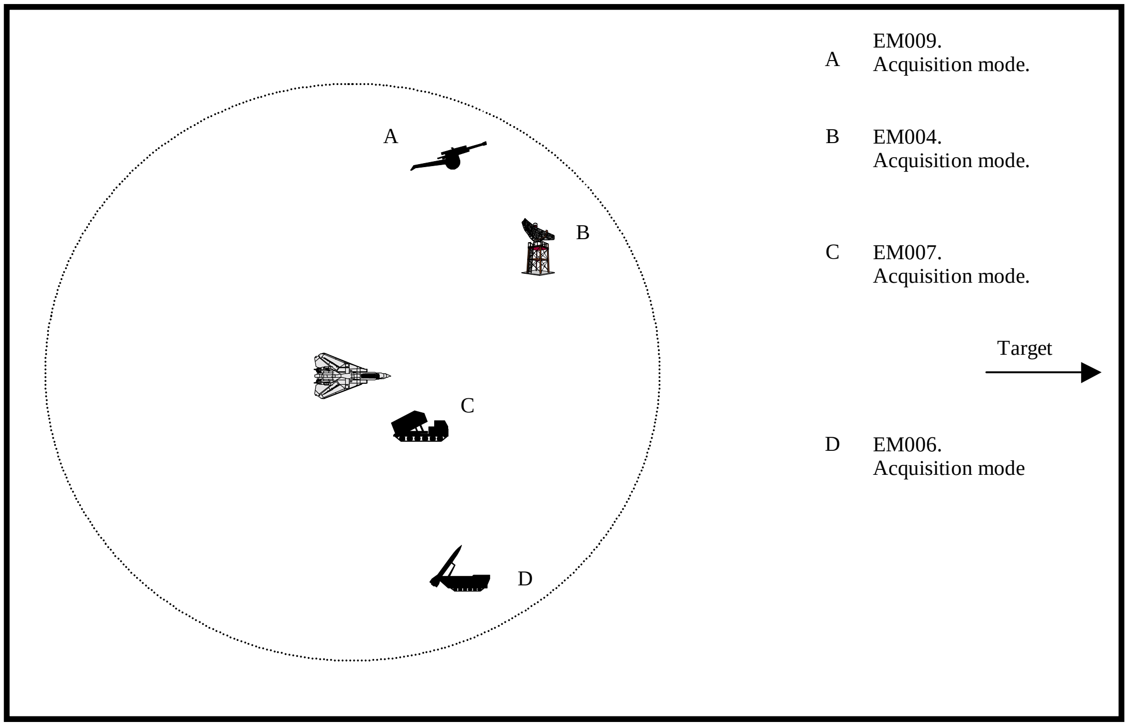


Figure 12 Second Scenario Initial Position

In Figure 12, since we are in passive mode, no distance information is given. Consequently, the pilot can deal only with the information provided by enemy's emissions, like relative bearing, type of emitter, threat(s) associated with each emitter, and the emitter signal's strength. In the polar display of that figure, the distance between the tracks and the center is related to signal's strength, which is not a reliable indication of the emitter's distance. The farther from the center, the stronger a signal is.

Four emitting threats can be seen on that particular snapshot. Track "A" emission comes from a 15NM-range radar utilized in two of enemy's defense systems: a mobile gun system, and a static multiple-guns system. The mobile gun system is a truck-mounted weapon system composed of four 23mm guns, with an effective range of 2 nm, while the fixed system is a set of twin 23mm guns (2NM range each) spread over a 2 nm area. The first is a system used mainly for defending a specific point, while the former is more suitable for defending a small area. Given the circumstances, both are not likely to do any harm. In addition, that radar is still in acquisition mode and the aircraft should be out of the system's limited effective range. However, a strong signal means a nearby source, which makes it a threat to be considered.

Threat "B" emission comes from a more powerful acquisition radar with a 30 to 40 nm range (EM004) that is used in conjunction with one of two enemy tank-mounted semi-active missile systems: SAM1 and SAM2. The first system carries three 13 nm range, two stage, solid fuel, low to medium altitude semi-active guided missiles; the latter carries four 15 nm range, solid-fuel, low to medium altitude semi-active guided missiles.

Both systems use EM004 for acquisition. Once a target is acquired and enters into the attached system weapons' range, EM004 passes all data to the attached system's fire control radar, which assumes the final tracking procedures. SAM1 system uses EM005, a conical scanning pulsed radar that has an optical backup system for controlling the fire in case of jamming at the radar or presence of anti-radiation missiles (which may demand the fire control radar to be shut-off). SAM2 system uses EM006, a monopulse guidance radar that provides fire control capacity against high-performance aircraft and cruise missiles at low to medium altitude.

Intelligence sources about those semi-active missile systems reports that the enemy has 30 SAM1 and 20 SAM2 systems. Awareness of those quantities allows us to perform the following probability assessment.

- P(SAM1|EM004&)=.60, that is probability of SAM1 semi-active missile system given EM004 and previous information is (30)/(30+20) = 60%.

- P(SAM2|EM004)=.40, that is probability of SAM2 semi-active missile system given EM004 and previous information is (20)/(30+20) = 40%.

Threat “C” is an emission from an EM007, a radar that comes with the SAM3 system. The system is comprised of a six-wheel truck that carries the EM007 radar and four 8 nm range, single stage, solid-fuel, low altitude semi-active missiles. The radar provides both target acquisition and tracking; its conical scanning based fire control guidance can direct two missiles at the same target, each one with a different frequency to frustrate ECM.

Although it is in acquisition mode and still a weak signal, two factors contribute for raising its threatening status: the system’s high degree of accuracy and its relative proximity. The last is inferred by the fact that a weak signal of a 19 nm-range radar may indicate that its source can be at about 19 nm away (less than 3 minutes away). However, as it was said before, signal strength is not a reliable source for range estimation.

Threat “D” is an emission of EM006, used in the SAM2 system. Along with the inherent strong ECM resistance of its monopulse radar, accuracy is also a solid characteristic of this system. Although still in acquisition mode, the strong signal increases the likelihood of a higher threat level assignment.

Figure 13, although taken only 30 seconds after Figure 12, shows a quite different situation. Here, practically every threat changed its status and the MAW has detected a missile launching (portrayed as threat “E”).

The MAW relies on disturbances of the IR spectrum that happen during the launching of a missile, however false alarms are a concern in this kind of equipment. In this work, the static false alarm rate is assumed to be .3 (30%). That is, before any other consideration, it is known that 3 out of every 10 alarms are false. However, in case of an alarm, the pilot (and any automatic decision system) has to consider current aircraft's location, altitude, terrain characteristics, and other relevant factors when assessing the dynamic probability of false alarm.

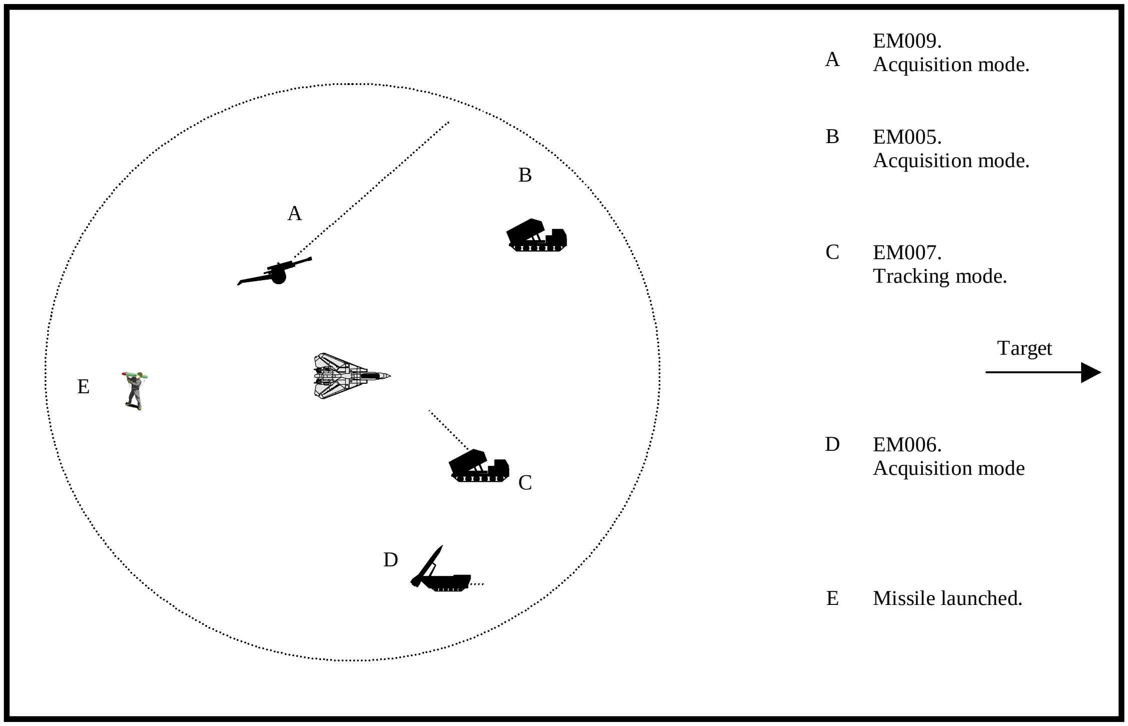


Figure 13 Second Scenario, Second Snapshot

Threat “A” has moved fast in polar terms and its signal decreased in strength. Intuitively, its threatening status should have decreased, but it is still a considerable threat.

Threat “B” is now sending emissions of an EM005, which probably means that a SAM1 was associated with those previous EM004 emissions. Though showing some decreasing, the signal still has considerable strength; furthermore, it is in tracking mode, which means a rather significant threat.

Threat “C” has increased its signal strength, which means that it is getting closer. In addition, it is now in tracking mode, which means that its radar is trying to lock on the aircraft. Once locked on, a missile would be launched in about 4 seconds.

Threat “D” apparently had slightly decreased its signal and showed a slow polar moving. These are clues for an approaching but still out-of-range threat. A threat level assessment to that track would not have shown a considerable increase, since it also had not changed to acquisition mode.

On top of the previous threats, the detection of a missile launch (treat “E”) has considerably increased the overall threatening status of the situation portrayed by Figure 13, while still maintaining its great level of uncertainty. Among the questions to be answered and actions to be taken we can list those below:

- What is the most dangerous threat at this moment? The missile is obviously a good candidate, but it has also a big deal of uncertainty, as its probability is something like .7 of being a dangerous threat and .3 of being no threat at all. On the other way, track “B” is certain to be an immediate threat.

- Should a Flair and Chaff load be released in addition to a defensive maneuver in order to deny the missile threat? If so, those countermeasures will alert other enemy’s radar about the intruder aircraft.

- Should the jammer be activated to deny threat “B”? Apparently yes, but if the decision of no chaff and flare is made, the jammer will be the one to alert the other enemy’s radar and an evasive maneuver would be more suitable. Acquisition does not necessarily imply in a future tracking procedure, although the situation is more likely to imply it.

Without the support of an automated real-time decision system, these questions and some other pertinent ones are not likely to receive a detailed assessment by the pilot, at least not into a suitable time. What happens in actuality is that pilot's conditioned training and experience will dictate the actions to be taken. However, an automated system would bring together the responsiveness of current fast computers with the experience of most experienced pilots to every combat aircraft of the fleet. The next three Chapters are intended to show how each approach might face this challenge.

CHAPTER 3

THE KNOWLEDGE SYSTEMS APPROACH

3.1 Some Initial Thoughts

Winston (Winston, 1986) argued that in contrast with other fields of human research, Artificial Intelligence is far from being easily definable. In practice, there might be as many (correct) definitions as there are people in the field, and an eventual search for a definitive definition is very likely to keep one completely diverted from the importance of the field itself.

Although it has been receiving a great amount of attention from the media in the two past decades, the field of Artificial Intelligence is not a new-fashioned issue as one might think. According to Harmon (Harmon, 1985), the first attempts to create an electronic machine that could be able to mimic the human behavior when solving problems date from the end of Second World War.

At that time, a few scientists began to idealize that the newly created electronic computer might some day be programmed to achieve a human-like logical processing, whereas most of the scientific and academic communities were devoted to explore the strong numerical capabilities of the new invention.

Nevertheless, many authors (e.g., Lucas, 1990, Walker, 1989) regard the formal commencement of the Artificial Intelligence as being the Summer Seminar held in 1956 at Dartmouth College. It is interesting to comment that during this summit, predictions were made asserting that in 25 years everyone would be involved in recreational activities, while computers would be doing all the work (Walker, 1989).

While most of us were not involved in recreational activities, nearly all activities in human life are done in part with the help of computers. Within that context, knowledge systems are one of the most popular tools helping to make computers even more useful, by enabling it to work out problems that only humans are currently capable to solve.

Knowledge systems attempt to take advantage of the cumulated knowledge of many experts in a given field in order to make inferences and solve problems in that domain. In short, the approach taken is to synthesize the cumulated knowledge in a set of rules of thumb, called heuristics (Lucas, 1990), while using a computer-based inference engine for applying the correct subset of heuristics for each domain specific question.

One of the pioneer efforts in making that theoretical approach a real life application came from Stanford University's MYCIN project (Buchanan, 1985), a medical domain knowledge system that set a landmark in the artificial intelligence field. I decided to use MYCIN as a base for the “Wise Pilot 1” system not only because of its significance for the IA field, but also for taking advantage of its simple and elegant approach in storing and using human knowledge for practically any domain.

3.2 Reasoning Under Knowledge Based Systems

This section is heavily based on the work of Buchanan and Shortliffe (1985), and is intended to convey a brief refreshment on the theory of MYCIN-like knowledge based systems, in which category “Wise Pilot 1” is included. A complete coverage on this subject is not in the scope of this work. However, the interested reader will find vast literature on knowledge based systems (e.g. Hayes-Roth, 1983, Buchanan, 1985, Walker, 1989, and Lucas, 1990), and on its many applications as well (e.g. Quinlan, 1987).

Heuristics about a given domain are stored in the system's database in a proprietary format that is most convenient to the developer. In “Wise Pilot 1”, reflecting the approach employed by MYCIN, knowledge is stored in quadruples with are formed by triplets with the format “object-attribute-value” and by a “certain factor” assigned to that triplet.

Objects, also called in MYCIN as “contexts”, are entities that have separate function in the system, each of these entities have a set of attributes. An attribute is a specific characteristic of a given object, and it can assume different values. A certain factor is a number between –1 (minus one) and 1, representing the uncertainty about the fact conveyed in that quadruple, where 1 means absolute confidence on that fact. Section 3.3 discusses about certain factors with more detail.

While quadruples store the system's current knowledge about a given domain, the inferential reasoning that updates this knowledge is encoded in what is called "production rules" (Buchanan, 1985), each of those being formed by a premise and an action. A premise is a set of triplets regarding triplets in the knowledge database (object-attribute-value), and an action is a statement about what should be done in the case of a true premise. As an example from "Wise Pilot 1", Rule 002 is shown below:

Rule 002

IF:	TERR is "enemy".
THEN:	THREAT is "yes" for all TRACK.

TERR means terrain, which is an attribute of the ITERATION context. In other words, each iteration of the program is treated as an object, with terrain nature as a member of its attribute list that may have values "enemy", "friend", or "neutral". In this case, if the premise "TERR is enemy" is true, the action "THREAT is yes for all track" is taken. This means that if the value of the TERR attribute is currently set to enemy then the attribute THREAT of all TRACK objects is set to value "yes".

The control structure for the reasoning process is a goal-oriented backward chaining of rules (Buchanan, 1985), which means that the system retrieves the list of rules in a way to establish the necessary conclusions for its goal. In the above example, Rule 002 could be activated when the system wants to know whether a track is a threat or not. For controlling this backward chaining logic process, I used the same mechanism employed in MYCIN, which is shown in Figure 14.

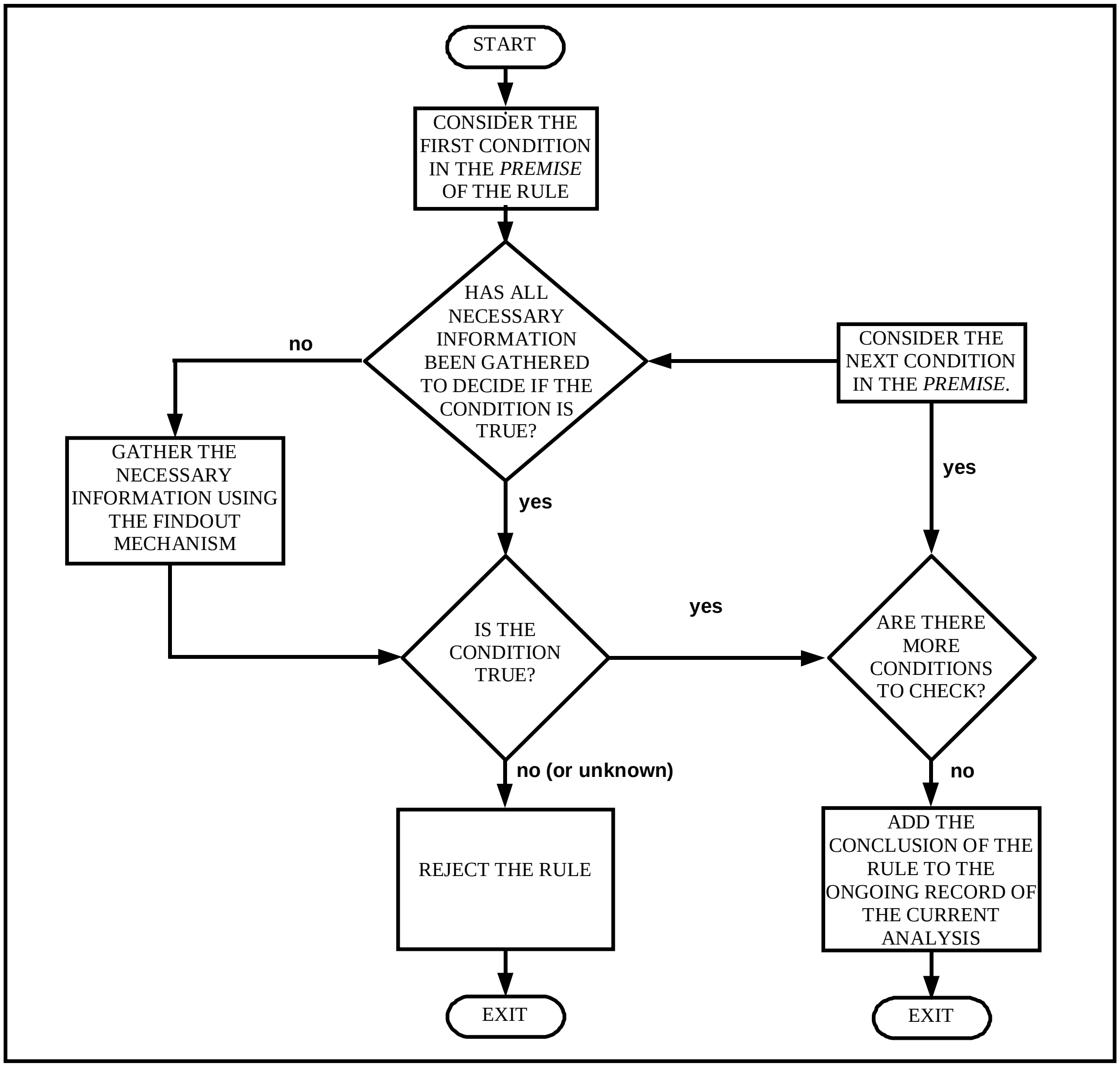


Figure 14 MONITOR’s Logic Mechanism

This mechanism is a flow chart extracted from Buchanan (1985), and explains how the system browses through the rules in the list. Initially, the main rule is activated (Rule 001 in the “Wise Pilot 1” system) and its premises are checked, if the first premise is satisfied then the system goes to the second premise and so on until the last premise is verified. If all premises are true the system applies the action for that rule, which updates the knowledge database. When a premise is false the rule is rejected.

Sometimes, the logical mechanism will find that there is no sufficient information to verify a premise. In those cases, it activates the FINDOUT mechanism, which is responsible for determining where to find the required information. Figure 15 is a flow chart adapted from Buchanan (1985) describing the logic behind FINDOUT.

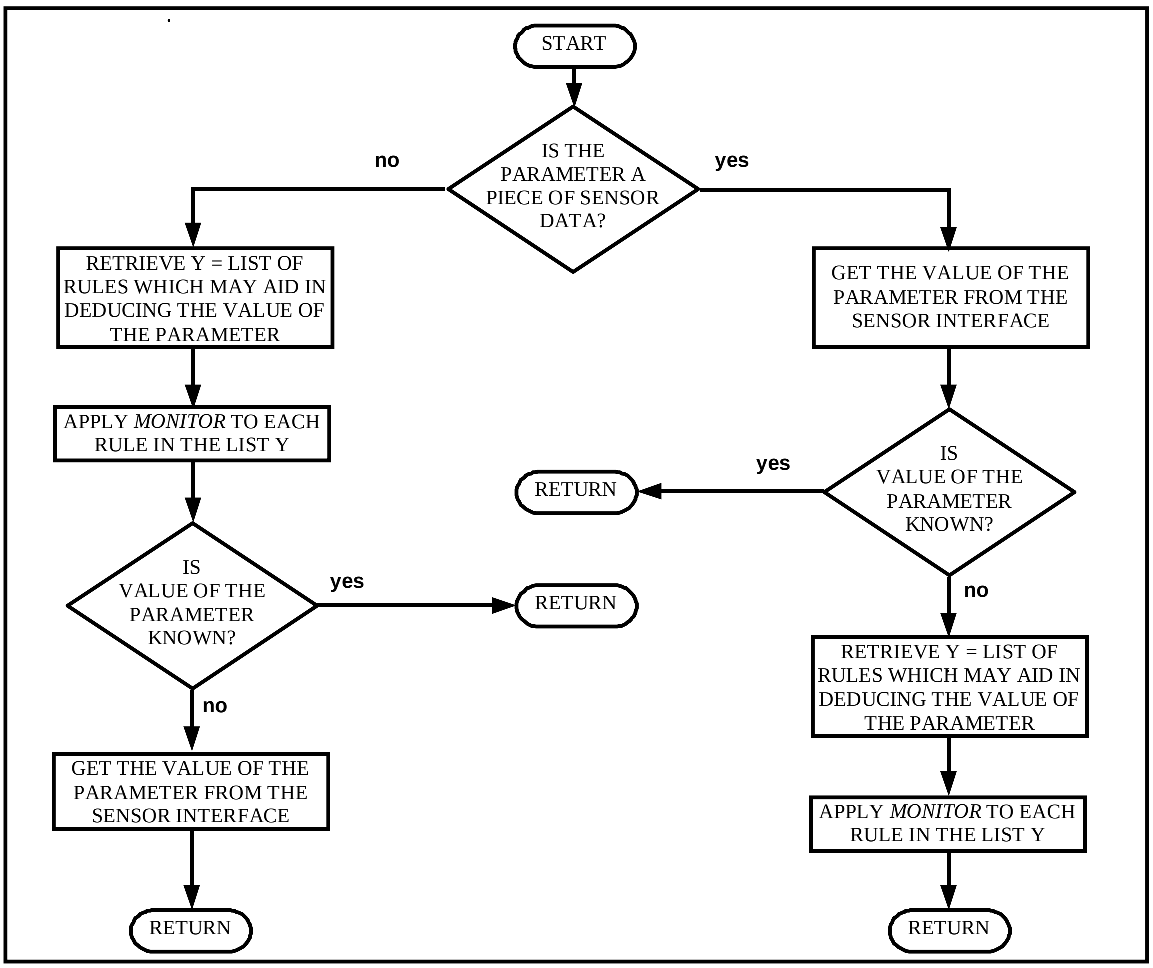


Figure 15 FINDOUT's Logic Mechanism

The strategy depicted in Figure 15 is used to decide whether a premise can be verified by existing data from sensors or another rule should be triggered in order to get that data. The combination of MONITOR and FINDOUT mechanisms is the key for the goal-oriented backward chaining of rules that comprises the inference engine of the expert system as a whole.

In a simple example of this combination, each iteration of the “Wise Pilot 1” system begins with the activation of rule 001. In order to verify rule 001’s first premise, MONITOR activates FINDOUT for gathering the necessary data. FINDOUT, in its turn, realizes that another rule has to be triggered to get the necessary information. However, this recently triggered rule will also have its premises, which may force another rule to be activated and so on. This procedure will finish only after a comprehensive survey through the available data from sensors and knowledge from the database, in an analytical process that ideally mimics the expert’s decision making process.

3.3 A Word About Uncertainties

As stated before, from the beginning of their conception Knowledge systems were intended to capture expert’s knowledge in the form of “chunks of information” called heuristics. However, even the best expert will have a great level of uncertainty in his or her judgment, and the information cataloged in the form of rules also had to convey the amount of uncertainty.

When faced with the problem of expressing this uncertainty, the initial approach adopted by most decision making systems' developers was the subjectivist approach of probability theory (Adams, 1976). However, the major complain about this approach was the lack of a practical way of properly estimating "a priori" probabilities required by Bayes theorem (recall the theorem's explanation in page 14). To avoid these practical difficulties, Buchanan and Shortliffe (1985) developed the "certain factors" concept in an attempt to provide the experts with an intuitive way of stating their uncertainty on a given hypothesis. They regarded this concept as a convenient alternative to the established probability theory.

A certain factor (CF) measures the change in the degree of belief on a given hypothesis as a result of new evidence. In other words, if an expert believes that evidence "*e*" increases the likelihood of hypothesis "*h*" then it will have a positive CF (between 0 and 1). Likewise, if that evidence decreases the expert's belief the hypothesis, the CF will be negative ranging between –1 and 0.

Actually, the certain factor is the difference between two functions, measure of disbelief MD[h,e] and measure of belief MB[h.e]. The first quantifies the decrease of belief in hypothesis "*h*" in face of evidence "*e*", while the latter reflects the case in which evidence "*e*" increases the likelihood of hypothesis "*h*". Thus, when MD[h,e] is greater than 0, MB[h,e] is 0, since an evidence can either increase or decrease the expert's belief in a given hypothesis.

In MYCIN-like systems, the CF for a given hypothesis is combined with the existing belief through the formulas

$$\boldsymbol{CF}[\boldsymbol{h},\boldsymbol{e}] = \frac{\boldsymbol{MB}[\boldsymbol{h},\boldsymbol{e}] - \boldsymbol{MD}[\boldsymbol{h},\boldsymbol{e}]}{1-\min\{\boldsymbol{MB}[\boldsymbol{h},\boldsymbol{e}],\boldsymbol{MD}[\boldsymbol{h},\boldsymbol{e}]\}}$$

and

$$\boldsymbol{CF}_{\boldsymbol{COMBINE}}[\boldsymbol{X},\boldsymbol{Y}] = \begin{cases} \boldsymbol{X}+\boldsymbol{Y}(1-\boldsymbol{X}) & \boldsymbol{X}>0,\boldsymbol{Y}>0 \\ \dfrac{\boldsymbol{X}+\boldsymbol{Y}}{1-\min(|\boldsymbol{X}|,|\boldsymbol{Y}|)} & \boldsymbol{X}>0,\boldsymbol{Y}<0 \quad ,\boldsymbol{or}, \quad \boldsymbol{X}<0,\boldsymbol{Y}>0 \\ -\boldsymbol{CF}_{\boldsymbol{COMBINE}}(-\boldsymbol{X},-\boldsymbol{Y}) & \boldsymbol{X}<0,\boldsymbol{Y}<0 \end{cases}$$

where X and Y are current and the updating certain factors for a given hypothesis (the order is irrelevant). When faced with new evidence, the system will use the above formulas for updating the hypothesis' CF, which will be stored in its respective quadruple in the database.

The system's inference engine uses a set of operators in order to obtain the necessary information for verifying a rule's premise. Each operator follows a "CF threshold" in order to either return true or false. As an example, if rule 222's second premise questions about the knowledge of parameter "p", the system will use the operator "known", which verifies whether a given parameter is known by the system or not. Then, the database's pertinent quadruple will be compared with the operator's CF threshold. In other words, if the threshold is +.2 and the current CF is .1 the operator returns false, indicating that the system considers that parameter as unknown, which answers the premise's question.

Initially, the success achieved by MYCIN would imply in a wide acceptance of the CF model employed in it. However, some recognized drawbacks of this approach made even the MYCIN's developers to consider alternate approaches (Buchanan, 1985). I will not comment these drawbacks, since it is not my purpose to go into this level of detail about the MYCIN's development, for it will not add insight to this work's objectives. In addition, most of these drawbacks will be considered in section 3.5, which conveys the conclusions about "Wise Pilot 1" (that uses the CF model). Nevertheless, the interested reader will find extensive coverage on this subject in the works of Adams (1976), and Buchanan (1985).

The alternate approach for MYCIN developers was Dempster-Shafer's theory of evidence (Shafer, 1976), which can be easily understood through the explanation provided by Schum (1994). "Wise Pilot 1" uses the CF model because of its simplicity and efficacy in the parameter elicitation in the system's domain of knowledge. Besides, that alternate approach has its own drawbacks in a highly dynamic domain (see Buede, 1988, and 1997). Yet, the CF model assures the computational easiness and simplicity essential for a real time application like "Wise Pilot 1" and that resulted in MYCIN's empirical success as an actual expert system.

3.4 The "Wise Pilot 1" Knowledge-Based System

Knowledge systems like MYCIN are conceived for dealing with static situations, where all available information is given before the system arrives at a final solution and usually does not change until then. However, “Wise Pilot 1” has to deal with a highly dynamic situation, where data is changing at a continuous pace. To overcome this incompatibility, a discretization of the continuous time is made by the program. Here, succeeding cycles are executed in a repeated process that relies on a fast processing capability to give a quick output. The pilot would perceive that response as continuous. Every cycle is called ITERATION, and can be thought as a thorough MYCIN section.

One of the firsts steps in “Wise Pilot 1” was to define a context list, where all objects of interest in the system were listed. Table 4 shows these objects and its respective meanings.

Table 4 Context List for the “Wise Pilot 1” Knowledge System

CONTEXT	**DETAILS**
AAA	A TRACK that was identified as an Anti Aircraft Artillery site
ACQ	A TRACK that was identified as an acquisition radar site
AIRCRAFT	A TRACK that was identified as an aircraft
EMITTER	A TRACK that had emissions perceived by the RHAW
ITERATION	First level of the context tree. Is the discrete time interval in which a complete program cycle is executed.
SAM	A TRACK that was identified as an Surface-to-Air missile site
TRACK	An object that is perceived by any of the aircraft’s sensors.

That list does not represent a partition of system's possible objects, which means they are not mutually exclusive. In other words, an object AIRCRAFT can be also an EMITTER and an EMITTER can be an AIRCRAFT, an AAA, or a SAM. All of them are TRACK objects, and so on. What make the distinction amongst objects are their attributes, which are parameters asserting characteristics that are exclusive to a given object.

Because of the discretization process employed by the system, a special context was created for capturing the specific details of each cycle of the program. Table 5 brings these parameters.

Table 5 Possible Attributes for the ITERATION Context.

ATTRIBUTE	DETAILS / EXPECTED VALUES
ALTITUDE	Host aircraft's current altitude.
AUTOPILOT	Autopilot system status / reflects the current mode of the system (i.e. "disengaged", "full", "mode#", etc).
IFF	IFF transceiver status / either "on" of "off"..
ITNUMBER	Iteration Number / a simple counter for keeping track of the Iterations.
RADAR	RADAR status / either "on" of "off"
SAFLFSEC	Safest lateral front sector, defined by a system's internal algorithm / possible values are "2" and "3"
TERR	Territory. Returns "enemy" whenever the aircraft is flying over enemy territory, "neutral" and "foe" are the other possible outcomes. This is a datum based on the aircraft's position that is given by its navigation system.

The importance of ITERATION's parameters resides in the fact that they provide the system with a notion of its current capabilities. As an example, if the parameter IFF is set to "off", when a rule's premise requires some piece of information that could be taken from the IFF the system inference engine's FINDOUT scheme (see Figure 14 on page 68) will not attempt to retrieve the IFF sensor. Instead, it will go directly to the alternative of that procedure. The other sensor parameter of an iteration context is the RADAR.

TERR is an example of parameter that adds information to the system through the host aircraft's navigation sensors. For each iteration, the system (and the fighter pilot as well) will have different expectations about an unknown track depending upon the nature of the terrain it is flying over. In other words, the likelihood of an unknown track being an enemy is usually greater when flying over enemy terrain than it would be when flying over friendly terrain. TERR is the parameter that feeds the system with this iteration-related information.

Another important context in the system is the TRACK, a name that resembles from the radar terminology but has a broader meaning in "Wise Pilot 1" system. Unlike what happens in the radar domain's denotation, where a track is a moving target (usually an aircraft), the system considers every perceived object as a track. This broader definition includes AAA and SAM objects, which are not "tracks" in a radar scope. Table 6 shows the possible attributes that can be found in a TRACK context.

Table 6 Possible Attributes for a TRACK Context Type.

ATTRIBUTE	DETAILS / EXPECTED VALUES
IFFRET	IFF return / “yes” when a friendly response is received by the IFF transceiver, “no” otherwise
LAUNCH	A missile launching has been detected from the TRACK / “Yes” or “No”
SECTOR	Relative sector of the aircraft in which the TRACK is located. Values range from “1” (front sector) to “6” (rear sector). Refer to Figure 16 for the sector partition.
STATUS	“Friend”, “Foe”, or “unknown”
THREAT	“yes” or “no”
THREATLV	Threat Level / 0-5, where 0 means no present danger to the system at the current ITERATION, while 5 is the highest danger level
TRACK#	Track number, where “#” is the assigned number for the track.
TRACKALT	0 – 99,900 Ft (100 Ft increments), “unknown”
TRACKPRJDT	Projected distance calculated by means of radar information.
WEAPONGD	Guidance method employed by the TRACK’s main weapon. “IR” (Infrared guided weapon) or “RG” (radar guided weapon).

Generally speaking, the parameters of a track context are related either to the track’s status or to its physical position. An example of a “status” attribute is the THREATLV (threat level), which quantifies the amount of danger a given track is imposing to the Blue fighter. This attribute can be seen as a “final product” of the system’s inference scheme, since it is the result of many queries made by the system about that track. Indeed, the actions to be taken will depend heavily on this attribute value.

An example of a “physical position” parameter is the TRACKALT attribute, which informs the system about the track’s current altitude. This information can be retrieved either by the host aircraft’s radar or by an external source of information, like a data link with an AWACS aircraft or with a ground control.

However, we assumed in Chapter 2 (refer to page 48) to that no synergy is considered in order to prevent an unnecessary increase in the system’s complexity that would not add insight to this work’s purpose. Yet, the addition of a data link as a virtual sensor in the track attribute list is not a challenging task so an eventual implementation for an actual system might be fairly straightforward.

Another example of a “physical position” parameter is the SECTOR attribute, which carries the azimuth information provided by the RHAW with respect to a given track. Given the inherent impreciseness of this sensor, which relies on the track’s emissions for azimuth discrimination, the space surrounding the blue fighter was divided in six large sectors as shown in Figure 16. Yet, the RHAW is usually calibrated in order to provide a better discrimination at the front and rear sectors, which explains the smaller angular size of these sectors.

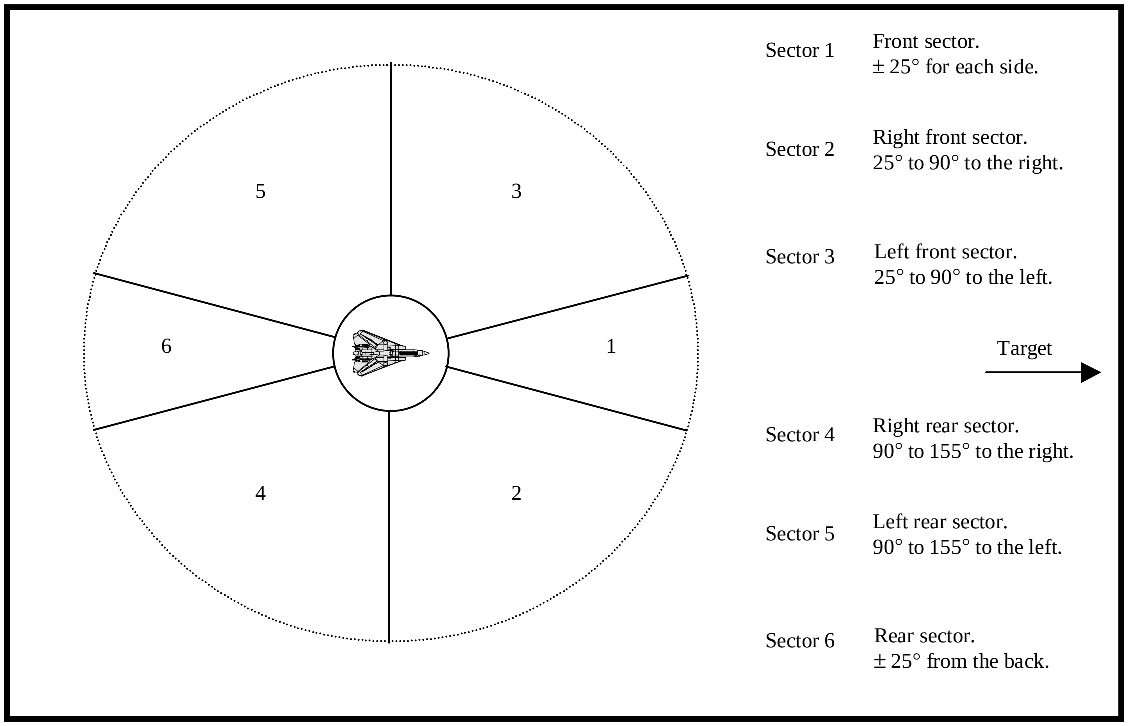


Figure 16 Aircraft Relative Sectors in “Wise Pilot 1” System's Logic.

In an object-oriented programmer point of view, contexts AIRCRAFT, SAM, and AAA can be considered as instances (or children) of a TRACK object. However, I will not get into detail about programming techniques and nomenclature, since the main point here is to present these objects in order to achieve a broad understanding on how “Wise Pilot 1” was conceived and the way it works.

Once a TRACK object is considered an aircraft, it will receive additional attributes that will be valuable for the program’s inference on this type of tracks. Table 7 shows these attributes.

Table 7 Possible Attributes for an AIRCRAFT Context Type.

ATTRIBUTE	DETAILS / EXPECTED VALUES
ACTTYPE	Aircraft Type / (ONEOF (AIRCRAFT), where (AIRCRAFT) is a list of all aircraft that can be found at the Theater. The system's users (EW Officers) shall update this list in a regular basis.
F#	Formation number, next integer available for a new F#
FORMATION	Yes or No, obtained by means of radar information.
FREEFT	Yes or No, obtained by means of radar information.

Among attribute ACTTYPE's details the concept of a list is introduced. This is a toll for making rule-based systems' maintenance an easier task. When a new aircraft is introduced in the scenario, the system's user would have only to add it to the list of aircraft types considered. In addition, there are other lists conveying data on that aircraft, which will also have to be updated in order to allow the system to distinguish it from other aircraft.

An example is the emitter list, which has all the possible emitters a system could recognize. This list is actually a set of possible values of the attribute EMTID (emitter ID), which in its own way is part of the EMITTER object. Table 8 shows all attributes of an EMITTER object, which includes all tracks that use the electromagnetic spectrum in the Theater of Operations. In other words, this context covers all tracks that can be distinguished by its own emissions.

Table 8 Possible Attributes for the EMITTER Context.

ATTRIBUTE	DETAILS / EXPECTED VALUES
ACQRG	Acquisition mode maximum range (nm).
EMTSS	Emitter Signal Strength, possible states are “none”, “low”, “medium”, and “high”.
EMTID	Emitter identification / (ONEOF (EMITTER), where (EMITTER) is a list of all emitters that can be found at the Theater. The system’s users (EW Officers) shall update this list in a regular basis.
LOCKRG	Lock-on mode maximum range (nm).
MODE	The operation mode in which a track’s radar is operating, possible states are “none”, “acq” (acquisition), “trk” (tracking) or “lock”(locked-on).

Another context of the system is the AAA context, which has only three attributes, AAATYPE, AAAEA, and AAARG. The first, like ACFTID and EMITID, is a list that encompasses all anti-aircraft artillery systems that are known by the system. The last two are attributes that convey performance parameters of the respective AAA system, which will be used for assessing its danger level.

Table 9 Possible Attributes for an AAA Context Type.

ATTRIBUTE	DETAILS / EXPECTED VALUES
AAATYPE	AAA Type / (ONEOF (AAA), where (AAA) is a list of all anti-aircraft artillery that can be found at the Theater. The system’s users (EW Officers) shall update this list in a regular basis.
AAAEA	Maximum effective altitude of that AAATYPE, in feet.
AAARG	Maximum effective range of that AAATYPE, in nm.

Similar to the AAA context type, the SAM context type has a “list type” attribute for identification purposes (SAMID), and two performance-related attributes, SAMEA and SAMRG. Table 10 shows the possible attributes for a SAM context type.

Table 10 Possible Attributes for a SAM Context Type.

ATTRIBUTE	DETAILS / EXPECTED VALUES
SAMTYPE	SAM Type / (ONEOF (SAM), where (SAM) is a list of all surface-to-air missiles that can be found at the Theater. The system’s users (EW Officers) shall update this list in a regular basis. If a launch is detected and there was no previous knowledge about SAMTYPE, either “portable” or “unknown” can be assigned to it, depending on the situation.
SAMEA	Maximum effective altitude of that SAMTYPE, in feet.
SAMRG	Maximum effective range of that SAMTYPE, in nm.

In spite of their similar attributes, there is a subtle difference among AAA tracks and SAM tracks related to the track context attribute WEAPONGD. For AAA tracks, this attribute will be automatically set to “RG”, since all AAA rely on radar guidance[8] whereas SAM systems can also be Infrared guided.

WEAPONGD conveys the information about the main guidance mode of the system. Although this data can be used as an auxiliary tool for discrimination (identification) procedures, its main objective is to determine which action should be

8 Actually, there is also the optical guidance option for AAA (and for some SAM systems as well), but I am referring to the system’s main guidance method.

taken against that track. Firing a flare will have no effect against an active radar guided missile, while a chaff cloud would not deceive an IR guided missile.

The last object to be highlighted is the acquisition radar site. Acquisition radars usually perform a wide angle, medium range, search-oriented coverage with the main purpose of detecting airborne intruders and allocating them to fire control radars, which take care of the weapon aiming and guidance procedures. Depending on the system, an acquisition radar can be connected to one or more tracking radars (fire control radars). In some modern systems, like the already famous Patriot system, the same radar platform is used for both acquisition and tracking procedures.

The ACQ context also has a "list type" attribute for identification purposes (ACQID). However, unlike the AAA and SAM cases, there is only one attribute regarding the system performance spec, ACQRG. Table 11 brings the two possible attributes for the ACQ context.

Table 11 Possible Attributes for an ACQ Context Type.

ATTRIBUTE	DETAILS / EXPECTED VALUES
ACQID	EW identification / (ONEOF (EW), where (EW) is a list of all acquisition radars that can be found at the Theater. The system's users (EW Officers) shall update this list in a regular basis.
ACQRG	Maximum effective range of that ACQID, in nm.

Defining the possible objects, attributes, and their respective values provided us with a basis for beginning the construction of the list of rules, which must convey all the cumulated expert knowledge available in that domain. Usually, the list of rules is a numbered list that follows the most convenient classification criteria. In “Wise Pilot 1” I separated the rules in accordance to the context of interest of each rule, so there will be iteration rules, aircraft rules, AAA rules, and so on. This approach facilitated the configuration management of the list of rules, a key issue when developing a system supposed to gather information from many experts.

The system was not conceived to be a comprehensive solution for the pilot’s problem. Instead, its main purpose was to understand how a rule-based system would behave in this domain, what are its limitations and strengths, and how adequate it is in solving the proposed scenarios.

Aiming at the proposed objectives, I used my on expertise and gathered opinions of some experts from both USAF and BAF for defining a set of rules that would be applied to the scenarios. It is important to note that even the resulting tiny set of rules demanded a great amount of men-hours of work, and a complete system would require a large programming structure backed with good system engineering techniques. The rules used for solving the scenarios are in Appendix A.

After concluding on a minimum set of rules, I applied them to both scenarios and analyzed the system behavior and the solutions presented for each case. Appendix B is a condensed explanation on the system’s steps on a rule-to-rule basis, bringing the backward chaining process lead by MONITOR until the solution was finally reached.

3.5 Analysis of the Solutions

As stated before, the purpose of the “Wise Pilot 1” system is just presenting an illustration on how a knowledge-based system could be used in order to solve the given dynamic scenarios. In light of these requirements, developing a real system would be an overkill approach. Nevertheless, “Wise Pilot 1” did accomplish its mission of providing means for a general analysis on the suitability of an extensional system for solving a complex dynamic problem.

Standing on top of the noticeable advantages of “Wise Pilot 1” design is the fact that it did provide a fairly functional automated weapon and sensor allocation system, which is a clear progress in comparison with a human-only decision making scheme. In addition, a real system with the same design would provide easy maintenance and modifiability for its users, since most improvements would be done with a simple alteration in the program's lists.

Rules are also modifiable, which makes the system easily adaptable for changes in operation doctrine or more profound issues concerning the employment of the aircraft as a weapon platform. Another positive point is the relative easiness in assessing the experts’ opinions, since most of the questions made were related with the day-to-day experience on their domain of operation. In other words, most questions looked like: "if you see this, in that situation, what would you do?"

In spite of those advantages, there are some pitfalls that may prevent an effective use of the knowledge system's approach for that particular problem, as explained in the following paragraphs.

- *Only local evidence is considered in the reasoning process.*

This is a direct effect of the static nature of its extensional approach. In the original MYCIN everything is solved in a single section, and every section is completely independent of each other. Likewise, every system's iteration is a discrete independent single section, carrying only some parameters defined on the closest previous iteration .

Although this independence is useful for keeping the modularity of the rules, a restrictive side effect is that the scheme prevents the use of evidence that is spread on non-concurrent events, which is fairly common in dynamic situation modeling.

As an example, consider a situation in which knowledge about an emission of radar "E" could imply in the existence of an "X" fighter, with probability of .7 (70%), or in the existence of an "Y" fighter, with probability of .3 (30%). Suppose that evidence of radar "E" is detected during a given iteration and the program assigned it as a support for hypothesis “X” fighter. Now, suppose that during other iteration, new evidence shows that of the program and after some iteration, hypothesis “Y” fighter is confirmed.

Initially, the system used radar “E” emission for increasing both “X” and “Y” hypothesis. Since we learned that hypothesis “Y” is true; it is intuitive that the support of emission of radar “E” to hypothesis “X” decreases. In other words, since there is an “Y” fighter flying around for sure and “Y” fighters have radar “E”, receiving an “E” emission is an evidence that does not support the existence of an “X” fighter as it would in the case of uncertainty about an “Y” fighter flying around.

Rule based systems do not provide an automatic updating for this conditional dependence among evidence and hypothesis. It is true that adding rules for dealing with this updating in a domain with few cases of conditional dependencies could solve the problem. However, a complex problem usually has many concurrent conditional dependencies, making unfeasible that palliative approach.

- *Excessive number of rules required for covering all possible situations.*

As the number of lower level branches in a decision tree grows geometrically with the number of possibilities per node, the number of rules necessary to cover a single situation grows with the number of premises to be checked in the same pace. This fact, when confronted with the high number of variables to be verified in a dynamic decision process would generate the necessity of an extremely large number of rules in order to perform an exhaustive decision making process.

An example of this issue inside "Wise Pilot 1" is given by the sequence between Rule 327 and Rule 338. Here, twelve rules with repetitive premises were necessary to cover a situation involving only three variables. Each of the first two variables have binary outcomes ("trk" or "acq", and "yes" or "no"), while the third may assume only three possibilities ("high", "medium", or "low"). If we want to have more complex variables with more possibilities, a fairly reasonable situation in dynamic systems, the number of necessary rules (and premises) would grow really fast. Therefore, a bottleneck may be created regarding the real time requirements of the system, since the program's inference engine has to check all premises.

- *Excessive number of premises that a rule would have to have in order to catch the specific details on each situation.*

As another consequence of the dynamic nature of the situation portrayed in the scenarios, usually it is very hard to cover all the possibilities in which a track can appear. This characteristic leaves the system with two alternatives for framing the situation, it can either use a large number of premises in order to cover every possible situation or concentrate only on some of the most important ones.

“Wise Pilot 1” uses the second, as we could see in some of its rules. As an example, although Rule 14 has seven premises, it certainly does not cover all the variables that an X fighter could have. Issues like formation size and type, flight path, velocity, sector in which it is coming from, and others that are mentally considered by the pilot are not considered in Rule 14. Considering all these issues would make “Wise Pilot 1” too big of a system for running in real time.

Even though the group of rules in which Rule 14 is inserted does a fair job in defining X fighter's THREATLV, a more comprehensive coverage would require a lot more of premises. As explained in the previous paragraphs, those extra premises would certainly demand more rules, which will decrease the system's performance.

- *The discretization process forces a trade off between the number of rules and the time required for each iteration.*

This is a clear consequence of the two previous weaknesses, since the demand for more rules will have to be limited in order to maintain the real time response requirements of the system. The backward chaining system does not provide a fast algorithm for dealing with a large number of rules in real time. Nevertheless, every iteration has to be executed within a limited time frame.

Depending upon the required response time of the system, even a powerful dedicated computer will not provide resources needed to check a huge number of rules.

In face of that constraint, a trade-off between real time requirements and the maximum number of system's rules shall be made.

- *The CF model does not provide a flexible way to deal with dynamic uncertainty.*

The CF model of uncertainty focuses on how an evidence increments (or decrements) the knowledge on a certain hypothesis, leaving no further consideration on the alternatives of that hypothesis (for the sake for modularity of the rules). As a direct consequence, those alternatives have to be considered by creating other rules on that same evidence, since the model does not allow multiple probability assignments on the conclusions of a single rule.

We can easily perceive this fact when we have evidence that supports mutually exclusive hypothesis, which is fairly common in sensor fusion issues. As an example, a radar that could be linked to one of three different systems with known probabilities (say .6, .4, and .2) would result in a rule stating only about one of them.

A probable rule for that example would be “IF radar ‘R’, THEN system ‘1’ (.6)”. Although the information on the probability of system ‘1’ (.6) is conveyed by the rule, the same is not valid for the others. Hence, if the probability assignments for the alternative hypothesis (i.e. .4 and .2) are not considered somewhere else in the program, They will be lost and therefore would not be used in future assessments.

Another example of the lack of a proper structure for dealing with probabilities is commented in page 186 of Appendix B. There, we could see that when the system learns that TRACK2 is an “Y” fighter, the probability of this track being an enemy would change to .25. However, the CF system does not provide proper means for automatically updating these probabilities.

As a conclusion for the application of a rule-based approach in the proposed scenarios, I would say that resulting system provided doctrinally correct answers in both cases and in real time. However, during the development of the system, I could perceive the above-cited pitfalls when trying to extend the system’s coverage for situations other than the portrayed in the scenario.

In this attempt to make the system more comprehensive, the lack of a consistent probability system proved to be a major obstacle for modeling complex situations with many conditionally dependent variables. I would say that even with up-to-date computational capabilities, these deficiencies would prevent the creation of a definitive substitute for a human-based decision system in this domain.

CHAPTER 4

THE PROPOSED APPROACH

4.1 Setting the Stage

The Decision Analysis theory applies the “divide and conquer” strategy for solving complex problems (Henrion, 1991), where the issue is divided in smaller, more manageable parts. This approach makes easier for the decision system's builders to achieve a computing scheme able of mimicking, while outpacing human thought.

The start point of such a scheme is to define the objectives and values that will guide the decision process. Keeney and Haiffa (Keeney, 1976) pointed out that any serious theorist on the objectives of a problem has come up with some sort of hierarchy of objectives. Following this line of thought, the first effort in the DDN process is to determine and classify the objectives that guide the pilot's decisions.

When confronted with a situation like those portrayed in the scenarios, a fighter pilot initially has to analyze the aircraft's sensor data, assessing the degree of danger each track is imposing on him. Then, based on his findings about each track's danger and on his experience, he will assess the possible outcomes of the decisions he has to make and decide what is the best set of actions to perform in order to accomplish the mission's objective, we will call this a decision policy.

However, accomplishing the mission's objective is still too abstract an idea for defining a decision policy. As an example, firing a missile against a given track does impact the likelihood of accomplishing the mission, but how this impact could be assessed? At first, it is necessary to consider how this missile action will improve the pilot's chances of survival; attacking a track would eliminate the danger provided by it, but would also increase the danger provided by other tracks. In addition, there may be chances of that track being a friend, which will make the missile action a fratricide fire. Ultimately, the pilot has to consider how that action will impact his ability to reach the mission's target, another important issue for accomplishing the mission in a successful way.

Considerations on those aspects shall be raised for each decision the pilot has to make. The impact of a decision on the overall mission's objective can only be realized by weighting its effect on each of those three parameters. Adopting a nomenclature used in most literature on decision theory (e.g. Keeney, 1992, Kirkwood, 1996), these parameters can be seen as fundamental objectives of the case study's problem. Then, a fundamental hierarchy of objectives for the pilot's decisions is the one depicted in Figure 17.

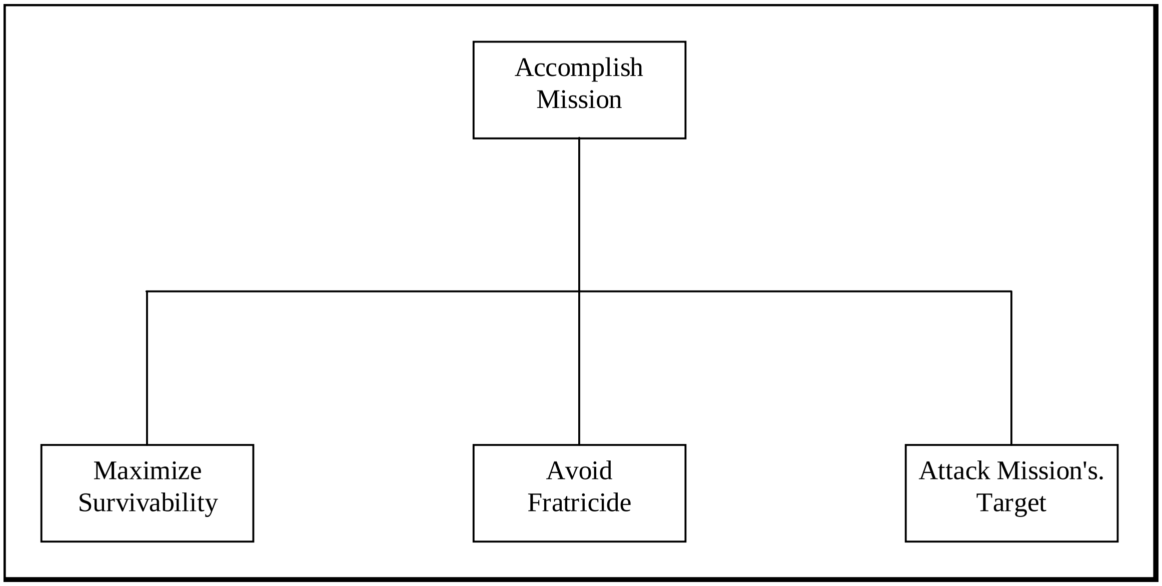


Figure 17 Pilot's Objective Hierarchy.

Since the DDN system is intended to mimic the pilot's decision-making scheme, that same objective hierarchy will be used. In order to provide a straightforward explanation on how “Wise Pilot 2” works, the system's features will be presented in the same order as they occur chronologically in the pilot's mental decision-making process.

In order to take advantage of the DDN features, every real-time iteration of the system's reasoning process will contain a separate Bayesian Network evaluating each individual track. Then, all track-assessed data is to be merged and weighted in a unique influence diagram, which will infer the best decision policy to be taken. Among other things, this policy will determine which sensors will be available for the next iteration, along with its respective mode of operation.

4.2 The “Wise Pilot 2” Dynamic Decision Network System

4.2.1 The Decision Sub-Net

Since the DDN system is supposed to mimic an experienced pilot's decision process, a good point to start structuring its decision part is the pilot's objective hierarchy (Figure 17, page 94). Thus, the main value node of the system's influence diagram is called Accomplish Mission, and is molded by the confluence of three nodes: Attack Mission's Target, Avoid Fratricide, and Maximize Survivability. All of them have only two states (yes or no), and the degree of belief achieved on each will have a direct impact on the value node. The translation from the pilot's value structure to the DDN system is shown in Figure 18.

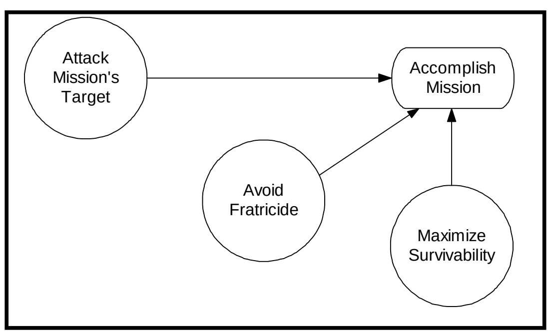


Figure 18 System's Value Structure

To reach this objective, the pilot has to make a set of decisions that we may call strategy policy. Basically, the decisions will be about whether to take defensive/offensive actions and sensor management. “Wise Pilot 2” will cover four types of decisions, each being represented by its respective node in the system's influence diagram.

The first decision is concerned with the aircraft's self defense. Depending upon the situation, the pilot can take four different actions: perform a break maneuver while attempting to deceive the opponent weapon's guidance system, executing the same deception but only deviating the aircraft (instead of a break maneuver), deviate without use of deception features, or taking no action.

The break maneuver is a curve with the maximum angular acceleration achievable at that time. The objective of such a violent move is to avoid being hit by a fast approaching missile. Loosely speaking, it is an attempt of turning at a higher rate that a missile can afford. However, due to the strong maneuvering skills of current missiles, that trick can only succeed when applied simultaneously with the use of a decoy.

The purpose of the decoy is to seduce the missile's guidance system, preventing it to perceive the aircraft's abrupt change of direction. For IR missiles, this decoy is called “Flare” and can be viewed as a deceptive sequence of fireballs that is launched from the aircraft tail. Those fireballs burn at the same IR frequency of the aircraft's engine nozzle. For active missiles, the decoy is a highly sophisticated sequence of electronic pulses emitted in the same frequency as the missile's emitter. Roughly speaking, those pulses are synchronized in a way that gives a wrong hint on the aircraft's range to the missile's guidance system.

Nevertheless, that option has some pitfalls. By using all of its energy for a maximum turn, the aircraft become vulnerable to further attacks and may not have energy enough to do it again. In addition, decoys are inherently active measures so they will act as a beacon for prospective interceptors. Because of that characteristics, the break & decoy maneuver is to be considered as a "last chance" to escape from a near impact.

The second and third alternatives are less radical in terms of maneuver, since a simple direction or altitude change can achieve the required deviation. Other factors like the desire in maintaining a level of stealthiness will dictate whether this deviation will be executed concurrently with a decoy or not.

Deviating from a threat may impact the aircraft's capacity of getting to the mission's target, while launching a decoy will clearly increase the vulnerability of being detected. Thus, it is fair to say the defense action to be taken will have a direct impact on both Attack Mission's Target and Detectability nodes. Finally, sometimes doing nothing (the fourth alternative) may have a greater return in light of the current situation.

The next decision to be taken is whether to launch a missile against a threatening track, only to engage on it, or to take no missile action. Because of the initial assumptions in the scenarios[9], only air-to-air missiles will be considered so there will be no missile action against ground tracks. Engaging on a track means to lock the aircraft's radar on it,

[9] According to the scenario's definition, the aircraft is not equipped with air-to-ground missiles; however, adding this option to the DDN structure is not a hard problem. Actually, easy updating is one of the advantages of this approach.

which is the same action as launching a missile but without pressing the trigger. This will put the enemy's pilot under a lot of pressure, since he will not be able to distinguish whether the missile was actually launched or not. A missile action against a track will have a direct impact on node Avoid Fratricide, since the decision to launch will probably cause the annihilation of that track.

Following the defensive/offensive actions, the system will have to decide about the sensor management. Initially, it will have to choose whether to turn the radar on at a higher mode (necessary for missile actions), a lower mode (with low probability of detection), or turn it off. Then, the last decision is whether to turn the IFF on or off. Finally, as a requirement for a proper influence diagram, there has to be a path that begins in a decision node, passes through all decisions, and finishes in the value node (refer to rule 4, page 27). Figure 19 shows the inclusion of the decision nodes in the influence diagram's value structure.

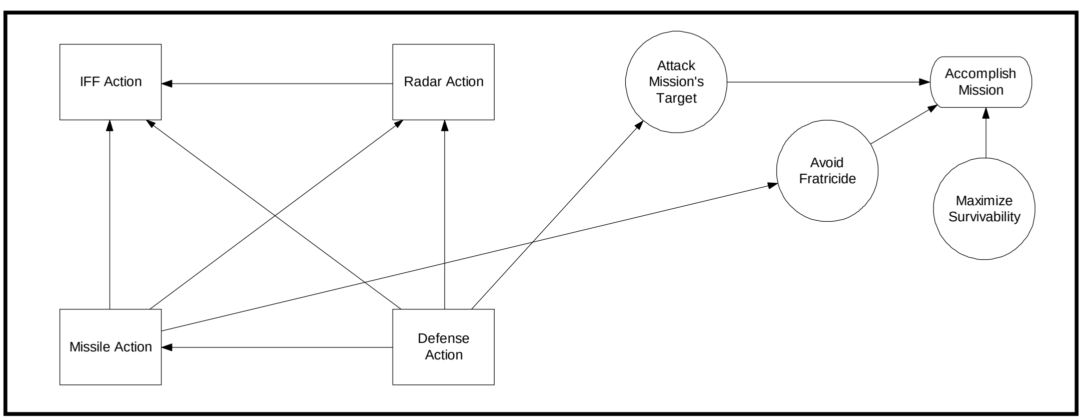


Figure 19 System's Value Structure and the Decision Nodes

However, there are parameters that will also have an impact on the value structure nodes directly, thus influencing the main objective node in an indirect fashion. These nodes are Detectability, Aircraft Awareness, and Deny Attacks.

Detectability is concerned with the aircraft vulnerability of being detected by the enemy, its preferable state is “low” and the less preferable state is “high”, while “medium” stands in the middle. Being detectable does not change directly the beliefs of avoiding a fratricide or attacking the mission's target, but will have a great influence on the aircraft's survivability. Thus, a direct link exists between nodes Detectability and Maximize Survivability in the system's influence diagram.

Aircraft Awareness may be considered as the opposite from detectability, since it is related with the system's capacity of detecting enemy aircraft[10]; this time, the most preferable status is “high”, followed by “medium” and “low”. A better knowledge on the aircraft in the scenario will increase one's chance of firing against a friendly aircraft, and will also allow increase its chances of survival. Thus, node Aircraft Awareness has a direct link to both Avoid Fratricide and Maximize Survivability nodes.

After establishing the connections of these newly added nodes with the value structure, one can easily realize that it is also necessary some connections between these nodes and the decision nodes. As an example, the Radar Action decision node will have

---

[10] Awareness of ground tracks is not a concern to the system, since there is no decision taken by the system may impact it. This happens because of the fact that the RWR and MAW are always on, and the radar and IFF are related to aircraft only.

an unequivocal impact in both Detectability and Aircraft Awareness nodes, and the conflicting interests of these nodes will cause the need of a trade off between them in order to define which option to take

Like the radar decision, it will impact both Aircraft Awareness and Detectability nodes; consequently, the IFF decision will also be based on a trade off between the interests of these nodes. However, the difference is that the IFF decision will be made after the radar decision, so its outcome will also be a function of what has been decided with respect to the radar.

Deny Attacks is a result of the data fusion process of all tracks' respective Bayesian Networks. Denying enemy attacks will allow the aircraft to get to the mission's objective, since enemy attacks would damage the aircraft's navigation and weapon systems. In addition, it is also obvious that denying attacks increase the probability of staying alive, thus a connection between nodes Deny Attacks and Survivability is also Needed.

Now we have completed what we would call the decision sub-net of the dynamic decisions network system. From section 2.3.2 we know that we have modeled how the fused data from the tracks will trigger the system's decision, so the next step is to uncover how does that fused data arrives in the decision sub-net. For establishing a visual landmark, Figure 20 brings a view of the decision sub-net.

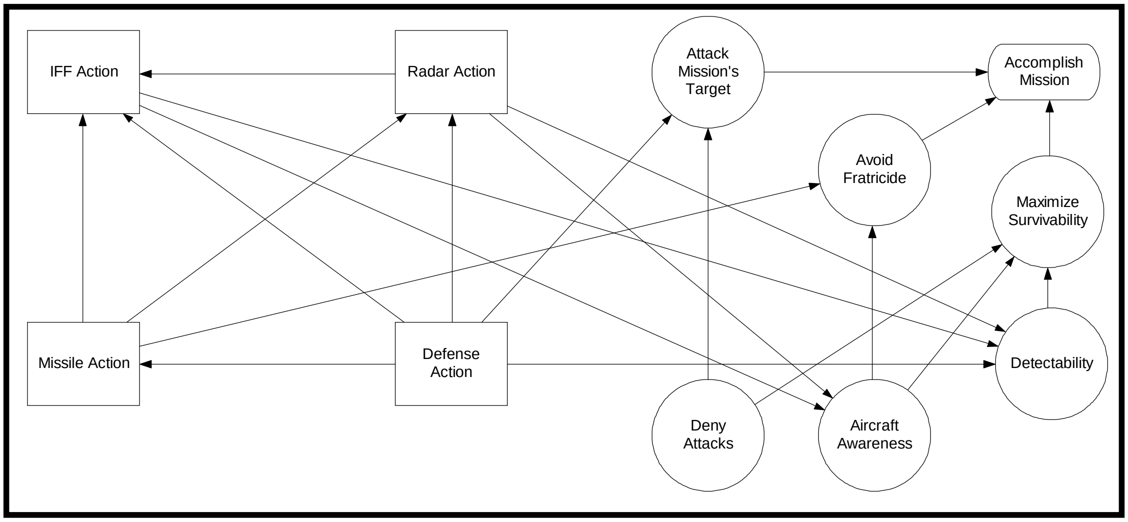


Figure 20 The Decision Sub-Net of the DDN System

4.2.2 The Data Fusion Sub-Net

After collecting and processing all data on the tracks, the Bayesian Networks of the DDN system's inference sub-net will propagate each track's assessed beliefs two data fusion nodes: Avoid Ground Attack and Avoid Aircraft Attack. Thus, there will be a set of two nodes for each Bayesian network in the inference sub-net, in other words, two nodes for each track detected by the system's sensors.

The new beliefs of all Avoid Ground Attack nodes are then propagated to node Deny Ground Attacks, while all Avoid Aircraft Attack nodes will pass its belief to node Deny Ground Attack. Figure 21 shows this data fusion process for a two tracks case.

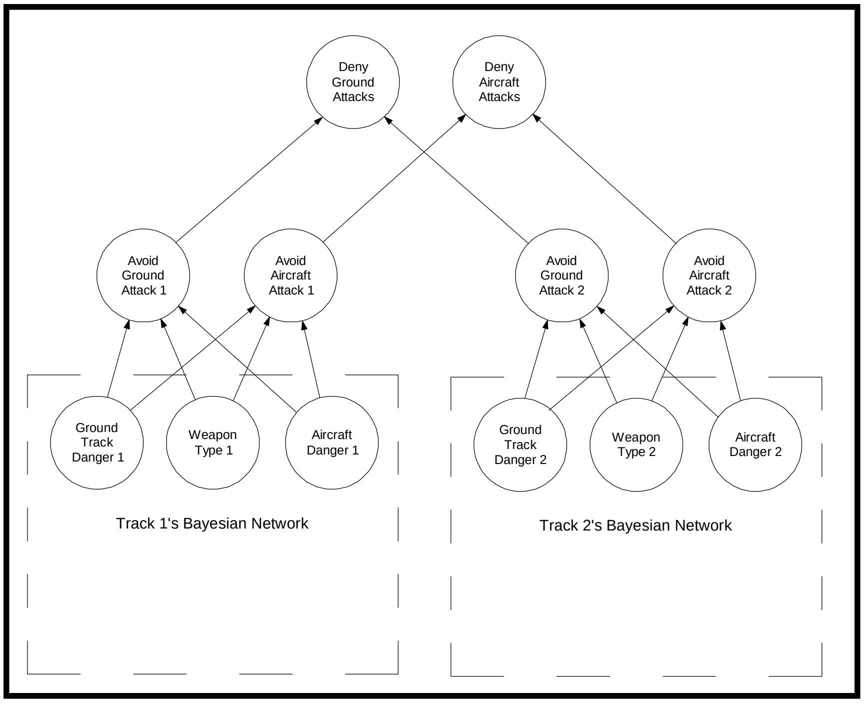


Figure 21 Data Fusion Process in the DDN System

Node Deny Attacks merges the beliefs of both Deny Ground Attacks and Deny Aircraft Attacks nodes. However, it is clear that a given degree of awareness on all aircraft in the scenario will be directly related to the capacity of denying aircraft attacks, while the same does not hold for the capacity of denying ground attacks (recall footnote 10, page 99). Because of that characteristic, there will be a link between nodes Deny Aircraft Attack and Aircraft Awareness.

Figure 22 shows how the data fusion process is embedded with the decision's parameters and value nodes. In addition, a link between node Altitude and nodes Aircraft Awareness and Detectability is also shown, since both are affected by the aircraft's current flight level. To avoid cluttering the picture, links between node Altitude and all BNs is not shown, despite their obvious existence.

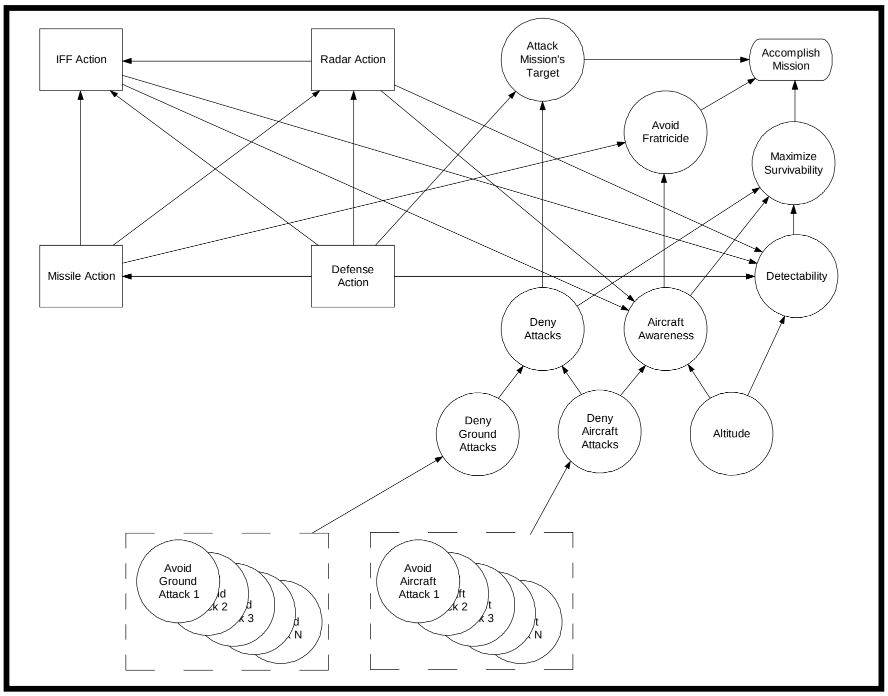


Figure 22 Linking the Data Fusion and Decision Parameter Nodes

Yet, in order to uncover all the connections between the data fusion sub-net and the decision sub-net, some remarks should be made on the impact of the decisions in the data fusion process.

The way an alternative contributes in avoiding attacks from the various targets is a key issue; each defense action will have a different impact for every track, which explains the connection between the defense action decision node and each track's Avoid Ground Attack and Avoid Aircraft Attack nodes. Hence, the number of states in the defense action decision node will vary in accordance with the number of tracks, an evident illustration of the dynamic characteristic of the DDN technique.

In current up-to-date weapon systems, only one missile can be launched at a time (i.e., one for each iteration of the DDN). Thus, the impact of a missile action also has to be measured individually for each track, which is achieved by a direct link between node missile action and each track's Avoid Aircraft Attack node. This last connection finalizes the merge between the decision and data fusion sub-nets of the DDN system, depicted by Figure 23. Our next step is to find out how the treated external data arrives at the data fusion nodes.

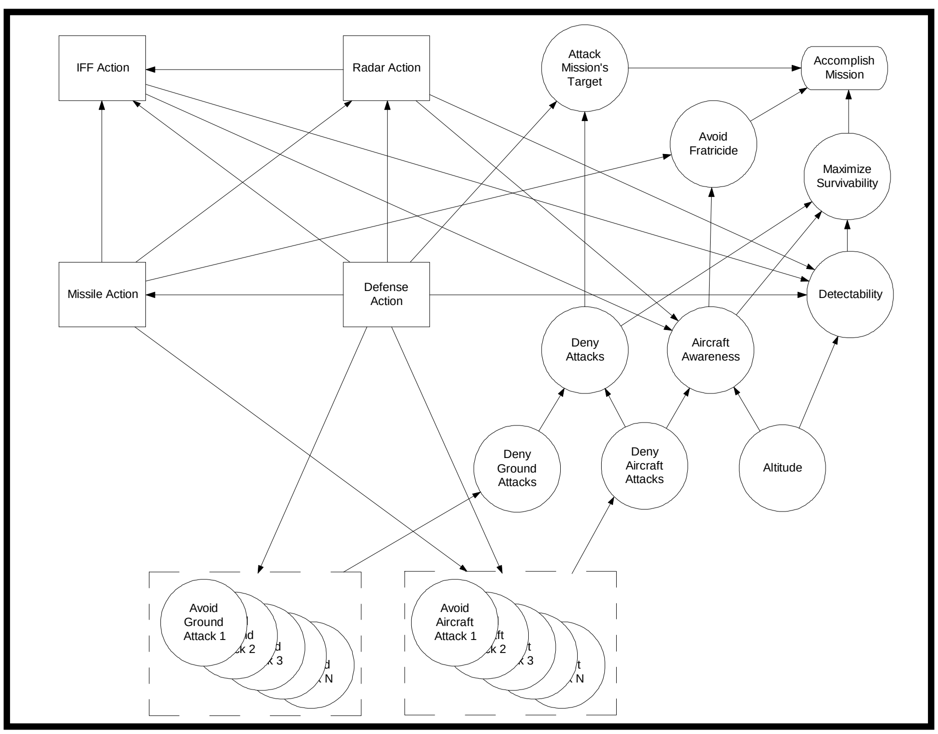


Figure 23 Decision and Data Fusion Sub-nets of the DDN System.

4.2.3 The Inference Sub-Net

Initially, we would consider what sort of data each Bayesian Network receives and produces when evaluating a particular track. Figure 24 shows the general flow that goes through the assessment process.

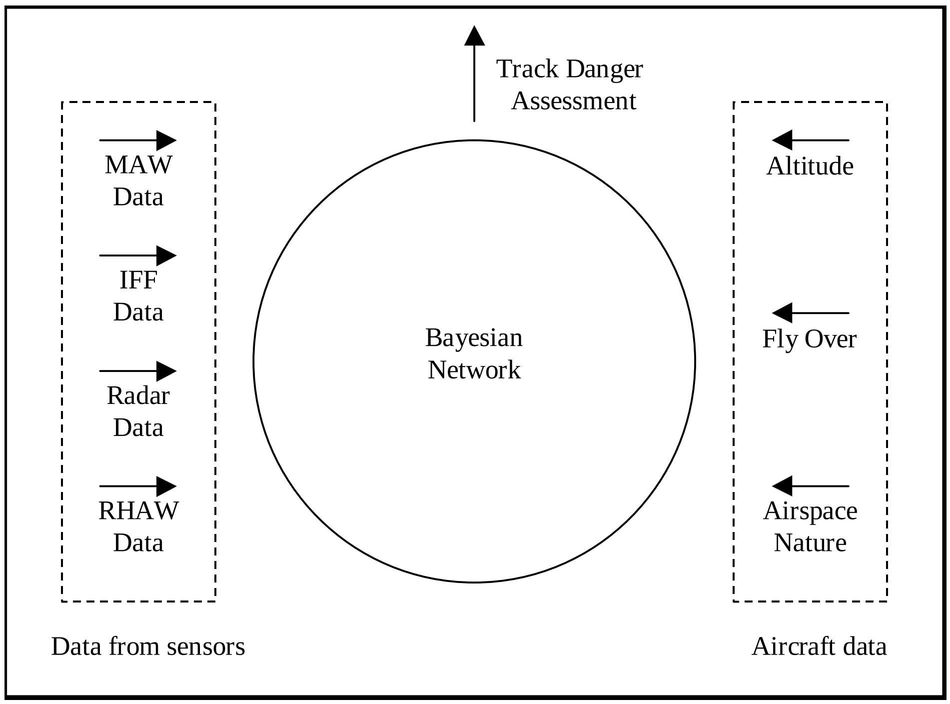


Figure 24 General Track Danger Assessment Scheme

Each Bayesian network in the DDN system will receive two types of information about its respective track of interest: sensor data and aircraft data. The first comprises all track data that was gathered by the aircraft's sensors, this type of information acts as likelihood evidence vectors feeding the BN. Those vectors will be refereed in this work as “reports”, since it contains sensor-selected data that is specific to the track of interest of that BN. The latter is related to data about the aircraft itself (the one that carries the system), which can be considered as available with certainty and acts as prior evidence vectors in the BN's bi-directional propagation.

Figure 25 brings a more detailed view on how the external data is exchanged among each track's Bayesian Network, the aircraft's sensors and navigation system, and the DDN data fusion nodes. In essence, each Bayesian Network will use the knowledge contained in its structure (node's interrelationships and probability distribution functions), and the aircraft's positioning vector (prior data), to transform information available from aircraft's sensors (posterior data) into a reliable assessment on the danger imposed by its respective track.

This assessment will then be propagated to the DDN system's data fusion sub-net. As shown in Figure 25, three nodes are responsible for this interface: Ground Track Danger, Aircraft Danger, and Weapon Type. The first two nodes have four states (high, medium, low, and none) representing the estimated danger imposed by either a ground-based or airborne threat, while the third has three states (IR guided, Radar Guided, and No Weapon), representing the type of weaponry associated to that respective threat.

Obviously, the mutual exclusive relationship among the two first nodes implies that once a track has its Ground Track Danger node rated as high, medium, or low, it has to have a “none” rating for the Aircraft Danger node. Table 12 brings a summary of the nodes in each track's BN that exchange information outside the BN.

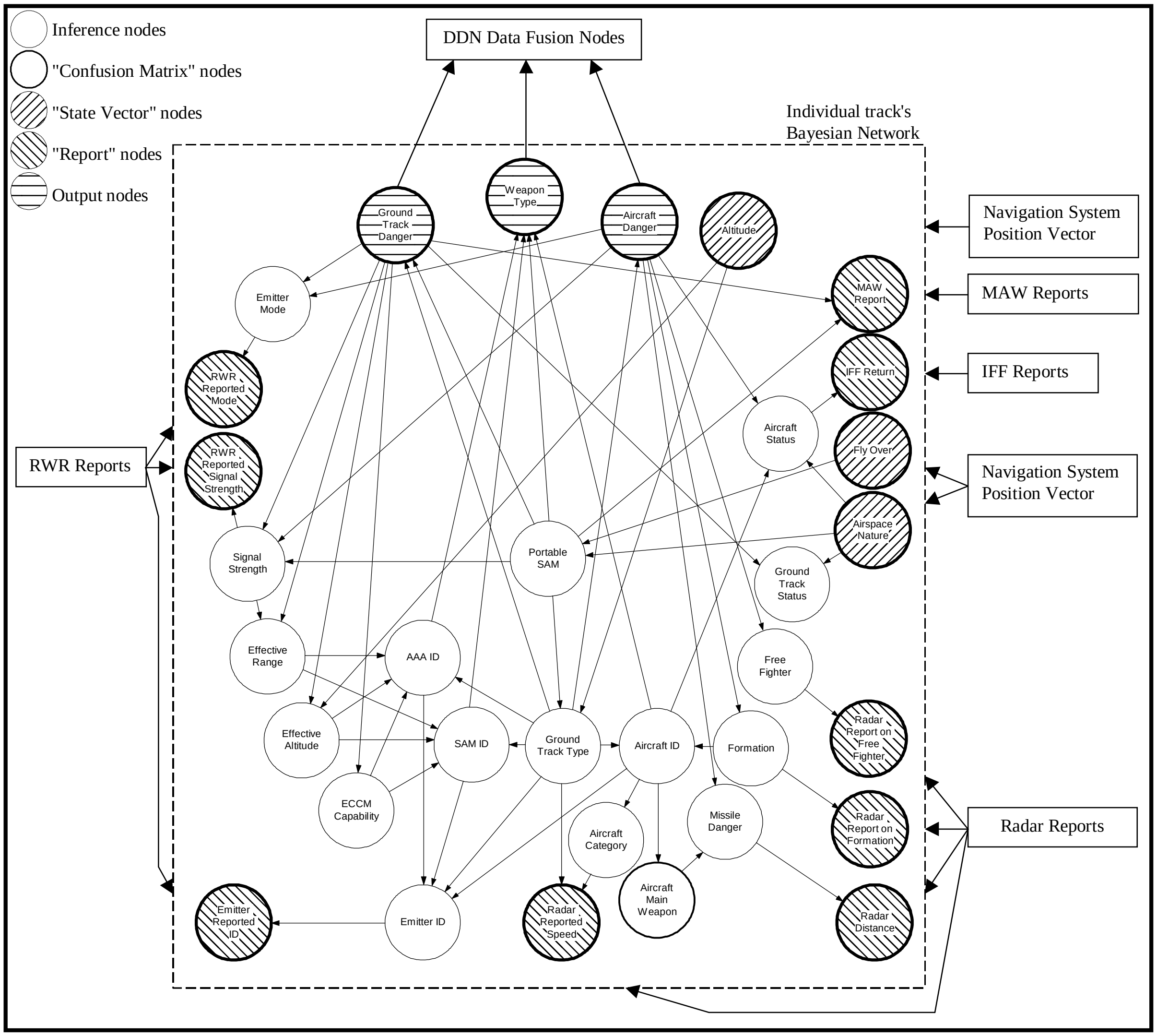


Figure 25 Individual Track's BN Information Exchange Scheme

Aircraft's main navigation system provides data to four nodes: Altitude, Fly Over, Airspace Nature. The first is intended to “tell” the BN whether the aircraft is under flight level 050 (5,000 ft), between it and FL 100 (10,000 ft), between FL 100 and FL 200, or over FL 200 (20,000 ft). Fly Over, as the name implies, informs the BN whether the aircraft is flying over land or sea (the node's two states). The former node, Airspace

Nature, conveys the information about the "political" nature of the terrain where the aircraft is currently flying over, which may be friend, enemy, or neutral.

Table 12 Nodes that Exchange Information Outside of the BN.

Node Name	Receives From	Sends To	States
Aircraft Danger	BN internal nodes	DDN data fusion nodes	High, Low, Medium, and None.
Ground Track Danger	BN internal nodes	DDN data fusion nodes	High, Low, Medium, and None.
Weapon Type	BN internal nodes	DDN data fusion nodes	IR guided, Radar guided, and No Weapon
Fly Over	Navigation System	BN internal nodes	Sea or Land
Airspace Nature	Navigation System	BN internal nodes	Friend, Enemy, or Neutral (terrain)
Altitude	Navigation System	BN internal nodes	Below FL050, Between 050 & 100, Between 100 & 200, and Above FL200.
RHAW Reported Mode	Aircraft's RHAW	BN internal nodes	Locked, Tracking, Acquisition, and No Emissions.
RHAW Reported Signal Strength	Aircraft's RHAW	BN internal nodes	High, Low, Medium, and None.
RHAW Reported ID	Aircraft's RHAW	BN internal nodes	List of emitters present in the theater of operations
MAW Report	Aircraft's MAW	BN internal nodes	Launch or No Launch (detected)
IFF Return	Aircraft's IFF	BN internal nodes	True or False (return of a signal)
Radar Report on Free Fighter	Aircraft's Radar	BN internal nodes	True or False (track is a free fighter)
Radar Report on Formation	Aircraft's Radar	BN internal nodes	Two, Three, More than four, or No Formation
Radar Distance	Aircraft's Radar	BN internal nodes	Less than 23 nm, Between 23 and 27, Between 27 and 40, Between 40 and 44, and More than 44 nm.
Radar Speed	Aircraft's Radar	BN internal nodes	Below Mach .8, Between .8 and .99, and Above Mach .99

Aircraft data nodes have fairly broad states, which means that highly precise equipment like INS and GPS are more than enough to provide data within the required accuracy for those node's purposes. Sensor reports, in contrast, have to gather information from entities other that it's own platform (the aircraft), most of them being non-cooperative tracks preventing the collection of stable data. As a consequence, data accuracy is considerably lower than that provided for aircraft nodes.

Also in Figure 25, four nodes receive information vectors (reports) from the aircraft's radar: Radar Distance, Radar Report on Formation, Radar Report on Free Fighter, and Radar Reported Speed. It is interesting to recall that radar information is used for aircraft tracks only, no information on speed or distance of ground tracks is provided[11]. Nevertheless, absence of radar information, as we may infer from the system's structure, does consist of evidence of a non-aircraft track.

The Radar Distance node is intended to discriminate how far the threat is with relation to missile launch decision's limits (recall chapter 2, pp. 45). Radar Reported Speed node is concerned only with three broad velocity ranges in which a track may be: Above Mach .99, Between Mach .8 and Mach .99, and under Mach .8. The first is suitable for supersonic aircraft only, the second for both supersonic and high subsonic aircraft, and the former for all aircraft and tracks with no information on speed.

Track distance and velocity are attributes that an aircraft's radar can provide with great precision. However, all other “report” nodes receive information that cannot be

[11] Actually, radars are used against ground targets. However, this use is made under specific radar modes that are more related to weapon guidance purposes, like electronic illumination or range finding. These issues would not add insight to “Wise Pilot 2” DDN-based system and are not related with the scope of this work.

considered accurate enough to be used in BN's reasoning process. This creates a problem, since information from those reports is clearly relevant in assessing a track's degree of danger, but it can not be treated as perfect information.

As an example, the fact that the aircraft's RHAW is reporting a medium strength emission from track “X” does not mean it is actually so. Instead, a more legitimate interpretation is that there is strong evidence (assuming a reliable RHAW) that track “X” is emitting a medium strength signal. This difference of meanings will surely have an impact in the final BN assessment; thus it cannot be neglected.

Fortunately, there is a way of getting along with this problem. Instead of considering the reports as perfect information and applying it directly to other BN nodes, we may use an intermediary node that takes account of the equipment's impreciseness. Buede and Girardi (1997) refer to this special node as “confusion matrix”, regarding the probability distribution matrix between that node and its respective report node. This special node will have the same states as it respective report node, and the probability distribution between them will reflect how precise a report is.

Table 13 shows an example of confusion matrix. According to this example, if the RHAW report says that track “X” is emitting a medium strength signal, the likelihood vector that will be passed to the BN internal nodes is [.07 .90 .17 .02]. Equipment's impreciseness is tacit when we compare this vector with the one that is provided when “perfect information” is available ([0 1 0 0]).

Table 13 RHAW Signal Strength Confusion Matrix

RHAW Reported Signal Strength				Signal Strength (actual)
High	Medium	Low	None	
0.90	0.07	0.01	0.02	High
0.04	0.90	0.04	0.02	Medium
0.02	0.17	0.63	0.18	Low
0.005	0.02	0.025	0.95	None

In "Wise Pilot 2", all report nodes but Radar Distance and Radar Reported Speed nodes have a confusion matrix. However, confusion matrices for nodes IFF Return and MAW Report are not explicit; instead, they were absorbed by the adjacent nodes. Nodes that work strictly as confusion matrices in the system were shown in Figure 25 with their borders in bold typing.

4.2.3.1 Assessing a Ground Track

In order to classify a ground track, the system will consider three exhaustive, mutually exclusive types of land-based threats: AAA, SAM, and Portable. Although the last two types include the same kind of ammunition, the difference resides in their guidance (how they go after the aircraft). The system considered as SAM threats employing some kind of emissions in order to guide their missiles, while non-emitting threats (mostly IR guided missiles) are classified in the Portable category. The last category's name is derived from the fact that practically all non-emitting threats are IR guided missiles that may be launched from the shoulder of an infantryman. Each of the three ground threat categories has a node containing a list of all threats that may be found in the theater of operations. As an example, case study's node AAA ID has states: AAA1, AAA2, AAA3, and none (recall Table 2, page 39).

A reliable way of classifying emitter-type threats is the RHAW's emitter ID report. Thus, nodes SAM ID and AAA ID are directly linked to node Emitter ID (the confusion matrix of RHAW Reported ID). However, airborne tracks also send emissions, so the node Ground Track Type was added in order to increase the level of certainty about what kind of threat send a given signal. This node interconnects all emitter-related notes (AAA, SAM, and Aircraft ID) plus the Portable node, and uses information on both Radar Reported Speed[12] and Altitude nodes for discrimination purposes.

After assessing the identification of a SAM of an AAA threat, the next step is to calculate its level of danger. The system uses three nodes as parameters of effectiveness of a ground weapon system: Effective Range, Effective Altitude and ECCM Capability. All three are connected to the Ground Track Danger node, which means that the level of harm a SAM or AAA weapon system may cause will be dictated by the beliefs contained in those nodes. Nevertheless, the kind of weaponry is directly assessed from the SAM ID and AAA ID nodes, as they are straightly connected to node Weapon Type. This arrangement can be checked in Figure 26.

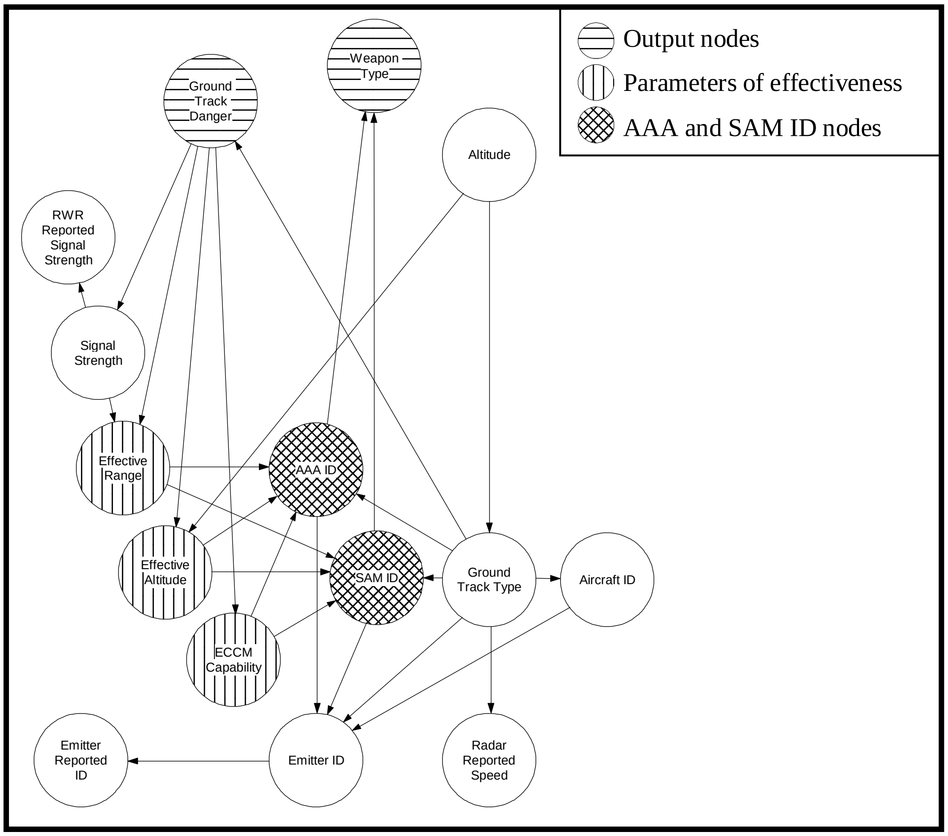


Figure 26 Assessing AAA and SAM Threats

ECCM Capability node has four states: high, medium, low, and none. All accounting for the AAA or SAM system's capability on electronic counter-counter measures, that is the capacity of denying electronic counter measures taken by its target. A higher ECCM capability means a more dangerous threat. Effective Altitude node relates each AAA or SAM system's vertical range (how high it can hit a target) with aircraft's current altitude; so an AAA with maximum vertical effectiveness of 5,000 ft will not be likely to harm an aircraft at FL150. Effective Range node uses the same

[12] As suggested before in page 109, lack of information from node Radar Reported Speed will increase the likelihood of a ground threat in contrast with an airborne threat.

rationale of Effective Altitude node but in the horizontal axis. However, since no information on distance is provided, Signal Strength node's data is considered in order to make a rough estimation of how far a threat is.

Portable SAM node has states “true” and “false”, and combines information from nodes MAW Report, Ground Track Type, Fly Over, Airspace Nature, and Signal Strength in order to assess whether there was a portable missile launching or not. A given threat is more likely to be a portable missile if the MAW reports a launch, the aircraft is flying over enemy land, and there is no signal from the RHAW.

Node Ground Track Danger is also connected to two other nodes: Emitter Mode (confusion matrix of RHAW Emitter Mode) and Ground Track Status. The first has four states regarding the mode in which the threat is emitting, an important clue to define whether there is an imminent danger or not. As noted earlier, an emitter in lock-on mode can usually fire a missile within 4 seconds, being a clear threat for the aircraft. The latter considers that it is more likely to find an enemy ground threat when flying over hostile territory, which explains its connection with node Airspace Nature.

As a result of this arrangement, all information that entered the BN will update the beliefs of nodes Ground Track Danger and Weapon time, which will be propagated to the DDN data fusion nodes. Figure 27 shows the ground track assessment process.

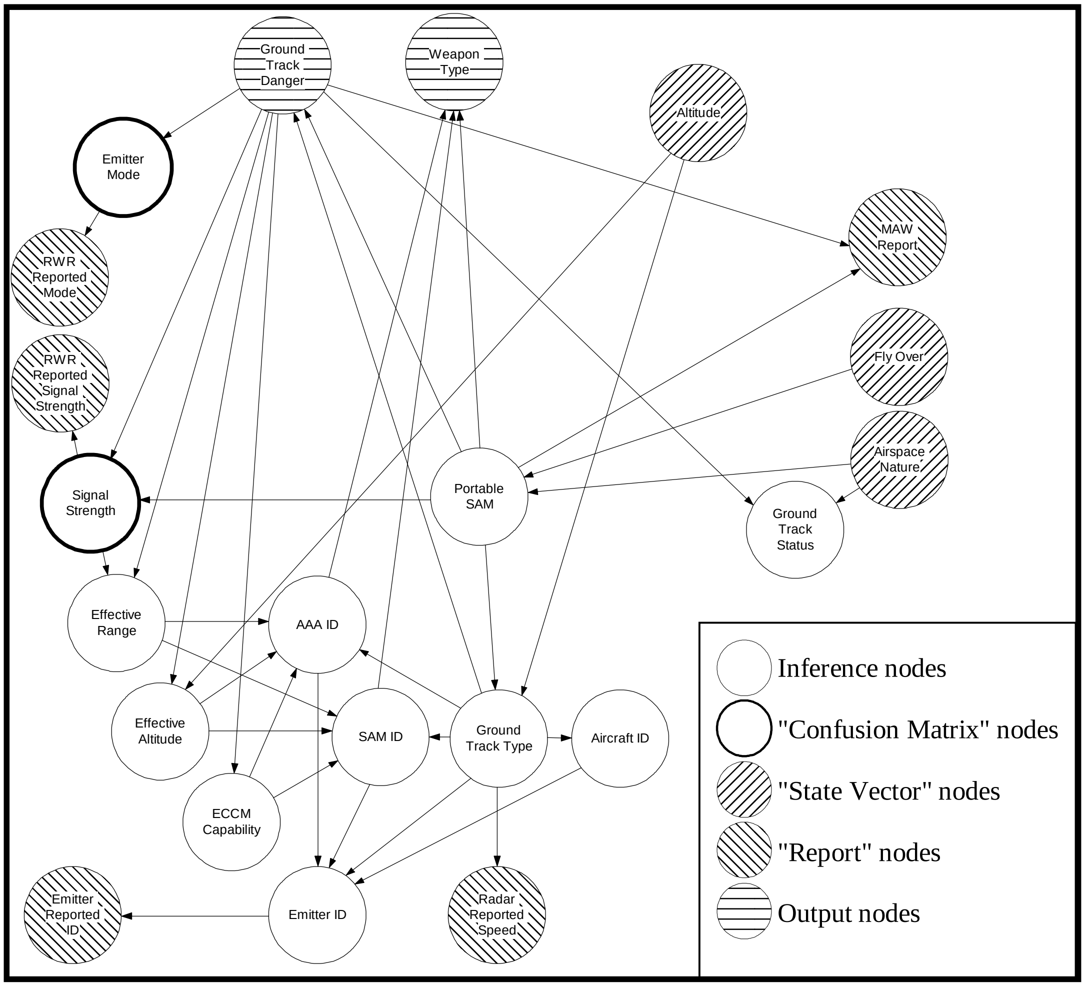


Figure 27 Ground Track Danger Assessment Scheme

4.2.3.2 Assessing an Aircraft Track

Unlike the ground track assessment, aircraft's danger does not have three distinct types. Instead, all aircraft types are included in node Aircraft ID. States for this node comprise a list of all aircraft that may produce any harm to the system, that is all airplanes that have the ability to launch air-to-air ordnance, “other aircraft” (all airplanes that do not fit into that list), and “not aircraft” (a ground type threat). In order to determine the identity of an aircraft, beliefs from nodes Aircraft Category, Aircraft Main Weapon, Formation, Aircraft Status, Emitter ID, and Ground Track Type are considered to infer Aircraft ID node's own belief. Then, that information is propagated to nodes Aircraft Danger and Weapon Type. This scheme is shown in Figure 28.

Aircraft Category node uses Radar reported speed data to classify an aircraft into three categories: supersonic, high subsonic, and subsonic. In a usual theater of operations, most (if not all) aircraft flying at supersonic speeds are interceptors, so a track flying at supersonic speed is very likely to be one of them. Tracks with high subsonic speed capacity includes all supersonic aircraft, plus most bombers, some unarmed aircraft, and even commercial airliners. The last state, “less than Mach .8”, is a common place for ground tracks (no information on speed) and all aircraft.

Aircraft Main Weapon node stores information about each aircraft's most capable weapon system (with respect of range), and its states are “BVR missile”, “IR missile”, and “none” (which includes gun-armed only and unarmed aircraft). As an example, “X” fighters carry both BVR and IR missiles (recall Table 1, page 38); since the BVR missile has greater range (27 nm x 15 nm), the system will consider it as “X” fighter's main weapon system.

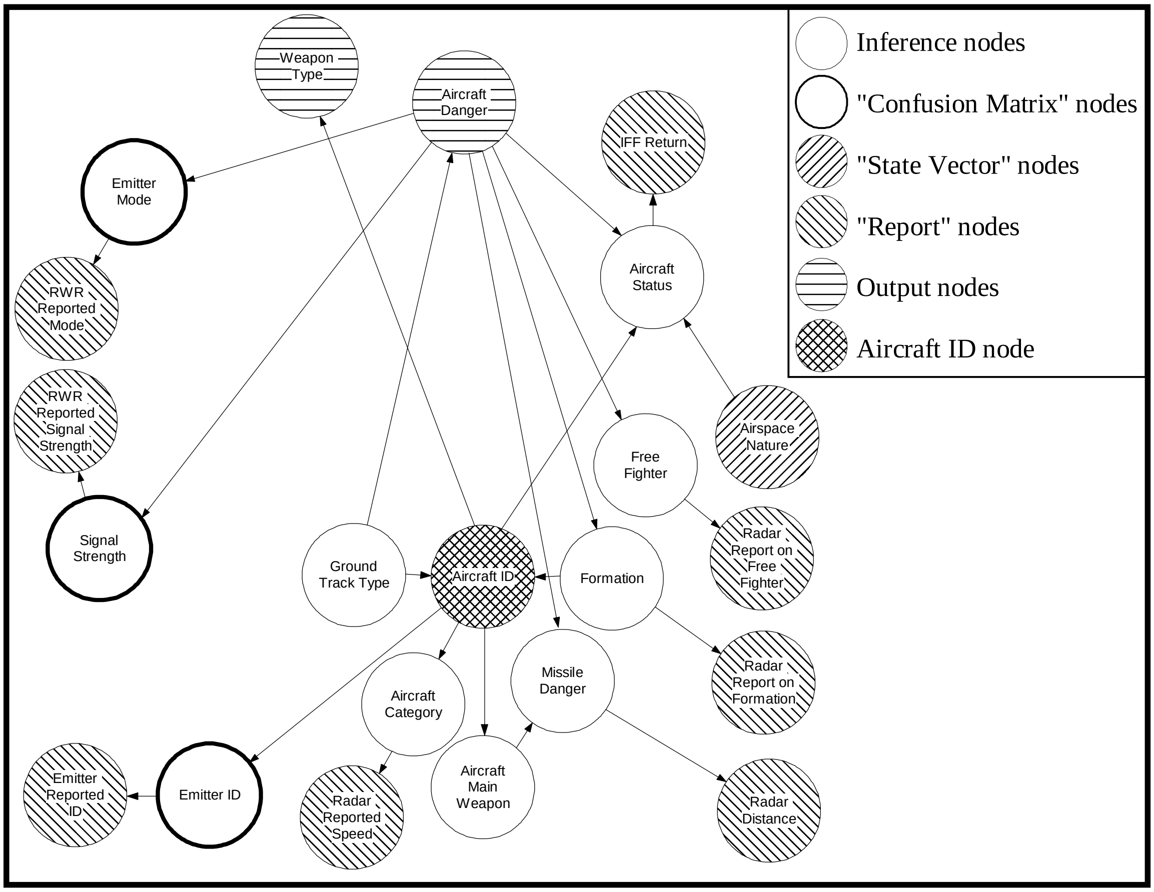


Figure 28 Aircraft Danger Assessment Scheme.

There is a close relationship among nodes Aircraft ID, Aircraft Main Weapon, Missile Danger, and Radar Distance. As stated before in this chapter (page 109), node Radar Distance is concerned with limits for the decision of launching a missile. This information, plus data in Aircraft Main Weapon node will be used to infer about the aircraft's exposition to an enemy missile (Missile Danger node), which will be propagated to node Aircraft Danger. However, information about an aircraft's main weapon system depends on its correct identification, which shows the interdependency among these four nodes.

Formation node, a confusion matrix from node Radar Report on Formation, is also related to Aircraft ID node. The rationale of this relationship lies in the fact that interceptors and attack aircraft almost always fly in formations of two, three, and four (or more) elements; depending upon mission's objective and current aircraft employment doctrine of the respective air force. Thus, knowing that two tracks are in formation flight increases the likelihood that both are interceptors or attack aircraft, so that information is then propagated to node Aircraft Danger.

Among the advantages of flying in formation, on of the most valuable is the use of air combat tactics as a means of subjugating the enemy. Considering the case of two interceptors against one intruder, in order to take advantage of numerical superiority the formation has to split, so the intruder will have to keep his attention focused in one interceptor while loosing contact with the other. The interceptor that stays keeping pressure on the intruder is usually called “engaged” fighter, while the other is denominated “free” fighter.

It is clear that a free fighter imposes a great danger to the intruder, even more then the set up by the engaged fighter, since it is common wisdom in the air combat domain that “the worse enemy is the one you can not see”. Free Fighter node, a confusion matrix of node Radar Report on Free Fighter, explores this characteristic of air combat techniques and is directly related to Aircraft Danger node. Between two equal tracks, if one has a greater belief of status “true” in node Free Fighter, it will have a higher danger level assessment.

Although Emitter ID is a good parameter to determine the identification of an aircraft, it will not declare whether this aircraft is friend or foe. Actually, this is a hard issue, since the same emitter can be found in both sides of a conflict. Aircraft Status node is intended to resolve this problem. To do so, it will gather information of two nodes: Airspace Nature and IFF Return. The first uses “prior” information (π) on the likelihood of finding an enemy aircraft when flying over friendly, enemy, or neutral air space. The former relies on the IFF transceiver as an “a posteriori” source of information (λ).

The IFF, when activated, sends a decoded inquiry pulse that is supposed to be acknowledged only by aircraft with the same cryptographic equipment, which is not the enemy's case. Though very useful for determining a track status, the IFF is an active equipment (it transmits pulses) so its use is not always recommended, mainly when stealthiness is a concern. The belief of node Aircraft Status, no matter whether or not updated by the IFF Report likelihood vector, is then propagated to Aircraft Danger node.

As a final remark with respect to Aircraft Danger assessment, nodes Emitter Mode and Signal Strength are used with the same rationale as in the ground track danger assessment. Figure 29 brings the complete BN as created with software NETICA™.

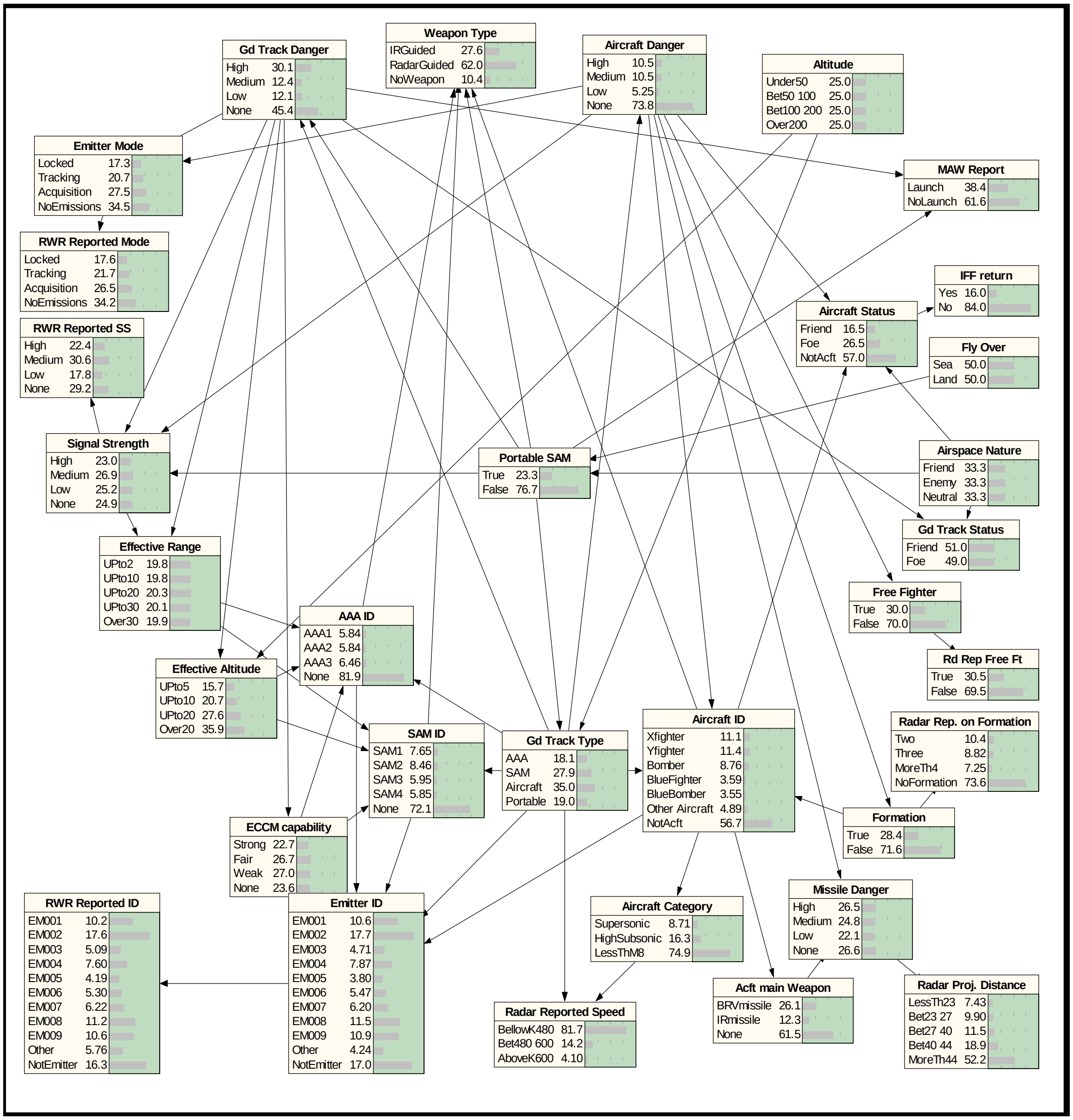


Figure 29 Individual Track's Bayesian Network

4.3 Analysis of the Solutions

Though it is based on the fairly established concepts of Bayesian Networks and Influence Diagrams, Dynamic Decision Networks is a new technique and lacks specific software packages for it. However, current software will fit in most DDN applications with little or no adaptations, depending upon the complexity of the application. Yet, dynamic characteristics of the method applied in this model (i.e., variable number of states and dependence amongst iterations) prevented the straightforward use of any package.

However, this issue did not characterize a major drawback, since it was possible to run the model using available commercial software (Netica™) with two minor adaptations, which can be easily incorporated during the implementation of the system's software code.

The first adaptation was the need of building distinct structures for each case scenario's snapshot, since the number of tracks changed between snapshots[13]. Different number of tracks implies in modifying the probability distribution matrix of data fusion nodes and changes in the number of status of Missile Action and Defense Action decision nodes. In addition, modifications in the number of status of the decision nodes cause the need of also modifying the probability distribution matrix of the nodes that are directly connected to them (i.e. nodes Attack Mission's Target, Avoid Fratricide, Detectability, and each track's Avoid Ground Attack and Avoid Aircraft nodes).

[13] With the exception of snapshots 2 and 3 of the first scenario, since they both have three tracks the same structure could be used in the analysis.

When implementing the DDN's data fusion sub-net, this adaptation can be made at least in two ways: actually creating one DDN structure for each given number of tracks or developing an algorithm that does it automatically. The first is easier to achieve, while in a real life system it would constraint the number of tracks because of available memory limitations. The second may be harder to implement, but would decrease the system's memory requirements considerably while maintaining its capacity. For the sake of simplicity, I opted to construct distinct structures for analyzing each scenario portrayed in Chapter Two.

The second adaptation is related to the performance needed for a real time application. Here, instead of compiling the network as a whole, the best approach is to do it by blocks, a procedure that can be made automatically during the implementation of the program's code. This will reduce drastically the demand for computational resources and will increase the performance to acceptable levels for real-time applications[14].

The first compilation was absorbing the node Altitude, since it has a known value (state) it will not create spurious linkages between nodes in its Markov blanket during the absorption process. The next step is to compile all track's Bayesian Networks, which will give as a result the networks shown in figures 35, 37, 39, 42, and 43 (all in Appendix C). Finally, the remaining influence diagram can be compiled in order to extract the optimum decision policy for that structure. Table 14 brings the results achieved by "Wise Pilot 2".

[14] In an non-optimized structure, using common software and a mid-range personal computer (Pentium II™ 266 MHz equipped with 128 MB of RAM) the average time for analyzing a 5-track structure was 4.7 seconds.

The results were pretty consistent with what we would expect for real situations and with the results achieved with the knowledge-based approach.

Table 14 Decision Policies Adopted by the "Wise Pilot 2" DDN System

Scenario / Snapshot	**Defense Action**	**Missile Action**	**Radar Action**	**IFF Action**
1^{st} / 1^{st}	None	None	On Low	Off
1^{st} / 2^{nd}	Deviate1	None	On Low	On
1^{st} / 3^{rd}	Dev1&ECM	Launch2	On High	On
2^{nd} / 1^{st}	Deviate4	None	Off	Off
2^{nd} / 2^{nd}	Break5	None	Off	Off

Even though they provided good criteria for assuring the model's consistency, performance, and capabilities, the results itself were only secondary parameters in the light of this thesis' purpose. Consistency can be achieved by calibrating the probability distribution functions (or even reassessing the node's relationship), performance comparisons would require much more than just two random situations and is not in the scope of this work, and capability can be improved by adding new nodes or states to the structure.

Yet, as stated in Chapter 2, the main objective is the conceptual comparison among approaches, which requires a more profound analysis on the primary characteristics offered by each approach during the development of both "Wise Pilot" systems. From this point of view, the following paragraphs cover some interesting remarks on the model's development under the Dynamic Decision Networks approach.

- *The processes of analysis and probability distribution assignments add a great amount of insight about the problem.*

Henrion (Henrion, 1991) has already pointed out that the most important product of the analysis that precedes the influence diagram development is the improvement in the problem's insight of the decision-makers. In his research, Henrion focuses on the fact that the decision-maker's need of recommending one decision over another would force him or her to get closely involved with the analysis process, which has the direct consequence of improving his/her understanding on the problem's nuances.

This perspective may lead to the conclusion that both rule-based and Bayesian expert systems would provide this benefit, since they both require a close participation of the decision-maker in the analysis process. I would like to make a distinction here, based on the differences of the analysis process itself. In the rule-based approach, the decision-maker will face the problem of deciding among a number of alternatives for solving a given part of the system. Usually, he tends to grade these alternatives in accordance with his expertise on the subject and then chooses the one he graded as the best action for that particular problem. It is fair to call this a qualitative assessment, and the act of grading the alternatives does add the decision-maker's insight on that particular element of the system.

In Bayesian case, the need of assigning probability distribution functions to each node requires the decision-maker not only to grade among alternatives for solving a particular decision element, but to perform a detailed assessment on the relevance of each alternative for the structure as a whole. The need of a comprehensive evaluation in the BN's case has its roots on the way the nodes' interrelationship is established.

Unlike the modularity among rules in a rule-based approach, Bayesian Network's nodes have the property called intercausal inference (Henrion, 1991) or induced dependencies (Pearl, 1988), where totally unrelated propositions can become relevant to each other as a result of new facts. One example of it can be seen in Figure 30, in which two variables that have a causal relationship with a third variable become conditionally dependent on each other.

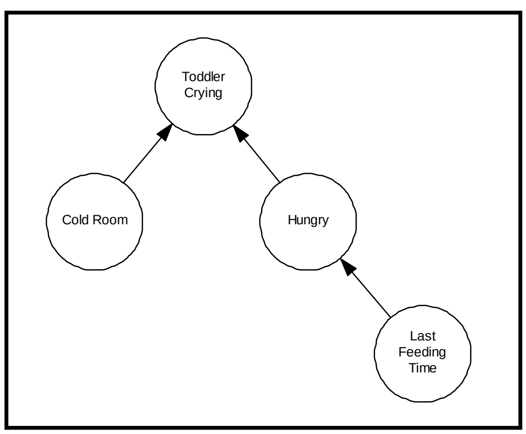


Figure 30 Conditional Dependency

By the moment I am writing this paragraph, I could perceive through the electronic nurse that my fourteen-month daughter is crying in her sleep. Among the many possible causes of her complain I graded two as being the most likely: she is uncomfortable with the room temperature and/or she is hungry. The hypotheses are neither mutually exclusive nor exhaustive, since both can occur concurrently and in conjunction with other hypotheses.

However, when I remember that her last feeding time happened just two hours ago and that her “endurance” in nighttime is usually between three to four hours, my belief on the “hungriness” hypothesis decreases. As a consequence, my belief on a too cold room increases, which makes the two hypotheses conditionally dependent, even though there is no direct connection between them.

As I sad in section 3.5 (see page 87), the rule-based scheme does not provide an automatic way of showing this special case of interrelationship among variables, and special rules had to be created for handling it, making the problem of excess of rules even worse. Graphical systems, like the DDN approach, not only provide a natural support for treating all kinds of variable’s interrelationship but also offer a straightforward representation for it.

As a consequence of this complete treatment of variables’ interrelationship, the decision-maker will have not only to assign probability distributions for each node but also assess whether (and why) it's assignments were correct or not, given the results of the overall structure. This is both a qualitative and quantitative analysis, which will provide the decision-maker with an even greater improvement in his knowledge on the problem than a rule-based system's analysis would achieve.

In addition, the probability distribution assignment will also force the decision-maker to gather more detailed information for each node, resulting in an even finer decomposition of the problem than the required by the rule-based system. I particularly felt this difference when I found myself looking for data to the “Wise Pilot 2” system that I didn’t even notice during the development of “Wise Pilot 1”.

- *Final diagram acts as a communication tool*

This characteristic is a consequence of the graphical orientation of the DDN technique. During the development of both “Wise Pilot” systems, every time I had to ask an expert about his opinion on a given subject it was first necessary to give him a broad view on the system. In the rule-based system’s case, this was not an easy task, since showing him a list of rules did not make my explanatory task easier. It was clear to me that most domain experts did feel more confortable when learning how to assess the variables (nodes) interrelationships in the graphical model than in the rule-based system. This characteristic can be extended to the non-domain-experts personnel involved in the system’s development, which is a desirable feature from the systems engineering point of view.

In addition, the easiness in perceiving the graphical system's logic and variables interrelationship is not only useful in the development process but also plays a valuable role in the system's maintenance and calibration. In case of unwanted results or inconsistent behavior, a graphical scheme provides the system's stakeholders with a powerful debugging tool, a point that can not be underestimated in treating with complex dynamic systems like "Wise Pilot".

- *The system covers all situations, not only the "preestablished" ones.*

This is another qualitative issue. Ultimately, rule-based systems analyze a given situation through a chain of "IF…Then" clauses until reaching the optimal solution. Unlike the number of possible situations in the real world, which can be considered for practical purposes as infinite, the number of "IF…Then" clauses is finite. Thus, it is fair to say that not all situations will find a chain of "IF…Then" clauses that perfectly fits to the situation's characteristics.

Instead, what actually happens is an approximation of the real situation's characteristics for achieving the "closest match" with a preestablished chain of "IF…Then" clauses. How closely will a given situation be with the available "IF…Then" chains of clauses is a function of the number of rules, a known constraint of the system.

A DDN, being a model of the reality, will also approximate real life situations to available states of knowledge modeled by the system. However, the knowledge modeled in its nodes is not a set of predefined causation rules (like the “IF…Then” clauses), but a complex set of probability distribution functions among many variables. The number of possible states of this structure grows exponentially with the number of nodes, edges, and variable’s states, in a way that a relatively small DDN is able to assume a number of states that may be considered as infinite for most practical purposes.

A philosophical explanation for this difference can be found in the fact that rule-based systems’ “IF…Then” clauses carry the experts’ knowledge about the domain variables, their interrelationships, and the inferences about many probable combinations among them (which is intrinsically stored in the clauses). DDN’s probabilistic distribution functions, on the other hand, carry the experts’ knowledge about the domain variables and their interrelationships, while the system’s algorithms make the inferences about possible combinations during execution time.

As a result of this philosophical differentiation, rule-based systems would require a huge amount of memory (in the form of rules) for storing the same knowledge of a relatively small DDN, the trade-off being that rule-based systems tend to outperform DDNs in terms of execution time. However, current computer technology in both hardware and software domains already shrunk this theoretical performance gap between the two approaches to almost imperceptible levels.

- *System values can be explicitly adjusted according to the situation.*

I regard this characteristic as one of the most powerful arguments for using DDN in this application's domain and in others as well. This opinion is based on the fact that changes in the value structure can be easily incorporated to the system, which makes it suitable for politically and/or doctrinally different scenarios. In other words, the system is able to reflect the changes in the way airborne operations are conducted.

As an illustration regarding "Wise Pilot 2" value structure (refer to Figure 17, page 94), suppose that the Blue Air Force is operating in a given theater of operations where avoiding fratricide is a major concern to the Blue commanders. Then, the main objective value node's parameters can be changed in order to reflect this concern, and a sensitivity analysis can be made before the actual implementation in the aircraft fleet. Examples of how a change in the value structure would affect the system's output are shown further in this work (refer to section 5.3).

Modifying the value structure is a feature that only the DDN system can provide in a feasible way, since the values for each objective are explicitly stated as parameters for the main objective node. In the rule-based system, where the value structure is implicit in the system's rules, any change would require modifications in most if not all of the rules.

- *Sensitivity analysis on the value structure is straightforward.*

The ability of doing sensitivity analysis is a major advantage by itself. In a DDN, it is only necessary to change the values of a node and run the program in order to see the results, a fast process that can be easily repeated as much as necessary for establishing the desired policy. In rule-based systems, where the rules already convey the inferences, the value structure actually is intrinsically spread through these rules. Changing the policy under which the rules were made implies in modifying most if not all of them, an time consuming and complex process that makes performing sensitivity analysis on the value policy at least infeasible, if not impossible.

As a conclusion for the application of DDN in the proposed scenarios, I would say that resulting system also provided doctrinally correct answers in both cases and in real time. In addition, during the development of the system, I could perceive many advantages over the previously employed approach that resulted in a more reliable, probabilistic consistent system.

These characteristics are more than enough to show that the DDN approach proved to be the most suitable for building a system that can successfully substitute the traditional human-based decision scheme in this domain.

CHAPTER 5

COMPARING THE SYSTEMS' RESPONSES

This comparison focuses on the qualitative aspects that arose during the development process in both approaches. As I stated earlier, a complete system would demand a significant development effort using decision analytic and systems engineering techniques, the result of this research activity provide what one would call the "flavor" of real systems.

However, some interesting conclusions can be drawn from these sample systems. In an initial analysis, section 5.1 explores the results of each system with respect to the case scenarios' portrayed situations. Then, section 5.2 extends this analysis to different situations, in an attempt of finding cases in which the two systems had diverse solutions. Once again, the main purpose of this brief sensitivity analysis is to make a qualitative assessment on the differences between the two approaches; while quantitative aspects are only a secondary issue. Finally, section 5.3 uses some of the hypotheses of section 5.2's sensitivity analysis in order to illustrate the DDN system's adaptability for different political scenarios or doctrines of operation.

5.1 Reassessing the Results

As we could perceive from data in the appendices, both "Wise pilot" systems achieved almost the same results for the first scenario and identical results for the second scenario, as we can see through Table 15.

Table 15 Decision Policies Comparison

Scenario / Snapshot	**Defense Action**		**Missile Action**	**Radar Action**	**IFF Action**
	Rule-based System	**DDN System**			
1st / 1st	No action		No action	On Low	Off
1st / 2nd	None	Deviate from track A	No action	On Low	On
1st / 3rd	Deviate and employ ECM against aircraft "A"		Launch missile at aircraft "C"	On High	On
2nd / 1st	Deviate from track "D"		None	Off	Off
2nd / 2nd	Break against "portable"		None	Off	Off

The difference in the results of the second snapshot of the first scenario can be accounted for as a consequence of the discrete nature of the rule-based system's outcomes. In Appendix B (page 194), I mention this characteristic in order to explain how I had to use the CF of each threat's hypothetical THREATLV parameter in order to establish a precedence among them.

Here, what happened was that tracks "A" and "C" of Figure 10 (page 50), albeit increasingly threatening, did not reach the rule-based system's threshold that would force the system to react. In the DDN system, the augmenting danger is sensed by the decision sub-net through the changing probabilities of the Aircraft Danger node. The result of this probabilistic assessment is an almost continuous output, which avoids drastic variations like the rule-based system's THREATLV jump from "3" to "4". The smoother computation of these parameters allows the DDN system to have more gradually spaced thresholds that are established by explicit statements about both uncertainty and value. As an overall result, the DDN approach will have less radical gaps between subsequent actions.

In this particular example, the DDN system was faster in perceiving the increasing danger provided by two tracks. For the rule-based system to achieve the same reaction time it would be necessary either to push the threshold farther, which might wind up in a too sensitive system, or to create new action rules for predicting this specific situation. However, adding new rules will certainly decrease the overall system's response time and memory requirements, while being not effective against similar situations that have slightly different parameters that would trigger different rules. In general, a well programmed DDN system will present a smother set of responses for the environment changes, while achieving better average reaction times than rule-based systems.

In spite of the specific difference in the systems' reactions I just commented, still most responses to the scenarios where pretty similar. There are two factors that could fairly explain why this was not a coincidence. The first relates to the common source for the expert assessments for both systems. In the developing phase of a system, a comparison between the initial results and the experts' judgements is a good system engineering technique for assuring a calibration of that system. The second factor is the relative predictability of the scenarios, where the actions were fairly acceptable by most experts in the domain.

Combining these two factors, one can perceive that the actions the experts were willing to take in the situations portrayed by the scenarios actually provided a good "calibration standard" for both systems, which explain the similarities of their outcomes.

However, even tough the answers for the scenarios were similar, we still could observe several qualitative differences between the systems during the development process of each. These distinct characteristics are stated in the last sections of Chapter 3 and Chapter 4, where the pros and cons of each system were commented. Besides, in spite of the already cited factors that contributed for the identical answers, it is fairly intuitive that the conceptual differences between the approaches would result in distinct outcomes in some situations. The next section is intended to present and analyze a number of these cases.

5.2 "What – If" Analysis

In this phase of the thesis work, many different situations were created and tested in both systems. In general, most of these situations received similar or at least consistent solutions by the two systems, mainly due to the factors commented upon in the previous section. However, the qualitative differences between the approaches also provided this sensitivity analysis phase with divergent outcomes in some situations.

It is not my intention to present an exhaustive report on all of these particular situations or on the phase as a whole, for it would not add much insight to the understanding of each system's strengths and limitations. Instead, my intention is to perform an analysis in a few situations that would permit inferences on those behavioral aspects and its influence on the overall performance of each approach in producing an effective, reliable system.

In order to avoid the unnecessary setup procedures of new scenarios, which might have forced new assumptions and definitions, I opted to present a few insightful situations that were consisted of modifications of the scenarios already established in Chapter 2.

The first of these situations is an adaptation of the first scenario's second snapshot (refer to Figure 10 in page 50). Four hypotheses were made from that depicted situation, all of them were related to track B, which is assumed to be 37 nautical miles from the Blue fighter, instead of the original scenario's 41 nm.

Hypotheses 1 (H1) and 2 (H2) assume that since the track has already passed through the 40 nm limit, both systems had already sent IFF inquiries and track B failed to answer these inquiries, which increases the likelihood of an enemy track. Hypothesis 3 (H3) and 4 (H4) do not take that assumption, so no IFF information is considered previously by both systems. Figure 31 depicts the new situation.

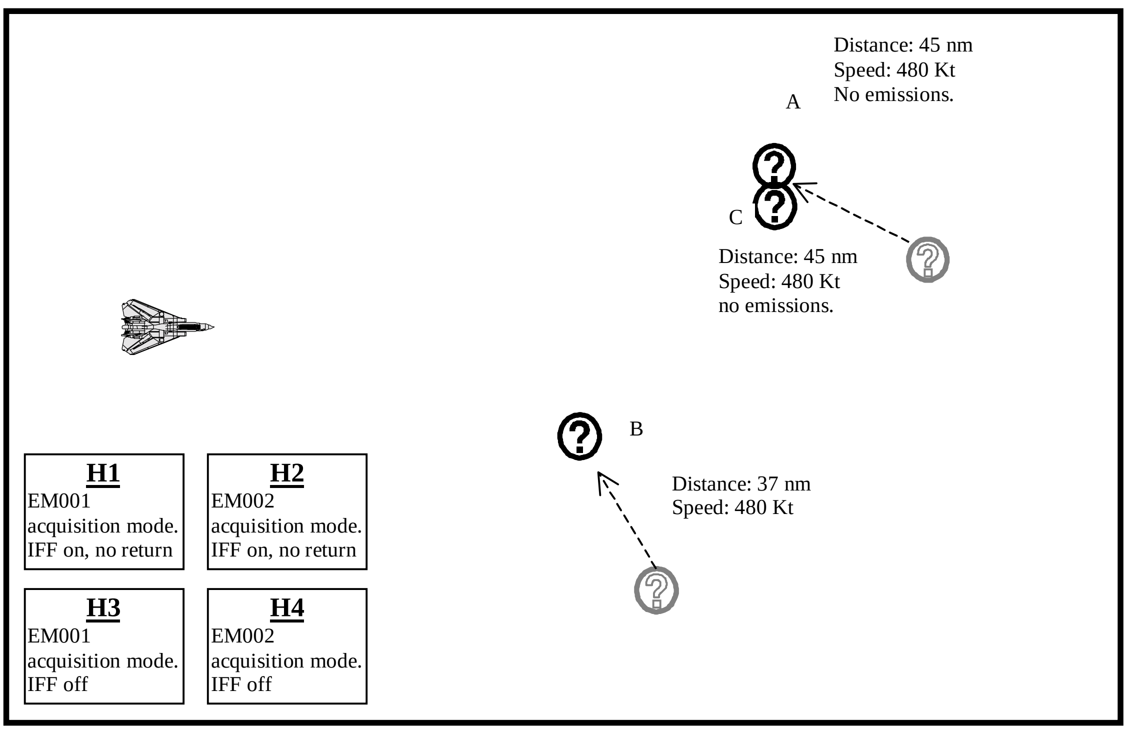


Figure 31 Situation for Hypotheses 1 through 4.

In addition, Hypotheses 1 (H1) and 3 (H3) assume that the track is emitting a medium signal in acquisition mode, and this signal is cataloged by the Blue Force's EW

Officer as being from an emitter EM001. In contrast, Hypotheses 2 (H2) and 4 (H4) assume the same signal strength and mode, but as coming from an emitter EM002.

Following “Wise Pilot 1” system’s rules (refer to Appendix A), the backward chaining process will go through a similar path of rules until arriving at Rule 128, which will define the THREATLV parameter for track B as “4” in both H2 and H4. That outcome makes Rule 618 dictate the rule-based system’s response for both hypotheses, which is to turn the radar in high mode and launch a missile to track B. In the case of both hypotheses 1 and 3, the system will use Rule 137 for defining a THREATLV parameter of “3”, and Rule 623 for making the system to turn the IFF on and send an inquiry.

“Wise Pilot 2”, when fed with the data in the hypothesis tended to stress the effectiveness of X fighters as interceptors when confronted with Y fighters. Although the same concern is implicit in the rules of “Wise Pilot 1”, the fact that track B’s distance below the 40 nm limit apparently dictated the actions. Table 16 brings the outcomes of all decisions but the IFF action, which is to turn the IFF on for all hypotheses in both systems.

Table 16 Comparison of the Decisions for Hypotheses 1 through 4.

	Defense Action		Missile Action		Radar Action	
	Rule-based System	**DDN System**	**Rule-based System**	**DDN System**	**Rule-based System**	**DDN System**
H1	None	Deviate from track A	Launch at track B		On High	
H2	None	Deviate from track A	Launch at track B	Engage at track B	On High	
H3	None	Deviate from track A	None	Launch at track B	On Low	On High
H4	None	Deviate from track A	None	Engage at track B	On Low	On High

In a general sense, the rule-based system made no intermediate action; it either opted to take the toughest action (launch a missile) or to do nothing. The tough actions were taken when there was no return to a sent IFF inquiry (H1 and H2) without any regard to the type of track, while no action was taken when in the absence of the IFF information (H3 and H4).

I consider this behavior not desirable, since in all hypothesis we are dealing mostly with the same parameters (i.e. same distance, emission mode, signal strength, etc), and there already is enough data to evaluate the amount of danger implicit in each hypothesis. That assessment would allow the system to take optimized actions, instead of taking radically different attitudes for relatively similar situations.

It is important to observe that this behavior was not a consequence of the absence of an intermediate action rule in the list of rules of “Wise Pilot 1”. Actually, there is at least one rule (i.e. Rule 621) that might have been applied to this situation. However, the sudden change of the THREATLV parameter from “2” to “4”, a consequence of the system’s 40 nm threshold, prevented the application of that intermediate action. This example is an interesting illustration on how the deficiencies of the rule-based approach translated in a system that takes undesirable actions, even when this system has rules that were made in order to avoid it.

In both H3 and H4, we do have a track that is very likely to provide some amount of danger to the Blue fighter, albeit still in acquisition mode this track may suddenly become a serious threat. Instead of just sending an IFF inquiry, most experts would rather have taken some preventive actions, like deviate from it or threaten it with a missile engagement. In H1 and H2, where track B is a known enemy, most fighter pilots would agree with the system’s solution for H1 because of the higher peril brought by an X fighter, while some would still consider just engaging the missile instead of launching it. However, in H2 the majority of experts would prefer to just put pressure on the Y fighter pilot by engaging a missile, so they might save a valuable, scarce asset (the missile) while still avoiding the lesser threat just before having to deal with the two unknown upcoming tracks A and C.

In contrast to the “do all or do nothing” behavior of the rule-based system, the DDN system took more consistent actions, regarding the difference between the performance gap between X and Y fighters, and relying more on the already collected information rather than being tight to the IFF. Indeed, the actions were pretty similar to the “most preferred” ones of the previous paragraph, with the exception of the H3’s option of launching the missile. In that situation, most fighter pilots would feel that it is preferable to just put pressure (by engaging a missile) on a track still emitting in acquisition mode, while waiting for the answer (or lack of answer) to the IFF inquiry. Using this solution they would be avoiding a fratricide fire, since the lack of IFF information implies in a still great amount of uncertainty on track B’s nature, while saving the missile for the two upcoming tracks.

As I said before, I regard the rule-based system’s unwanted behavior in this specific situation as a consequence of two qualitative issues. First, the discrete nature in which the knowledge is applied to the situations usually is responsible for radical differences between the solutions for situations near threshold parameters. Secondly, the primitive way in which uncertainty is treated in the base-ruled approach, a consequence of the lack of a consistent support for the probability theory, prevented the system to make an assessment that was not so dependent on the IFF information.

Nevertheless, both systems provide means for adjusting the above-cited shortcomings in its solutions. In the DDN system, one can merely change some node’s probability distribution functions in a way to achieve the desired behavior from the system. In the rule-based system, as a consequence of the qualitative limitations listed in Chapter 3, the same simplicity does not hold. In order to obtain the desired solutions for the hypotheses, it is necessary to add at least an equal number of new inference rules that allow the system to correctly assess the danger in each particular situation and new action rules that cover those new assessments. In addition, a throughout revision of the system's rules is necessary to prevent conflict between the new rules and the old ones.

As a final comment on these four hypotheses, it is important to note that adding and revising rules for that situation would not cover many other similar situations, for they will be focused in the parameters of those particular hypotheses. Actually, correcting undesirable outcomes always will imply in the addition of new rules (and redo all the consistency analysis), which leads to the trade-off between the number of rules versus the desired performance. This is an undesirable consequence of the static nature of the rule-based approach, and a convincing argument for the appropriateness of the DDN technique in that category of systems.

Another illustration of these characteristics can be seen in Figure 32, which portrays four additional hypotheses regarding the situation recently analyzed. Here, track "B" is still at 37 nm, and its emissions are assumed of being from an EM001. However, in order to increase the number of relevant threats and make it harder to sort them with respect to their "danger" level, tracks "A" and "B" are also brought to a closer distance. Also, since track "B" was already detected below the threshold for an IFF inquiry, it is assumed that all aircraft had failed to provide a reply to that inquiry, in other words, all aircraft have an extremely high probability of being enemy assets.

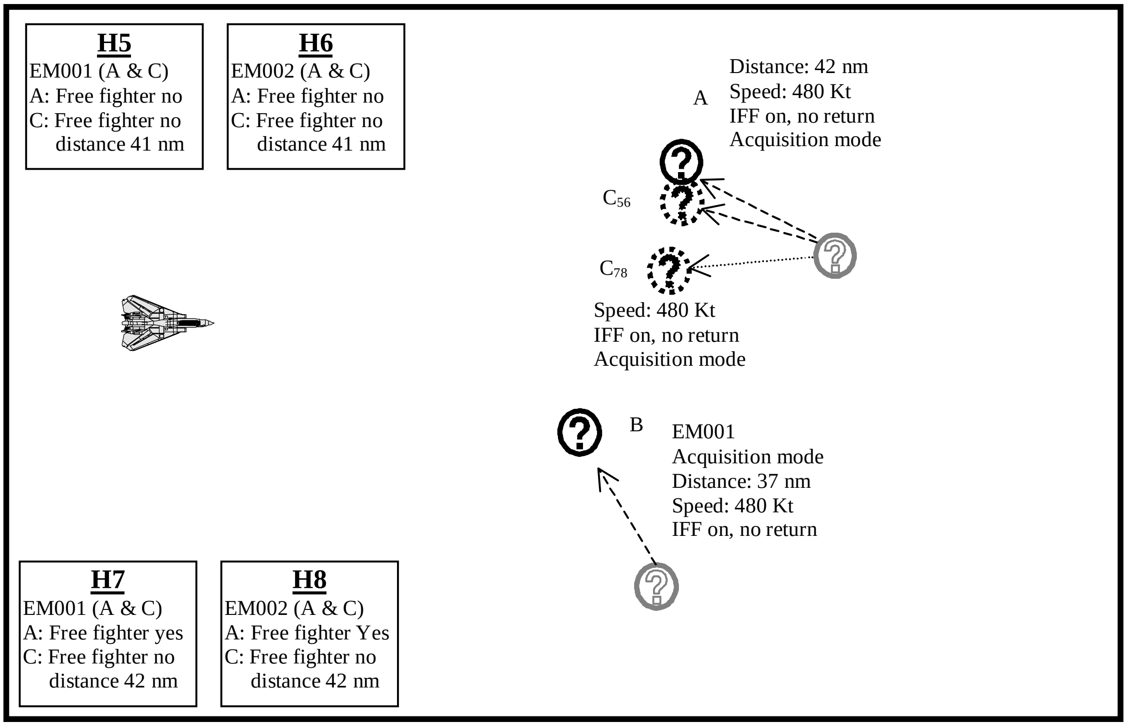


Figure 32 Situation for Hypotheses 5 through 8.

In hypotheses 5 (H5) and 6 (H6), tracks "A" and "C" are at 42 nm and still flying in close formation, while in hypotheses 7 (H7) and 8 (H8) they have split the formation and are flying at 42 and 41 nm respectively. Also, in H5 and H7, it is assumed that they are emitting an EM001's signal with medium strength, while in H6 and H8 the emitted signal comes from an EM002, also with a medium strength.

As we can intuitively perceive, this situation is quite undefined. All aircraft are considered as enemies and all provide a considerable amount of danger, but sorting them with respect to their respective "danger" level is far from being a straightforward task. Although an X fighter at 37 nm did provide a fair amount of danger to the Blue fighter, it is not clear whether this danger is lesser than the one provided by an opened formation of Y fighters (or by one of X fighters) just four miles farther.

Interestingly, this time the resulting backward chaining process will go through a path of rules that arrives at Rule 138, which will define the THREATLV parameter for track B as “2” for all hypotheses to both tracks “A” and “C”. A similar procedure will use Rule 131 for defining a THREATLV parameter of “4” to track “B”. This outcome makes Rule 618 dictate the rule-based system’s response for all of the present hypotheses, which is to turn the radar in high mode and launch a missile to track B.

In the DDN system, the outcomes reflected a balance among the various factors influencing the amount of danger to the Blue fighter. In both hypotheses with closed formations (H5 and H6), the system considered the isolated X fighter as the main threat. In contrast, the hypotheses in which a free fighter was present (H7 and H8), this track was considered as the main threat, while the isolated X fighter (and not the formation’s engaged fighter) was considered as the second threat. This last detail is a good illustration of the system’s ability to deal with multiple threats, where an optimal action was taken under the light of the situation as a whole, instead of focusing on the hardest threat only.

Table 17 brings the outcomes of all decisions but the IFF action, which is to keep the IFF on for all hypotheses in both systems.

Table 17 Comparison of the Decisions for Hypotheses 5 through 8.

	Defense Action		**Missile Action**		**Radar Action**	
	Rule-based System	**DDN System**	**Rule-based System**	**DDN System**	**Rule-based System**	**DDN System**
H5	None	Deviate from track B	Launch at track B	Launch at track A	On High	
H6	None	Deviate from track A & use ECM	Launch at track B		On High	
H7	None	Deviate from track B & use ECM	Launch at track B	Launch at track A	On High	
H8	None	Deviate from track B & use ECM	Launch at track B	Engage at track A	On High	

This time, the discrepancy between the systems is more evident. In the rule-based system, the formation did not reach the discrete "threshold" of 40 nm so it received a THREATLV parameter of 2, even though it is considered an enemy formation of X fighters (in H5 and H7) or Y fighters (in H6 and H8). Instead, the system focused on the isolated X fighter, which also provides a great amount of danger, and took the necessary actions for denying it.

In the DDN approach, the isolated X fighter was considered as having more potential danger than the closed formation of Y fighters in H6. However, the closed formation of X fighters of H5 and both open formations of H7 and H8 were considered as the main adversaries for the Blue fighter. Nevertheless, in all hypotheses, the DDN system has distributed the actions through the formation and the isolated fighter, so both are being denied by the system. Looking at a different perspective, by counteracting the danger imposed by each of the two distinct threats, instead of focusing only in the formation or on the isolated fighter, the system made an optimal use of its available resources for self-defense.

It is interesting to observe that the DDN system's Radar Distance node also has a discrete list of states that is based on ranges of distances, which might lead to the wrong conclusion that the same "discrete threshold" phenomenon would also occur. However, the DDN inference engine works does not treat a single node's value by the same way as the rule-based system reasons with its heuristics. Instead, as explained in Chapter 4, the probabilistic orientation of the DDN approach uses the knowledge implicit on the probability distribution functions interconnecting the various nodes as the basis for its inference engine. As illustrated by this example, although its individual nodes may contain discrete "thresholds", the DDN system provided a continuous-like danger assessment where a given track will not "jump" from a threat assignment of "2" to "4" just because it passed through a threshold.

Of course, the same corrections listed for readjusting the rule-based system in hypotheses 1 through 4 can be applied to hypotheses 5 to 8. However, as in the H1/H4 case, corrections based on adding new rules come at a price that may not prove to be reasonable, mainly if we consider that this price will be paid for solving a small set of situations only, adding few behavioral improvements to the system as a whole. Yet, there is also the possibility of modifying the threshold, which only changes the range in which the problem will occur, also adding no qualitative improvements to the system as a whole.

Another issue to be (re)emphasized is related to the system's ability of dealing with separated threats that have similar danger assignments. Figure 33 brings four hypotheses that are similar to those considered in Figure 32, but this time the formation is at 38 nm (H9 and H10) and at 37/38 nm (H10 and H11). This new distance ranges were intended to "break" the 40 nm threshold of the rule-based system, which will intuitively result in a greater THREATLV parameter for the formation.

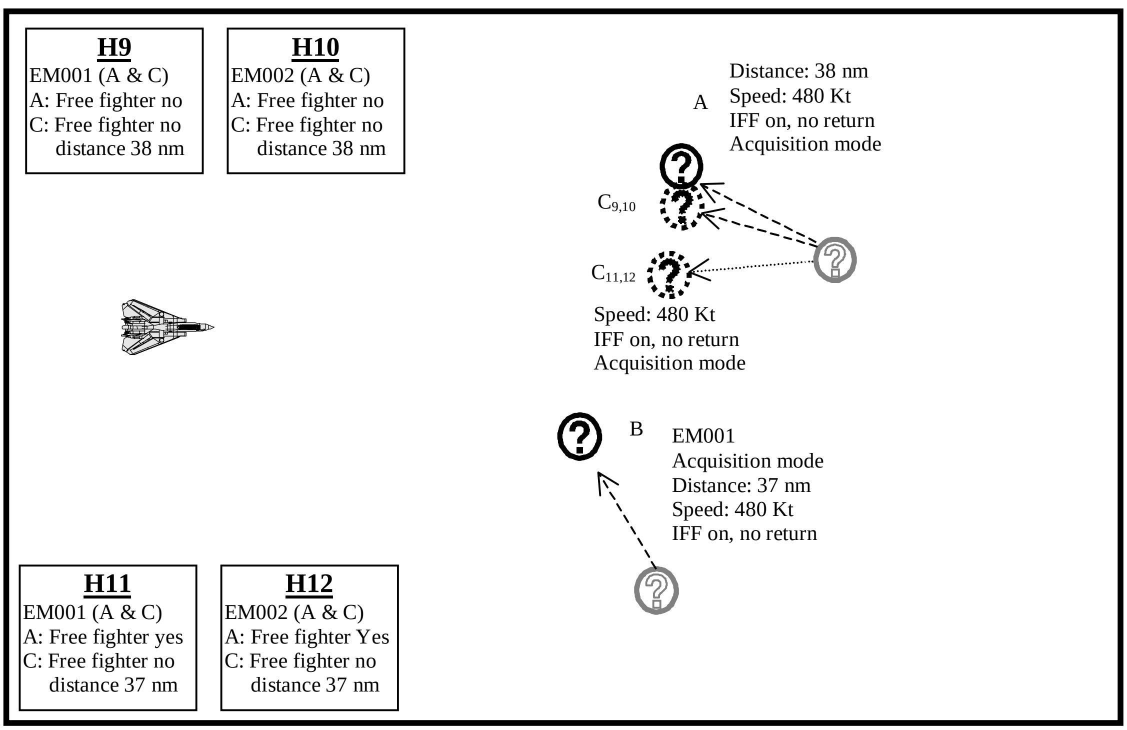


Figure 33 Situation for Hypotheses 9 through 12.

As expected, the rule-based system will use the very same path that includes Rule 131 for defining a THREATLV parameter of "4" to track "B". However, this time the formation will also receive a THREATLV of "4", in H9 and H10 this assignment comes from Rule 130, while in H11 and H12 it comes from Rule 129. In all hypotheses from H9 through H12 the system will use Rule 619 to launch a missile against the free fighter of the formation while employing ECM techniques against the other tracks inside the formation. Table 18 brings the results achieved by both systems.

Table 18 Comparison of the Decisions for Hypotheses 9 through 12.

	Defense Action		**Missile Action**		**Radar Action**	
	Rule-based System	**DDN System**	**Rule-based System**	**DDN System**	**Rule-based System**	**DDN System**
H9	Use ECM on track C	Deviate from track B & use ECM	Launch at track A		On High	
H10	Use ECM on track C	Deviate from track B & use ECM	Launch at track A		On High	
H11	Use ECM on track C	Deviate from track B & use ECM	Launch at track A		On High	
H12	Use ECM on track C	Deviate from track B & use ECM	Launch at track A		On High	

The DDN system, as expected, achieved results that were consistent to the ones from H5/H8. In all cases it considered the formation as carrying more danger than the isolated fighter, while also taking actions that distributed its resources through distinct threats, even considering that the formation's engaged X fighter in H11 theoretically has the same threat potential of the isolated X fighter. Indeed, both are X fighters with the same parameters, but the system rationale considered that the engaged X fighter is already receiving some pressure from the Blue fighter (a missile was launched against his wingman), while the isolated X fighter is still acting free.

Although the results in Table 18 are relatively consistent between the two systems, it is clear that the DDN system utilized its resources in an optimized way. Translating the DDN system's rationale in this specific situation into heuristics for the rule-based system not only would require more rules and adaptations for the old ones, but also would cover only this specific case.

As stated before, other hypotheses in different situations would be presented here to reinforce the problems encountered when using the rule-based approach in dynamic decision problems like the fighter's automated self-defense system. However, the main objective of this section was to give an idea on how the qualitative differences between the approaches that were stressed in Chapter 3 and Chapter 4 translated in divergent outcomes in some situations. A secondary objective was to show the obstacles imposed by the rule-based approach for achieving a system that would present the same behavior as the "Wise Pilot 2". Both objectives were fairly covered in the twelve hypotheses presented, and further examples would only reinforce the lessons learned in this section, since they would only present the same pitfalls of the rule-based approach through a different view.

5.3 One System for all Battles

Earlier in this work (refer to section 4.3), I regarded the ability of implementing changes in the value structure to the system as one of the most powerful features of that approach (see page 130). Also, I pointed out that it would be unfeasible to replicate that same feature in a rule-based system, since the value structure is embedded in its rules.

An implicit advantage of graphical probabilistic methods like Bayesian networks and DDNs is the possibility of changing any node and analyzing almost instantaneously the results of that change. In contrast, modifying rules is a complicated process (mostly because of the delicate balance of distributed knowledge among the rules) that would need a complete recompilation before any analysis could be made.

Nevertheless, modifying and analyzing nodes of the data fusion or the inference sub-net is a nice feature for the development phase. Once the system has the expert's knowledge correctly translated through the DDN's probability distribution functions there will be little use of modifying it. However, the real dimension of this capacity lies in the ability of changing the decision sub-net's nodes, which carry most of the information that is sensitive to political or doctrinal changes, even when the system is in actual use.

While the decision sub-net nodes contain all the information that guides the decision process, it is the main objective node that carries the policy to be followed by the system. In other words, each decision node's value dictates how important that respective parameter is to the system as a whole, and while the other nodes are related to secondary, situation-related parameters, the main objective node brings the decision-maker value structure. Thus, from all nodes of the decision sub-net, the natural candidate for any changes reflecting the doctrinal or operational decision-making policy is the main objective node.

In the previous section, a set of hypothesis was visited in order to emphasize the behavior of the two "Wise Pilot" systems under specific circumstances. However, that sensitivity analysis was made without any modification on the actual systems. The central purpose of this section is to show how changes in the parameters of the main objective node would affect the system's outcomes.

Once again, in order to avoid unnecessary setup procedures, I opted to analyze two already known situations that form a continuum in terms of potential danger to the Blue fighter: hypothesis H1 and H9. For hypothesis H1, refer to Figure 31 (page137), for hypothesis H9, refer to Figure 33 (page 147). The process of changing and analyzing the main objective node itself is fairly straightforward, and it is replicated here the same way it would happen in a real case.

The values in the main objective node, as explained in Chapter 4, convey the information on how important each objective of the decision-making value structure is to the process as a whole. This information is expressed as a percentage of each objective's importance, so it should be a number between 0 and 1. Although the software used in "Wise Pilot 2" (Netica™) uses a slightly different notation, I will present these values as a vector with three elements, each representing a system's objective. The elements will appear in the following order: Attack Ground Target, Maximize Survivability, and Avoid Fratricide. Also, in order to maintain consistence among the results, I have used the same value structure for all the work that has been done with respect to the "Wise Pilot 2" system.

In the structure used throughout this work, I assumed that the objective "Attack Ground Target" was the most important and assigned a value of .45; the second objective in the vector's order is also the second in importance, and a value of .30 was assigned to it. Therefore, the last objective, "Avoid Fratricide", will receive an importance value of .25, and the objective vector will be [.45, .30, .25]. Using this value vector, the "Wise Pilot 2" system decided to launch a missile against the most threatening track in both H1 and H9 hypothesis.

In this section's analysis, three extra runs of the program were made using the same parameters found in these hypotheses to the DDN system, but with different value vectors. The first value vector, [.33, .33, .33], was intended to verify how the system behaves when the objectives receive an equal importance assessment. The second vector, [.30, .30, .40], simulates an environment where avoiding a fratricide fire is a main concern in the operation. The last vector, [.25, .25, .50], is a dramatization of the previous one, and approximates a situation where avoiding a fratricide fire assumes a degree of importance that equals the sum of the remaining objectives. Of course, these three policies were set only for comparison purposes, and no attempt to resemble real situations was made. Table 19 shows the analysis' results.

Table 19 Results of Modifying the Main Objective Node.

<table>
<tr><th></th><th>[.45, .30, .25]*</th><th>[.33, .33, .33]</th><th>[.30, .30, .40]</th><th>[.25, .25, .50]</th></tr>
<tr><td>H1</td><td rowspan="2">Deviate from the secondary threat and launch a missile against the main threat</td><td>Same as the original</td><td>Deviate and employ ECM against main threat</td><td>Deviate and employ ECM against main threat</td></tr>
<tr><td>H9</td><td>Same as the original</td><td>Same as the original</td><td>Deviate and employ ECM against main threat</td></tr>
</table>

* Original vector

The outcomes reported in Table 19 follow a fairly predictable pattern. When increasing the value of "Avoid Fratricide", we are actually increasing the penalty for firing a missile against a friendly track. Due to this augmented cost, the expected value for all options that include firing a missile will decrease, and it is intuitive that system will be more conservative when deciding to take that kind of action.

When using the "equal weight" vector, both situations required the launch of a missile, which shows that even for a greater value of the "Avoid Fratricide" (.33 instead of .25) there was enough perceived danger in both hypothesis for justifying the risk of a fratricide fire. However, after a slight increase in the "Avoid Fratricide" value (.40 instead of .33), the system decided that the input parameters in H1 does not justify the decision for launching a missile, which was not true for the riskier H9, where the original decision was maintained. Finally, when the “Avoid Fratricide” received the same weight as the sum of the other two objectives, the system declined to launch a missile in both hypotheses.

As I stated before, the main goal of this analysis was not related to the pertinence of those decisions, once it can be modified through further adjustments in the decision sub-net. Instead, what must be carefully observed is the powerful capability brought by the DDN approach when providing a system that can be verified against an operational / doctrinal policy and modified accordingly.

CHAPTER 6

CONCLUSIONS

6.1 Reviewing the Major Ideas

This research was focused in presenting the Dynamic Decision Networks technique as the most suitable for complex, time evolving decision Problems. The fighter pilot's weapon and sensor allocation problem was chosen as a case study, since it contains inherent properties that prevented previous optimization approaches from providing a reliable automated decision system.

The basic concepts that support the DDN technique were presented in chapter 2, where we could find brief comments on Probability Theory, Multi-Attribute Utility Theory, Graph Theory; which provide the mathematical and philosophical background for the technique. Then, Bayesian networks and influence diagrams were introduced separately as a preparation for the DDN explanation, since this technique combines the first two graphical analytic methods.

A DDN was then defined as a flexible, probabilistically consistent, and mathematically powerful decision-making tool for real-time, dual control analysis of complex systems. The technique's structure can be divided in three independent parts, the decision sub-net, the data-fusion sub-net, and the inference sub-net. The first can have one or more influence diagrams, which are devoted to define the decision policy that provides the best expected value for the objective function. To perform this task, the influence diagrams will be fed by data coming from the second sub-net, which may consist of an influence diagram or one or more Bayesian networks devoted to consistently merge the data coming from the inference sub-net. The last sub-net receives external data from the system's external interfaces and uses a set of Bayesian networks as inference engines that update the uncertainty of the system.

Also, we learn that the DDN has not a static structure. Actually, the number of nodes in the data fusion sub-nets is function of the number of Bayesian networks in the inference sub-net, which in turn varies with the number of objects of interest to the system. This flexibility is one of the key concepts that make the DDN suitable for domains where requirements, objectives, and their respective expected values change over time.

The next step in chapter two was to present a comparative case study. The fighter pilot's weapon and sensor allocation problem was introduced in two different scenarios, both with threats that possessed time evolving uncertainty parameters. We could perceive that the pilot decisions had to be sensitive to time, as their outcomes also depend on it.

Chapter three began with an introduction of the rule-based knowledge systems approach. Then, the "Wise Pilot 1" automated decision system was introduced as a result from the application of that technique in the case study. Although a fairly functional system was constructed, we could see that the pitfalls of the approach prevented the reliable applicability of such a system. The main obstacle was the sloppiness in which the probabilities were treated by the system.

In chapter 4, the same scenarios were solved using the DDN approach. Although not a “ready-to-implement” system, “Wise Pilot 2” proved to be a suitable answer to the fighter pilot’s problem. Some positive characteristics were highlighted in the chapter’s conclusion, most emphasizing the flexibility of the system and other secondary, but also considerable, strengths of using a graphical approach.

Some personal observations were also commented during the development of both systems. The need for a deeper analysis in the DDN system was a nice surprise, where we were forced to think harder on the problem in order to answer the detailed questions posed by a probabilistic consistent system. I would stress that the increase in insight on the problem was not a lesser advantage, mainly in a domain were experts with different backgrounds are involved.

After observing the development of both systems, and their pros and cons as well, a sensitivity analysis was made in order to observe how the differences between the approaches influenced each system’s behavior. Chapter 5 brought some insightful examples on that sensitive analysis.

As a conclusion, the observed advantages of the DDN approach proved to be vital in designing a system that could handle the case study's challenge; and the applicability of the technique in similar real-time, complex problems is an intrinsic lesson to be computed.

6.2 Limitations of this Work

There are some points that I did not cover in this study, either because of its appropriateness to this thesis' scope or because of its excessive demand for time to its completion. Nevertheless, I would like to comment some of them in order to set the stage for future research.

One of the most important, yet relatively unexplored, topics in this field is related to the correct approach for managing the dual control problem, which bring us to the issue of how to control the value of information. For the sake of simplicity, "Wise Pilot 2" employed a single influence diagram in the decision sub-net for controlling both the decisions related with the aircraft's self defense and the decisions related with the data collection activities. The latter being based on the value of information brought by the sensors in contrast with the tactical cost of keeping these sensors on.

This is not a simple issue, since a better approach would be the adoption of separate influence diagrams for each type of decision. This would improve the response time and make the overall control more effective by establishing specialized structures for different problems and parameters.

Another possible advantage would be the consequent higher level of modularity in the system, where having separated structures translates in an easier debugging at development time and faster troubleshooting at operation time.

Yet, applying a unified structure has its advantages. The resulting structure has fewer variables, which means a decrease in both complexity and memory requirements of the system as a whole. In addition, a higher level of integration would increase the efficiency in the code, by decreasing the overall number of nodes a given input data has to pass through.

Nevertheless, albeit very important this issue would not be inserted in this work without a dramatic increase in the research time, and also it would be a “thesis under a thesis” given its complexity and relatively lack of literature on the subject. Thus, I opt to do not cover it, while still calling the attention for its importance.

Another point I would like to comment is the qualitative focus of this work. Instead of covering a large number of techniques, I adopted Pearl’s taxonomy (Pearl, 1988) for defining a comparative approach. I already justified this option in the beginning of this work (refer to section 1.2), but a prospective reader would inquire why I elected the DDN to represent the intensional group instead of using a third technique for increasing the comparison scope. As a reply in advance to such an inquiry, I refer to the metaphor made by the taxonomy's author.

According to Pearl's vision, handling with uncertainties require a trade off between our desire to use the computational permissiveness of extensional systems and our ability from committing semantic sins (Pearl, 1988). The metaphor compares the extensional approach situation with crossing a minefield on a wild horse with good instincts, where we would attach certain weights to it and hope it will keep you out of trouble. To prevent the fast ride from becoming a disaster, highly skilled knowledge engineers are needed. I hope "Wise Pilot 1" could have shown that a fast horse with instincts and skilled knowledge engineers indeed provide a good system, albeit with some sloppiness.

The counterbalance for the horse option is found in the intensional approaches, but the point here is to find a technique that provides a semantically safe approach while avoiding having to examine the entire minefield before each step. I regard the latter option (examining the entire minefield) as the dynamic programming approach to this problem, while the DDN offers a better (faster) approach while keeping the correctness semantic in probability statements.

Indeed, a complex system like "Wise Pilot 2" or even less demanding, non-real-time applications have its limitations when using Bayesian inference. Thus, it is clear to me that using the "check the entire minefield" approach would wind up in a system that may be everything but useful. Following this rationale I discarded the use of the dynamic programming approach in “Wise Pilot 2”.

Also, as I explained in Chapter 4, influence diagrams and Bayesian networks by themselves did not provide the tools necessary to solve the pilot’s problem, and it is not of my knowledge an intensional technique that offered such a complete set of apparatus for dealing with it. This explains my option to do not consider an additional intensional technique for qualitative comparison.

The last issue I would like to comment is the focus on qualitative issues in contrast to a comprehensive, quantitative analysis on the elected techniques. I would say that a quantitative approach would focus on performance related issues, which is important but not in the scope of this work.

It is my opinion that before building a fast system, our concern is to bring a correct, reliable system. A “wild horse” system would achieve an excellent review under quantitative aspects, but most people still rather cross the minefield with a more probabilistic consistent system that satiates the minimum crossing time requirements. Only after established the best qualitative approach for such a system we would go for quantitative measurements among solutions under that approach. This work was focused in defining this approach first.

6.3 Future Trends and Related Research

The growing interest for Bayesian inference techniques is by its own an unambiguous sign that the academic community has already perceived the potential of this approach. Yet, this is a relatively unexplored field and much research is to be done, much literature is to be written, and many applications are likely to keep appearing in the most diverse fields.

As we could perceive in this work, the DDN technique proved its suitability for solving complex problems where other approaches have consistently failed to succeed. However, there is still a gap between the experimental research done here and the implementation of actual applications, and this gap includes not only the need for large scale systems engineering efforts but also more research on the technique itself. Among the areas that could be explored are the points not covered in this work, like the analysis on the options for controlling the value of information and the characteristics of the data fusion processes that might be implemented in multi-sensor systems.

Returning our focus to the case study's problem, the increasing electronic battlefield complexity and the already high levels of "cockpit management" would make the need for automated decision systems a clear trend for future combat aircraft. This Thesis was intended to show that Dynamic Decision Networks is the most suitable approach for transition from a clear trend to reality.

APPENDIX A:

RULES USED IN "WISE PILOT 1" EXPERT SYSTEM

The following list contains the rules used in the "Wise Pilot 1" expert system:

ITERATION context rules

Rule 001

IF: 1) There is a TRACK that is a threat (THREAT = "yes"), and
2) All TRACK were checked on their THREAT parameter, and
3) All TRACK with positive THREAT parameter were checked if they could provide imminent danger.

THEN: 1) Execute the best actions for eliminating all imminent danger, and
2) Execute the best actions for decreasing all THREATLV parameter.

Rule 002

IF: TERR is "enemy".

THEN: THREAT is "yes" for all TRACK.

TRACK context rules

Rule 050

IF: 1) LAUNCH is "yes", and
2) TERR is "enemy", and
3) EMTSS is "none".

THEN: 1) TRACK is a SAM (.8), and
2) SAMTYPE is "portable" (.8), and
3) WEAPONGD is "IR" (.8).

Rule 051

IF: 1) LAUNCH is "yes", and
2) TERR is "neutral", and
3) EMTSS is "none".

THEN: 1) TRACK is a SAM (.2), and
2) SAMTYPE is "portable" (.2), and,
3) WEAPONGD is "IR" (.2).

Rule 052

IF: 1) LAUNCH is “yes”, and
2) TERR is “friend”, and
3) EMTSS is “none”.
THEN: 1) TRACK is a SAM (.1), and
2) SAMTYPE is “unknown”, and,
3) WEAPONGD is “IR” (.1).

Rule 053

IF: STATUS is “friend”.
THEN: 1) THREATLV is “0” (zero), and
2) THREAT is “no”.
ELSE: THREAT is “yes”.

Rule 054

IF: A TRACK has TRACKALT greater than “0” (zero).
THEN: The TRACK is an AIRCRAFT.

Rule 055

IF: THREATLV is greater than three.
THEN: This TRACK is an imminent danger.
ELSE: This TRACK is not an imminent danger.

AIRCRAFT context rules

Rule 100

IF: 1) MODE is “lock”, and
2) TRACKPRJDT is less than “27 nm”.
THEN: 1) THREAT is “yes”, and
2) STATUS is “foe”, and
3) THREATLV is “5”.

Rule 101

IF: AIRCRAFT is not EMITTER.
THEN: ACFTTYPE is “unknown”.

Rule 102

IF: IFF is “off”.
THEN: STATUS is “unknown”.

Rule 103

IF: 1) IFF is “on”, and
2) IFFRET is “no”.
THEN: STATUS is “Foe” (.9).

Rule 104

IF: 1) IFF is “on”, and
2) IFFRET is “yes”.
THEN: STATUS is “Friend”.

Rule 105

IF: 1) FORMATION is “yes”, and
2) There is a TRACK with same F# and STATUS “friend”.
THEN: STATUS is “friend”.

Rule 106

IF: 1) FORMATION is “yes”, and
2) There is a TRACK with same F# and STATUS “foe”.
THEN: STATUS is “foe”.

Rule 107

IF: 1) ACFTTYPE is “unknown”, and
2) TRACKPRJDT is more than “40 nm”, and
3) FORMATION is “no”.
THEN: The THREATLV is “1”.

Rule 108

IF: 1) ACFTTYPE is “unknown”, and
2) STATUS is “unknown”, and
3) TRACKPRJDT is more than “40 nm”, and
4) FORMATION is “yes”, and
5) FREEFT is “no”.
THEN: The THREATLV is “2” (.5).

Rule 109

IF: 1) ACFTTYPE is “unknown”, and
2) STATUS is “unknown”, and
3) TRACKPRJDT is more than “40 nm”, and
4) FORMATION is “yes”, and
5) FREEFT is “yes”.
THEN: The THREATLV is “2” (.8).

Rule 110

IF: 1) ACFTTYPE is “unknown”, and
2) STATUS is “unknown”, and
3) TRACKPRJDT is less than “40 nm”, and
4) TRACKPRJDT is more than “27 nm”, and
5) FORMATION is “no”.
THEN: The THREATLV is “3” (.4).

Rule 111

IF: 1) ACFTTYPE is “unknown”, and
2) STATUS is “unknown”, and
3) TRACKPRJDT is less than “40 nm”, and
4) TRACKPRJDT is more than “27 nm”, and
5) FORMATION is “yes”, and
6) FREEFT is “no”.
THEN: The THREATLV is “3” (.7).

Rule 112

IF: 1) ACFTTYPE is “unknown”, and
2) STATUS is “unknown”, and
3) TRACKPRJDT is less than “40 nm”, and
4) TRACKPRJDT is more than “27 nm”, and
5) FORMATION is “yes”, and
6) FREEFT is “yes”.
THEN: The THREATLV is “3” (.9).

Rule 113

IF: 1) ACFTTYPE is “unknown”, and
2) STATUS is “unknown”, and
3) TRACKPRJDT is less than “27 nm”, and
4) FORMATION is “no”.
THEN: The THREATLV is “3”.

Rule 114

IF: 1) ACFTTYPE is “unknown”, and
2) STATUS is “unknown”, and
3) TRACKPRJDT is less than “27 nm”, and
4) FORMATION is “yes”, and
5) FREEFT is “no”.
THEN: The THREATLV is “4” (.4).

Rule 115

IF: 1) ACFTTYPE is “unknown”, and
2) STATUS is “unknown”, and
3) TRACKPRJDT is less than “27 nm”, and
4) FORMATION is “yes”, and
5) FREEFT is “yes”.
THEN: The THREATLV is “4” (.6).

Rule 116

IF: 1) ACFTTYPE is “unknown”, and
2) STATUS is “foe”, and
3) TRACKPRJDT is more than “40 nm”, and
4) FORMATION is “yes”, and
5) FREEFT is “no”.
THEN: The THREATLV is “2” (.6).

Rule 117

IF: 1) ACFTTYPE is “unknown”, and
2) STATUS is “unknown”, and
3) TRACKPRJDT is more than “40 nm”, and
4) FORMATION is “yes”, and
5) FREEFT is “yes”.
THEN: The THREATLV is “2” .

Rule 118

IF: 1) ACFTTYPE is “unknown”, and
2) STATUS is “foe”, and
3) TRACKPRJDT is less than “40 nm”, and
4) TRACKPRJDT is more than “27 nm”, and
5) FORMATION is “no”.
THEN: The THREATLV is “3”.

Rule 119

IF: 1) ACFTTYPE is “unknown”, and
2) STATUS is “foe”, and
3) TRACKPRJDT is less than “40 nm”, and
4) TRACKPRJDT is more than “27 nm”, and
5) FORMATION is “yes”, and
6) FREEFT is “no”.
THEN: The THREATLV is “4” (.2).

Rule 120

IF: 1) ACFTTYPE is “unknown”, and
2) STATUS is “foe”, and
3) TRACKPRJDT is less than “40 nm”, and
4) TRACKPRJDT is more than “27 nm”, and
5) FORMATION is “yes”, and
6) FREEFT is “yes”.
THEN: The THREATLV is “4” (.7).

Rule 121

IF: 1) ACFTTYPE is “unknown”, and
2) STATUS is “foe”, and
3) TRACKPRJDT is less than “27 nm”.
THEN: The THREATLV is “4”.

Rule 122

IF: 1) ACFTTYPE is “X fighter” or “Y fighter”, and
2) STATUS is “unknown”, and
3) MODE is “lock”, and
4) TRACKPRJDT is less than “40 nm”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “5”.

Rule 123

IF: 1) ACFTTYPE is “X fighter” or “Y fighter”, and
2) STATUS is “unknown”, and
3) MODE is “lock”, and
4) TRACKPRJDT is more than “40 nm”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “4”.

Rule 124

IF: 1) ACFTTYPE is “X fighter” or “Y fighter” or “bomber”, and
2) MODE is “trk”, and
3) TRACKPRJDT is less than “27 nm”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “4”.

Rule 125

IF: 1) ACFTTYPE is “X fighter” or “Y fighter”, and
2) MODE is “trk”, and
3) TRACKPRJDT is less than “40 nm”, and
4) TRACKPRJDT is more than “27 nm”, and
5) FORMATION is “yes”, and
6) FREEFT is “yes”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “4” (.9).

Rule 126

IF: 1) ACFTTYPE is “X fighter” or “Y fighter”, and
2) MODE is “trk”, and
3) TRACKPRJDT is less than “40 nm”, and
4) TRACKPRJDT is more than “27 nm”, and
5) FORMATION is “yes”, and
6) FREEFT is “no”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “4” (.5).

Rule 127

IF: 1) ACFTTYPE is “X fighter” or “Y fighter”, and
2) MODE is “trk”, and
3) TRACKPRJDT is less than “40 nm”, and
4) TRACKPRJDT is more than “27 nm”, and
5) FORMATION is “no”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “4” (.2).

Rule 128

IF: 1) ACFTTYPE is “X fighter” or “Y fighter” or “bomber”, and
2) STATUS is “foe”, and
3) MODE is “acq”, and
4) TRACKPRJDT is less than “27 nm”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “4”.

Rule 129

IF: 1) ACFTTYPE is "X fighter" or "Y fighter", and
2) STATUS is "foe", and
3) MODE is "acq", and
4) TRACKPRJDT is less than "40 nm", and
5) TRACKPRJDT is more than "27 nm", and
6) FORMATION is "yes", and
7) FREEFT is "yes".
THEN: 1) THREAT is "yes", and
2) THREATLV is "4" (.8).

Rule 130

IF: 1) ACFTTYPE is "X fighter" or "Y fighter", and
2) STATUS is "foe", and
3) MODE is "acq", and
4) TRACKPRJDT is less than "40 nm", and
5) TRACKPRJDT is more than "27 nm", and
6) FORMATION is "yes", and
7) FREEFT is "no".
THEN: 1) THREAT is "yes", and
2) THREATLV is "4" (.6).

Rule 131

IF: 1) ACFTTYPE is "X fighter" or "Y fighter", and
2) STATUS is "foe", and
3) MODE is "acq", and
4) TRACKPRJDT is less than "40 nm", and
5) TRACKPRJDT is more than "27 nm", and
6) FORMATION is "no".
THEN: 1) THREAT is "yes", and
2) THREATLV is "4" (.2).

Rule 132

IF: 1) ACFTTYPE is "Y fighter" or "bomber", and
2) STATUS is "unknown", and
3) MODE is "acq", and
4) TRACKPRJDT is less than "27 nm", and
5) FORMATION is "yes", and
7) FREEFT is "yes".
THEN: 1) THREAT is "yes", and
2) THREATLV is "4" (.8).

Rule 133

IF: 1) ACFTTYPE is "Y fighter" or "bomber", and
2) STATUS is "unknown", and
3) MODE is "acq", and
4) TRACKPRJDT is less than "27 nm", and
5) FORMATION is "yes", and
7) FREEFT is "no".
THEN: 1) THREAT is "yes", and
2) THREATLV is "4" (.7).

Rule 134

IF: 1) ACFTTYPE is "Y fighter" or "bomber", and
2) STATUS is "unknown", and
3) MODE is "acq", and
4) TRACKPRJDT is less than "27 nm", and
5) FORMATION is "no".
THEN: 1) THREAT is "yes", and
2) THREATLV is "4" (.4).

Rule 135

IF: 1) ACFTTYPE is "Y fighter", and
2) STATUS is "unknown", and
3) MODE is "acq", and
4) TRACKPRJDT is less than "40 nm", and
5) TRACKPRJDT is more than "27 nm", and
6) FORMATION is "yes", and
7) FREEFT is "yes".
THEN: 1) THREAT is "yes", and
2) THREATLV is "4" (.5).

Rule 136

IF: 1) ACFTTYPE is "Y fighter", and
2) STATUS is "unknown", and
3) MODE is "acq", and
4) TRACKPRJDT is less than "40 nm", and
5) TRACKPRJDT is more than "27 nm", and
6) FORMATION is "yes", and
7) FREEFT is "no".

THEN: 1) THREAT is “yes”, and
2) THREATLV is “4” (.2).

Rule 137

IF: 1) ACFTTYPE is “X fighter” or “Y fighter”, and
2) STATUS is “unknown”, and
3) MODE is “acq”, and
4) TRACKPRJDT is less than “40 nm”, and
5) TRACKPRJDT is more than “27 nm”, and
6) FORMATION is “no”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “3”.

Rule 138

IF: 1) ACFTTYPE is “X fighter” or “Y fighter” or “bomber”, and
2) MODE is not “lock”, and
3) TRACKPRJDT is more than “40 nm”, and
4) FORMATION is “yes”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “2”.

Rule 139

IF: 1) ACFTTYPE is “X fighter” or “Y fighter” or “bomber”, and
2) MODE is not “lock”, and
3) TRACKPRJDT is more than “40 nm”, and
4) FORMATION is “no”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “1”.

Rule 140

IF: 1) ACFTTYPE is “X fighter”, and
2) STATUS is “unknown”, and
3) MODE is “acq”, and
4) TRACKPRJDT is less than “27 nm”, and
5) FORMATION is “yes”, and
6) FREEFT is “yes”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “4” (.9).

Rule 141

IF: 1) ACFTTYPE is “X fighter”, and
2) STATUS is “unknown”, and
3) MODE is “acq”, and
4) TRACKPRJDT is less than “27 nm”, and
5) FORMATION is “yes”, and
6) FREEFT is “no”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “4” (.8).

Rule 142

IF: 1) ACFTTYPE is “X fighter”, and
2) STATUS is “unknown”, and
3) MODE is “acq”, and
4) TRACKPRJDT is less than “27 nm”, and
5) FORMATION is “no”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “4” (.6).

Rule 143

IF: 1) ACFTTYPE is “X fighter”, and
2) STATUS is “unknown”, and
3) MODE is “acq”, and
4) TRACKPRJDT is less than “40 nm”, and
5) TRACKPRJDT is more than “27 nm”, and
6) FORMATION is “yes”, and
7) FREEFT is “yes”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “4” (.5).

Rule 144

IF: 1) ACFTTYPE is “X fighter”, and
2) STATUS is “unknown”, and
3) MODE is “acq”, and
4) TRACKPRJDT is less than “40 nm”, and
5) TRACKPRJDT is more than “27 nm”, and
6) FORMATION is “yes”, and
7) FREEFT is “no”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “4” (.4).

Rule 145

IF: 1) ACFTTYPE is “Bomber”, and
2) STATUS is “unknown”, and
3) MODE is “lock”, and
4) TRACKPRJDT is more than “27 nm”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “4”.

Rule 146

IF: 1) ACFTTYPE is “Bomber”, and
2) MODE is “trk”, and
3) TRACKPRJDT is more than “27 nm”, and
4) FORMATION is “yes”, and
5) FREEFT is “yes”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “3” (.9).

Rule 147

IF: 1) ACFTTYPE is “Bomber”, and
2) MODE is “trk”, and
3) TRACKPRJDT is more than “27 nm”, and
4) FORMATION is “yes”, and
5) FREEFT is “no”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “3” (.6).

Rule 148

IF: 1) ACFTTYPE is “Bomber”, and
2) MODE is “trk”, and
3) TRACKPRJDT is more than “27 nm”, and
4) FORMATION is “no”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “3” (.2).

Rule 149

IF: 1) ACFTTYPE is “Bomber”, and
2) STATUS is “foe”, and
3) MODE is “acq”, and
4) TRACKPRJDT is more than “27 nm”, and
5) FORMATION is “yes”, and
6) FREEFT is “yes”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “3” (.8).

Rule 150

IF: 1) ACFTTYPE is “Bomber”, and
2) STATUS is “foe”, and
3) MODE is “acq”, and
4) TRACKPRJDT is more than “27 nm”, and
5) FORMATION is “yes”, and
6) FREEFT is “no”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “3” (.6).

Rule 151

IF: 1) ACFTTYPE is “Bomber”, and
2) STATUS is “foe”, and
3) MODE is “acq”, and
4) TRACKPRJDT is more than “27 nm”, and
5) FORMATION is “no”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “3” (.2).

Rule 152

IF: 1) ACFTTYPE is “Bomber”, and
2) STATUS is “unknown”, and
3) MODE is “acq”, and
4) TRACKPRJDT is more than “27 nm”, and
5) FORMATION is “yes”, and
6) FREEFT is “yes”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “3” (.5).

Rule 153

IF: 1) ACFTTYPE is “Bomber”, and
2) STATUS is “unknown”, and
3) MODE is “acq”, and
4) TRACKPRJDT is more than “27 nm”, and
5) FORMATION is “yes”, and
6) FREEFT is “no”.
THEN: 1) THREAT is “yes”, and
2) THREATLV is “3” (.2).

Rule 154

IF: 1) ACFTTYPE is “Bomber”, and
2) STATUS is “unknown”, and
3) MODE is “acq”, and
4) TRACKPRJDT is more than “27 nm”, and
5) FORMATION is “no”.

THEN: 1) THREAT is “yes”, and
2) THREATLV is “3”.

Rule 155

IF: 1) ACFTTYPE is “Blue fighter” or “Unarmed”, and
2) MODE is not “lock”.
THEN: THREAT is “no”.

AAA rules

Rule 200

IF: 1) TRACK is AAA, and
2) MODE is “lock”, and
3) ALTITUDE is equal or less than AAAEA.
THEN: THREATLV is “5”.

Rule 201

IF: 1) TRACK is AAA, and
2) MODE is “lock”, and
3) ALTITUDE is greater than AAAEA.
THEN: THREATLV is “4”.

Rule 202

IF: 1) AAATYPE is “AAA1” or “AAA2”, and
2) MODE is “trk”, and
3) EMTSS is “high”
4) ALTITUDE is equal or less than AAAEA.
THEN: THREATLV is “4” (.8).

Rule 203

IF: 1) AAATYPE is “AAA1” or “AAA2”, and
2) MODE is “trk”, and
3) EMTSS is “medium”
4) ALTITUDE is equal or less than AAAEA.
THEN: THREATLV is “4” (.4).

Rule 204

IF: 1) AAATYPE is “AAA1” or “AAA2”, and
2) MODE is “trk”, and
3) EMTSS is “low”
4) ALTITUDE is equal or less than AAAEA.
THEN: THREATLV is “4” (.2).

Rule 205

IF: 1) AAATYPE is “AAA1” or “AAA2”, and
2) MODE is “trk”, and
3) EMTSS is “high”
4) ALTITUDE is greater than AAAEA.
THEN: THREATLV is “4” (.5).

Rule 206

IF: 1) AAATYPE is “AAA1” or “AAA2”, and
2) MODE is “trk”, and
3) EMTSS is “medium”
4) ALTITUDE is greater than AAAEA.
THEN: THREATLV is “4” (.4).

Rule 207

IF: 1) AAATYPE is “AAA1” or “AAA2”, and
2) MODE is “trk”, and
3) EMTSS is “low”
4) ALTITUDE is greater than AAAEA.
THEN: THREATLV is “3” (.9).

Rule 208

IF: 1) AAATYPE is “AAA1” or “AAA2”, and
2) MODE is “acq”, and
3) EMTSS is “high”
4) ALTITUDE is equal or less than AAAEA.
THEN: THREATLV is “3” (.8).

Rule 209

IF: 1) AAATYPE is “AAA1” or “AAA2”, and
2) MODE is “acq”, and
3) EMTSS is “medium”
4) ALTITUDE is equal or less than AAAEA.
THEN: THREATLV is “3” (.4).

Rule 210

IF: 1) AAATYPE is “AAA1” or “AAA2”, and
2) MODE is “acq”, and
3) EMTSS is “low”
4) ALTITUDE is equal or less than AAAEA.
THEN: THREATLV is “2” (.2).

Rule 211

IF: 1) AAATYPE is “AAA1” or “AAA2”, and
2) MODE is “acq”, and
3) EMTSS is “high”
4) ALTITUDE is greater than AAAEA.
THEN: THREATLV is “3” (.5).

Rule 212

IF: 1) AAATYPE is “AAA1” or “AAA2”, and
2) MODE is “acq”, and
3) EMTSS is “medium”
4) ALTITUDE is greater than AAAEA.
THEN: THREATLV is “2” (.4).

Rule 213

IF: 1) AAATYPE is “AAA1” or “AAA2”, and
2) MODE is “acq”, and
3) EMTSS is “low”
4) ALTITUDE is greater than AAAEA.
THEN: THREATLV is “2” (.9).

Rule 214

IF: 1) AAATYPE is “AAA3”, and
2) MODE is “trk”, and
3) EMTSS is “high”
4) ALTITUDE is equal or less than AAAEA.
THEN: THREATLV is “4” (.9).

Rule 215

IF: 1) AAATYPE is “AAA3”, and
2) MODE is “trk”, and
3) EMTSS is “medium”
4) ALTITUDE is equal or less than AAAEA.
THEN: THREATLV is “4” (.6).

Rule 216

IF: 1) AAATYPE is “AAA3”, and
2) MODE is “trk”, and
3) EMTSS is “low”
4) ALTITUDE is equal or less than AAAEA.
THEN: THREATLV is “4” (.4).

Rule 217

IF: 1) AAATYPE is “AAA3”, and
2) MODE is “trk”, and
3) EMTSS is “high”
4) ALTITUDE is greater than AAAEA.
THEN: THREATLV is “4” (.7).

Rule 218

IF: 1) AAATYPE is “AAA3”, and
2) MODE is “trk”, and
3) EMTSS is “medium”
4) ALTITUDE is greater than AAAEA.
THEN: THREATLV is “4” (.6).

Rule 219

IF: 1) AAATYPE is “AAA3”, and
2) MODE is “trk”, and
3) EMTSS is “low”
4) ALTITUDE is greater than AAAEA.
THEN: THREATLV is “3”.

Rule 220

IF: 1) AAATYPE is “AAA3”, and
2) MODE is “acq”, and
3) EMTSS is “high”
4) ALTITUDE is equal or less than AAAEA.
THEN: THREATLV is “3” (.9).

Rule 221

IF: 1) AAATYPE is “AAA3”, and
2) MODE is “acq”, and
3) EMTSS is “medium”
4) ALTITUDE is equal or less than AAAEA.
THEN: THREATLV is “3” (.6).

Rule 222

IF: 1) AAATYPE is “AAA3”, and
2) MODE is “acq”, and
3) EMTSS is “low”
4) ALTITUDE is equal or less than AAAEA.
THEN: THREATLV is “3” (.4).

Rule 223

IF: 1) AAATYPE is “AAA3”, and
2) MODE is “acq”, and
3) EMTSS is “high”
4) ALTITUDE is greater than AAAEA.
THEN: THREATLV is “3” (.7).

Rule 224

IF: 1) AAATYPE is “AAA3”, and
2) MODE is “acq”, and
3) EMTSS is “medium”
4) ALTITUDE is greater than AAAEA.
THEN: THREATLV is “2” (.2).

Rule 225

IF: 1) AAATYPE is “AAA3”, and
2) MODE is “acq”, and
3) EMTSS is “low”
4) ALTITUDE is greater than AAAEA.
THEN: THREATLV is “2”.

SAM rules

Rule 300

IF: SAMTYPE is “portable” or “unknown”.
THEN: THREATLV is “5”.

Rule 301

IF: 1) TRACK is SAM, and
2) MODE is “lock”, and
3) ALTITUDE is less than SAMEA.
THEN: THREATLV is “5”.

Rule 302

IF: 1) TRACK is SAM, and
2) MODE is “lock”, and
3) ALTITUDE is greater than SAMEA.
THEN: THREATLV is “4”.

Rule 303

IF: 1) SAMTYPE is “SAM1”, and
2) MODE is “trk”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “high”.
THEN: THREATLV is “4” (.8).

Rule 304

IF: 1) SAMTYPE is “SAM1”, and
2) MODE is “trk”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “medium”.
THEN: THREATLV is “4” (.5).

Rule 305

IF: 1) SAMTYPE is “SAM1”, and
2) MODE is “trk”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “low”.
THEN: THREATLV is “4” (.4).

Rule 306

IF: 1) SAMTYPE is “SAM1”, and
2) MODE is “trk”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “high”.
THEN: THREATLV is “3” (.8).

Rule 307

IF: 1) SAMTYPE is “SAM1”, and
2) MODE is “trk”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “medium”.
THEN: THREATLV is “3” (.5).

Rule 308

IF: 1) SAMTYPE is “SAM1”, and
2) MODE is “trk”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “low”.
THEN: THREATLV is “3” (.4).

Rule 309

IF: 1) SAMTYPE is “SAM1”, and
2) MODE is “acq”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “high”.
THEN: THREATLV is “3” (.8).

Rule 310

IF: 1) SAMTYPE is “SAM1”, and
2) MODE is “acq”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “medium”.
THEN: THREATLV is “3” (.5).

Rule 311

IF: 1) SAMTYPE is “SAM1”, and
2) MODE is “acq”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “low”.
THEN: THREATLV is “3” (.4).

Rule 312

IF: 1) SAMTYPE is “SAM1”, and
2) MODE is “acq”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “high”.
THEN: THREATLV is “3” (.6).

Rule 313

IF: 1) SAMTYPE is “SAM1”, and
2) MODE is “acq”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “medium”.
THEN: THREATLV is “2” (.5).

Rule 314

IF: 1) SAMTYPE is “SAM1”, and
2) MODE is “acq”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “low”.
THEN: THREATLV is “2” (.8).

Rule 315

IF: 1) SAMTYPE is “SAM2”, and
2) MODE is “trk”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “high”.
THEN: THREATLV is “4”.

Rule 316

IF: 1) SAMTYPE is “SAM2”, and
2) MODE is “trk”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “medium”.
THEN: THREATLV is “4” (.7).

Rule 317

IF: 1) SAMTYPE is “SAM2”, and
2) MODE is “trk”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “low”.
THEN: THREATLV is “4” (.5).

Rule 318

IF: 1) SAMTYPE is “SAM2”, and
2) MODE is “trk”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “high”.
THEN: THREATLV is “3” (.8).

Rule 319

IF: 1) SAMTYPE is “SAM2”, and
2) MODE is “trk”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “medium”.
THEN: THREATLV is “3” (.6).

Rule 320

IF: 1) SAMTYPE is “SAM2”, and
2) MODE is “trk”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “low”.
THEN: THREATLV is “3” (.5).

Rule 321

IF: 1) SAMTYPE is “SAM2”, and
2) MODE is “acq”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “high”.
THEN: THREATLV is “3” (.9).

Rule 322

IF: 1) SAMTYPE is “SAM2”, and
2) MODE is “acq”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “medium”.
THEN: THREATLV is “3” (.7).

Rule 323

IF: 1) SAMTYPE is “SAM2”, and
2) MODE is “acq”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “low”.
THEN: THREATLV is “3” (.6).

Rule 324

IF: 1) SAMTYPE is “SAM2”, and
2) MODE is “acq”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “high”.
THEN: THREATLV is “3” (.8).

Rule 325

IF: 1) SAMTYPE is “SAM2”, and
2) MODE is “acq”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “medium”.
THEN: THREATLV is “2” (.3).

Rule 326

IF: 1) SAMTYPE is “SAM2”, and
2) MODE is “acq”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “low”.
THEN: THREATLV is “2” (.6).

Rule 327

IF: 1) SAMTYPE is “SAM3”, and
2) MODE is “trk”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “high”.
THEN: THREATLV is “4” (.6).

Rule 328

IF: 1) SAMTYPE is “SAM3”, and
2) MODE is “trk”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “medium”.
THEN: THREATLV is “4” (.3).

Rule 329

IF: 1) SAMTYPE is “SAM3”, and
2) MODE is “trk”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “low”.
THEN: THREATLV is “4” (.2).

Rule 330

IF: 1) SAMTYPE is “SAM3”, and
2) MODE is “trk”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “high”.
THEN: THREATLV is “3” (.7).

Rule 331

IF: 1) SAMTYPE is “SAM3”, and
2) MODE is “trk”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “medium”.
THEN: THREATLV is “3” (.4).

Rule 332

IF: 1) SAMTYPE is “SAM3”, and
2) MODE is “trk”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “low”.
THEN: THREATLV is “3” (.3).

Rule 333

IF: 1) SAMTYPE is “SAM3”, and
2) MODE is “acq”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “high”.
THEN: THREATLV is “3” (.7).

Rule 334

IF: 1) SAMTYPE is “SAM3”, and
2) MODE is “acq”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “medium”.
THEN: THREATLV is “3” (.4).

Rule 335

IF: 1) SAMTYPE is “SAM3”, and
2) MODE is “acq”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “low”.
THEN: THREATLV is “3” (.3).

Rule 336

IF: 1) SAMTYPE is “SAM3”, and
2) MODE is “acq”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “high”.
THEN: THREATLV is “3” (.5).

Rule 337

IF: 1) SAMTYPE is “SAM3”, and
2) MODE is “acq”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “medium”.
THEN: THREATLV is “2” (.6).

Rule 338

IF: 1) SAMTYPE is “SAM3”, and
2) MODE is “acq”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “low”.
THEN: THREATLV is “2” (.9).

Rule 339

IF: 1) SAMTYPE is “SAM4”, and
2) MODE is “trk”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “high”.
THEN: THREATLV is “4” (.5).

Rule 340

IF: 1) SAMTYPE is “SAM4”, and
2) MODE is “trk”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “medium”.
THEN: THREATLV is “4” (.3).

Rule 341

IF: 1) SAMTYPE is “SAM4”, and
2) MODE is “trk”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “low”.
THEN: THREATLV is “4” (.2).

Rule 342

IF: 1) SAMTYPE is “SAM4”, and
2) MODE is “trk”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “high”.
THEN: THREATLV is “3” (.5).

Rule 343

IF: 1) SAMTYPE is “SAM4”, and
2) MODE is “trk”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “medium”.
THEN: THREATLV is “3” (.2).

Rule 344

IF: 1) SAMTYPE is “SAM4”, and
2) MODE is “trk”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “low”.
THEN: THREATLV is “3” (.2).

Rule 345

IF: 1) SAMTYPE is “SAM4”, and
2) MODE is “acq”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “high”.
THEN: THREATLV is “3” (.8).

Rule 346

IF: 1) SAMTYPE is “SAM4”, and
2) MODE is “acq”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “medium”.
THEN: THREATLV is “3” (.2).

Rule 347

IF: 1) SAMTYPE is “SAM4”, and
2) MODE is “acq”, and
3) ALTITUDE is less than SAMEA, and
4) EMTSS is “low”.
THEN: THREATLV is “3” (.2).

Rule 348

IF: 1) SAMTYPE is “SAM4”, and
2) MODE is “acq”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “high”.
THEN: THREATLV is “3” (.3).

Rule 349

IF: 1) SAMTYPE is “SAM4”, and
2) MODE is “acq”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “medium”.
THEN: THREATLV is “2” (.8).

Rule 350

IF: 1) SAMTYPE is “SAM4”, and
2) MODE is “acq”, and
3) ALTITUDE is greater than SAMEA, and
4) EMTSS is “low”.
THEN: THREATLV is “2” .

ACQ rules

Rule 400

IF: 1) ACQID is “Y”, and
2) TRACKSS is “high”.
THEN: THREATLV is “2”

Rule 401

IF: 1) ACQID is “Y”, and
2) TRACKSS is “medium”.
THEN: THREATLV is “1” (.5).

Rule 402

IF: 1) ACQID is “Y”, and
2) TRACKSS is “low”.
THEN: THREATLV is “1”

EMITTER rules

Rule 500

IF: EMTID is “EM001”.
THEN: ACFTTYPE is “X fighter”
WEAPONGD is “RG”.

Rule 501

IF: EMTID is “EM002”, and
MODE is not “acq”.
THEN: ACFTTYPE is “Y fighter” (.5).
WEAPONGD is “RG”.

Rule 502

IF: EMTID is “EM002”, and
MODE is “acq”.
THEN: ACFTTYPE is “Y fighter” (.2).
WEAPONGD is “RG”.

Rule 503

IF: EMTID is “EM003”.
THEN: ACFTTYPE is “Blue Fighter”

Rule 504

IF: EMTID is “EM004”.
THEN: 1) TRACK is an ACQ, and
2) ACQID is “Y”, and
3) ACQRG is “40 nm”.

Rule 505

IF: EMTID is “EM005”.
THEN: 1) TRACK is a SAM, and
2) SAMTYPE is “SAM1”, and
3) SAMEA is “360”, and
4) SAMRG is “13 nm”, and
5) ACQRG is “40 nm”, and
6) LOCKRG is “15 nm”, and
7) WEAPONGD is “RG”.

Rule 506

IF: EMTID is “EM006”.
THEN: 1) TRACK is a SAM, and
2) SAMTYPE is “SAM2”, and
3) SAMEA is “360”, and
4) SAMRG is “15 nm”, and
5) ACQRG is “40 nm”, and
6) LOCKRG is “24 nm”, and
7) WEAPONGD is “RG”.

Rule 507

IF: EMTID is “EM007”.
THEN: 1) TRACK is a SAM, and
2) SAMTYPE is “SAM3”, and
3) SAMEA is “330”, and
4) SAMRG is “8 nm”, and
5) ACQRG is “19 nm”, and
6) LOCKRG is “14 nm”, and
7) WEAPONGD is “RG”.

Rule 508

IF: EMTID is “EM008”.
THEN: 1) TRACK is a SAM (.5), and
2) SAMTYPE is “SAM4”, and
3) SAMEA is “330”, and
4) SAMRG is “5 nm”, and
5) ACQRG is “10 nm”, and
6) LOCKRG is “9 nm”, and
7) WEAPONGD is “RG”.

Rule 509

IF: EMTID is “EM009”.
THEN: 1) TRACK is AAA, and
2) AAATYPE is “AAA2” (.7), and
3) AAAEA is “050”, and
4) AAARG is “4 nm”, and
5) ACQRG is “10 nm”, and
6) LOCKRG is “9 nm”, and
7) WEAPONGD is “RG”.

ACTION rules

Rule 600

IF: 1) THREATLV is “5”, and
2) SAMTYPE is “portable”, and
3) AUTOPILOT is “off”, and
4) SECTOR is “6”.
THEN: 1) Launch FLARE01, and
2) Execute MNV6 to SAFLFSEC.

Rule 601

IF: 1) THREATLV is “5”, and
2) SAMTYPE is “portable”, and
3) AUTOPILOT is “off”, and
4) SECTOR is “5”.
THEN: 1) Launch FLARE01, and
2) Execute MNV5 to SAFLFSEC.

Rule 602

IF: 1) THREATLV is “5”, and
2) SAMTYPE is “portable”, and
3) AUTOPILOT is “off”, and
4) SECTOR is “4”.
THEN: 1) Launch FLARE01, and
2) Execute MNV4 to SAFLFSEC.

Rule 603

IF: 1) THREATLV is “5”, and
2) SAMTYPE is “portable”, and
3) AUTOPILOT is “off”, and
4) SECTOR is “3”.
THEN: 1) Launch FLARE02, and
2) Execute MNV3 to SAFLFSEC.

Rule 604

IF: 1) THREATLV is “5”, and
2) SAMTYPE is “portable”, and
3) AUTOPILOT is “off”, and
4) SECTOR is “2”.
THEN: 1) Launch FLARE02, and
2) Execute MNV2 to SAFLFSEC.

Rule 605

IF: 1) THREATLV is “5”, and
2) SAMTYPE is “portable”, and
3) AUTOPILOT is “off”, and
4) SECTOR is “1”.
THEN: 1) Launch FLARE03, and
2) Execute MNV1 to SAFLFSEC.

Rule 606

IF: 1) THREATLV is “5”, and
2) WEAPONGD is IR, and
3) AUTOPILOT is “full”, and
4) SECTOR is “6”.
THEN: 1) Launch FLARE01, and
2) Execute MNV6 to SAFLFSEC.

Rule 607

IF: 1) THREATLV is “5”, and
2) WEAPONGD is IR, and
3) AUTOPILOT is “full”, and
4) SECTOR is “5”.
THEN: 1) Launch FLARE01, and
2) Execute MNV5 to SAFLFSEC.

Rule 608

IF: 1) THREATLV is “5”, and
2) WEAPONGD is IR, and
3) AUTOPILOT is “full”, and
4) SECTOR is “4”.
THEN: 1) Launch FLARE01, and
2) Execute MNV4 to SAFLFSEC.

Rule 609

IF: 1) THREATLV is “5”, and
2) WEAPONGD is IR, and
3) AUTOPILOT is “full”, and
4) SECTOR is “3”.
THEN: 1) Launch FLARE02, and
2) Execute MNV3 to SAFLFSEC.

Rule 610

IF: 1) THREATLV is “5”, and
2) WEAPONGD is IR, and
3) AUTOPILOT is “full”, and
4) SECTOR is “2”.
THEN: 1) Launch FLARE02, and
2) Execute MNV2 to SAFLFSEC.

Rule 611

IF: 1) THREATLV is “5”, and
2) WEAPONGD is IR, and
3) AUTOPILOT is “full”, and
4) SECTOR is “1”.
THEN: 1) Launch FLARE03, and
2) Execute MNV1 to SAFLFSEC.

Rule 612

IF: 1) THREATLV is “5”, and
2) WEAPONGD is RG, and
3) AUTOPILOT is “full”, and
4) SECTOR is “6”.
THEN: 1) Launch the respective ECMPRG, and
2) Execute MNV6 to SAFLFSEC.

Rule 613

IF: 1) THREATLV is “5”, and
2) WEAPONGD is RG, and
3) AUTOPILOT is “full”, and
4) SECTOR is “5”.
THEN: 1) Launch the respective ECMPRG, and
2) Execute MNV5 to SAFLFSEC.

Rule 614

IF: 1) THREATLV is “5”, and
2) WEAPONGD is RG, and
3) AUTOPILOT is “full”, and
4) SECTOR is “4”.
THEN: 11) Launch the respective ECMPRG, and
2) Execute MNV4 to SAFLFSEC.

Rule 615

IF: 1) THREATLV is “5”, and
2) WEAPONGD is RG, and
3) AUTOPILOT is “full”, and
4) SECTOR is “3”.
THEN: 1) Launch the respective ECMPRG, and
2) Execute MNV3 to SAFLFSEC.

Rule 616

IF: 1) THREATLV is “5”, and
2) WEAPONGD is RG, and
3) AUTOPILOT is “full”, and
4) SECTOR is “2”.
THEN: 1) Launch the respective ECMPRG, and
2) Execute MNV2 to SAFLFSEC.

Rule 617

IF: 1) THREATLV is “5”, and
2) WEAPONGD is RG, and
3) AUTOPILOT is “full”, and
4) SECTOR is “1”.
THEN: 1) Launch the respective ECMPRG, and
2) Execute MNV1 to SAFLFSEC.

Rule 618

IF: 1) THREATLV is “4”, and
2) TRACK is AIRCRAFT, and
3) FORMATION is “no”.
THEN: 1) Set the radar to a high mode, and
2) Engage the semi-active missile on TRACK, and
3) Launch the missile as soon as a Lock-on Target is achieved.

Rule 619

IF: 1) THREATLV is “4”, and
2) TRACK is AIRCRAFT, and
3) FORMATION is “yes”.
THEN: 1) Set radar to a high mode, and
2) Engage the semi-active missile on TRACK, and
3) Launch the missile as soon as a Lock-on Target is achieved, and
4) Launch the respective ECMPRG against other tracks of formation.

Rule 620

IF: 1) THREATLV is “4”, and
2) TRACK is not AIRCRAFT.

THEN: 1) Deviate from track, and
2) Launch the respective ECMPRG.

Rule 621

IF: 1) THREATLV is “3”, and
2) TRACK is AIRCRAFT.
THEN: 1) Set the radar to a high mode, and
2) Engage the semi-active missile on TRACK.

Rule 622

IF: 1) THREATLV is “3”, and
2) TRACK is not AIRCRAFT.
THEN: Deviate from track.

Rule 623

IF: 1) THREATLV is not “1”, and
2) STATUS is “unknown”.
THEN: 1) Turn IFF “on”, and
2) Send an IFF inquiry.

Rule 624

IF: 1) THREATLV is “2”, and
2) STATUS is not “unknown”.
THEN: No further action is needed.

Rule 625

IF: 1) THREATLV is “1”.
THEN: No further action is needed.

APPENDIX B:

RULE CHAINING IN "WISE PILOT 1" FOR SOLVING THE SCENARIOS

Solving the First Scenario

In the following paragraphs, we will analyze what would be happening during a program's ITERATION in every snapshot of the first scenario. This will enable us to follow the system's rationale when dealing with this particular scenario.

The system will start by evaluating the premises of the goal rule, which is Rule 001 for this particular system. Following the MONITOR's logic scheme, the system will realize that there is still no TRACK context node in its working memory, thus FINDOUT must be called to see if there is any track detected by the sensors.

FINDOUT will return that the aircraft radar sensed two tracks. This will led the system to create automatically two new TRACK context nodes and fill their respective attributes with the available data. Being aware of the two tracks, the next step is to classify them, in order to be able to verify whether or not they are threats.

Like MYCIN, "Wise Pilot 1" uses the backward chaining reasoning with a deep-first search method, so it will look extensively through the TRACK1 branch until being able to confirm or deny premise 1, that is to define whether TRACK1 could be a threat or not. Only after this definition, the system will change for the TRACK2 branch. Here, we assume that TRACK1 is represented in the first scenario's representation as track "A", while TRACK2 corresponds to track "B".

Rule 002 will not be triggered in this scenario, since the parameter TERR is not "enemy", so MONITOR will skip to the next rule that would be capable to define the THREAT parameter for the TRACK context nodes.

Rule 053 tells that TRACK1 would be considered a threat if its STATUS is other than friend, but there is still no information about STATUS of TRACK1. Since there are no further track context rules that states about the STATUS parameter, the system will have to find out what is TRACK1's third level context before visit other context rules. That makes Rule 054 the next to be visited.

For defining whether or not TRACK1 in an aircraft, Rule 054 needs only the TRACKALT parameter, which is a sensor given data. After receiving the return from FINDOUT, the system realizes that TRACK1 is an AIRCRAFT and then goes for Rule 100, as it is the first one to state about TRACK1's STATUS. However, Rule 100 fails in its first premise, since there is no emissions in lock mode, which makes Rule 102 the next one to be visited.

Rule 102 has a unique premise, which returns true whenever the aircraft's IFF is "OFF". IFF is an ITERATION context parameter and is already known as "OFF". Therefore, the premise is verified and MONITOR finds out that the STATUS parameter has "unknown" as its current value. Thus, Rule 053's first premise fails, and the clause ELSE defines TRACK1 as a threat, which makes Rule 001's first premise true. MONITOR is now able to go for the second premise of Rule 001.

Verifying premise 2 implicitly requires all TRACK context nodes to be visited, so MONITOR begins to check TRACK2 branch on the context tree. Once again, the program will perform the same rule sequence for TRACK2 and will wind up to the same results. This will make premise 2 of Rule 001 true and MONITOR will go for the last premise of Rule 001.

Again, all TRACK context nodes will have to be visited. TRACK1 is the first to be checked on its imminent danger. Rule 055 requires the THREATLV parameter's value for defining whether a track is an imminent danger, so MONITOR will attempt to apply an aircraft context rule that states about this parameter. Rule 100 is the first on the list, but it fails in its first premise (no emissions were detected for TRACK1), so the Rule 107 becomes the next in line.

ACFTTYPE is "unknown" so the rule's first premise is true. The next premise requires knowledge about parameter TRACKPRJDT, which is the track's projected distance for the next program's iteration. This parameter is calculated by an internal algorithm that uses radar information on distance, bearing, and velocity of that track to predict were it will be after the time estimated for the next iteration. For this sample system, it is fair to assume that the interpolation's result will be greater than 40 nm, which makes the second premise true.

Rule 107's last premise requires the parameter FORMATION, which defines whether or not the respective track is flying in an aircraft formation. Again, an internal algorithm, based on radar data will give this answer. Since TRACK1 have no detected tracks closer than the algorithm's threshold, it returns as not in formation. Therefore, the last premise passes and TRACK1's THREATLV parameter is set to "1". TRACK2 is then checked by the same rule with the same output, which makes Rule 001's third premise return as true.

Given that all premises are true, MONITOR attempts to execute action 1. Since there is no imminent danger assigned to any TRACK, the program will skip directly to action 2. Rule 625 ends this ITERATION cycle with no further actions to be made by the program at that time.

After the ITERATION we have just analyzed, twenty seconds will pass before the next snapshot. During this period the program will perform a certain number of ITERATION cycles. This number is directly related to the computer's processing power, the number of TRACK nodes to be checked, and the depth to be reached in each branch (TRACK), the latter being a consequence of the scenario's complexity.

I will now analyze what will be happening during the second snapshot's ITERATION. In order to maintain coherence with the pictured scenario, I will assume that the aircraft's sensors kept contact with TRACK1 and TRACK2 during all subsequent ITERATION cycles before the second snapshot's ITERARION. This means that no ITERATION level parameters have changed, and since no action or change was required by the first snapshot's ITERATION, the system still has previous data on those TRACK context nodes. However, most TRACK parameters are volatile[15], which is to say they are not passed to the next ITERATION. THREATLV for example (needed by Rule 055) is a TRACK parameter, so it had no value at the beginning of current ITERATION.

The same inquiry of Rule 001 will show a new track (point C in Figure 10) and a new TRACK context node (TRACK3) will be created on the context tree. This time, the system knows that both TRACK1 and TRACK2 are threats[16] so premise 1 of the goal rule is true and the query jumps directly to premise 2. This makes MONITOR attempting to define whether TRACK3 is a threat. The same sequence is then repeated and realizes that that TRACK3 is also a threat, so MONITOR goes to the third premise of Rule 001 and begins to analyze TRACK1's THREATLV parameter.

The sequence Rule 001 ⇒ Rule 055 ⇒ Rule 053 (clause ELSE gives no clue on THREATLV) ⇒ Rule 100 (fails on first premise) ⇒ Rule 107 (calls Rule 101 for the

[15] With the exemptions of "CUR*" parameters, that are passed to the next ITERATION as "LIT *" parameters (e.g. CURBEAR is passed to the next ITERATION as LITBEAR), and the logical parameters ("yes" or "no" attributes).

[16] THREAT parameter is a logical parameter, thus it is not volatile.

first premise but fails on third premise) $\Rightarrow$ Rule 108 will define TRACK1's and TRACK3's THREATLV as "2". In the TRACK2 case, however, the sequence will be the same until it attempts to apply Rule 101 for validating Rule 107's first premise, since TRACK2 is now emitting. When trying to realize whether TRACK2 is an "unknown" aircraft (thus validating Rule 107's first premise), MONITOR will have to define the parameter ACFTTYPE.

There are no more rules stating on ACFTTYPE, so MONITOR jumps to the next context available for TRACK2 and reaches the emitter list. From this list, Rule 501 will have all of its premises verified so it defines TRACK2's ACFTTYPE parameter as "Y fighter".

This is an interesting point to stop and realize how a direct translation of Bayesian probabilities into CF is not suitable for using in this situation. The probability of TRACK2 being an enemy Y fighter is .25 (25%), however the two hypothesis that support this assessment were not independent. This means that once you know hypothesis Y (TRACK is an Y fighter) is true, your assessment on the probability of hypothesis E (TRACK is an enemy) will change. The CF model is based on the independence of its hypothesis, and this is not true in this specific situation.

Moreover, even if the two hypothesis were independent and we were able to construct a rule stating that "EM002" emitter is an Y fighter with a CF(Y) based on the Bayesian probabilities, a .25 score would indicate a negative CF (or positive MD). This rule would lead the program to consider "EM002" as not a THREAT, which is obviously a dangerous assumption.

To avoid such problems, the strategy used by "Wise Pilot 1" considers "EM002" as an "Y fighter" (in that specific situation) but does not state about its STATUS (friend or foe). In spite of this, the system implicitly considers the unknown STATUS parameter as being "foe", since it treats the TRACK as a threat. This strategy tries to mimic what would happen in a real case, where the worst case shall be considered for safety, while the STATUS assessment is made with the support of other evidences or actions.

After considering TRACK2 a "Y fighter, MONITOR returns to Rule 107 (and to the aircraft context list or rules), and acknowledges that its first premise failed. Then, the next step is to keep trying to find a rule that may define the THREATLV parameter, which is accomplished when Rule 139 establishes TRACK2's THREATLV as "1".

Now, the program will go for the action clauses of Rule 001 and the first one is immediately discarded by the results of Rule 055. Action 2 states about decreasing THREATLV. This fairly general assertion is made only for triggering the program's execution of the action rules. From those, just Rule 623 will have its premises passed, so the IFF will be turned on and an inquiry will be sent.

This second iteration showed an example of the implicit knowledge carried by the system. An aircraft with THREATLV greater than 1 is approaching (projected distance decreases) and there is no means for verifying its STATUS other than sending an IFF inquiry. A highly trained fighter pilot is supposed to have the same reaction, ideally with the same timing.

The third snapshot is the last of this scenario. Again, we already have a THREAT so the first premise will pass with no further inquiries. However, since this time there is no new TRACK the second premise of Rule 001 will also pass and MONITOR will proceed for checking the third premise.

Rule 001's third premise leads MONITOR toward Rule 055, where the program verifies whether TRACK is an imminent danger of not by verifying its THREATLV. Since there is no THREATLV assigned (recall volatility of TRACK parameters) MONITOR triggers the first rule that states about it, Rule 053, which first premise demands the STATUS parameter.

STATUS is given either by Rule 103 or by Rule 104, since the IFF is now ON. IFFRET is a sensor parameter that tells whether the respective TRACK answered the IFF inquiry, if "yes" the system will assign a "friend" STATUS for the TRACK and Rule 105 will assign the same status for TRACK3 (wingman). However, in order to gain more insight into the behavior of the system, we will assume that no return has been received so TRACK1 is assigned as "foe". As a consequence, Rule 106 makes the same assignment for TRACK3. The same chain of rules will assign TRACK2's STATUS also as "foe".

The "foe" assignment for TRACK1 denies the first premise of Rule 053, so MONITOR will look for the next rule that states about THREATLV and have successful premises. In its search, MONITOR winds up in Rule 107, which first premise asks for ACFTTYPE. Again, the program will be led to the emitter's rules, where Rule 501 forces FINDOUT to return that TRACK1's emissions are from an EM002 emitter.

The same line of thought used in Rule 501's assessment applies to Rule 500. That is no STATUS will be automatically provided, even considering that the probabilities are more consistent for the "enemy X fighter" hypothesis then they were in Rule 501's "enemy Y fighter" hypothesis, since an "foe" assumption would lead to an eventual "fratricide" fire[17] by the system. Again, other evidences will make STATUS assessment and in that particular case this assessment was already made (TRACK1 and TRACK3 are "foe").

As in the "Y fighter" case, "X fighters" THREATLV assignment is made possible by the many rules that can be found in the aircraft context list. From those, Rule 125 is the one that will have all premises passed for TRACK1, thus the assigned THREATLV will be "4" and MONITOR will then proceed with the assessment of TRACK2.

[17] "Fratricide" fire, or firing at a friend is one of the most concerning issues in the developing of automated systems. Not only because the eventual casualties or equipment losses, but mainly because of the extreme damage it produces in terms of morale of one's own forces.

TRACK2 is now at 47 nm and with the same parameters as when the second snapshot was taken but with a “foe” STATUS. Therefore, a THREATLV of “1” will be assigned, indicating that TRACK2 is not subject of imminent danger for the system.

The last step for satisfying the third premise of rule 001 is the assessment of TRACK3’s THREATLV. This process, which is rather similar to the TRACK1 assignment, will finish with Rule 126 returning a THREATLV of “4”. Now all TRACK context nodes already have their THREATLV assigned and MONITOR follows on to the action statements of Rule 001.

This time, the first action of Rule 001 will trigger Rule 619 for TRACK1 and TRACK3. The rationale here is to launch a missile before loosing contact with a maneuvering interceptor (the free fighter), while using electronic counter measures (ECM) to prevent the engaged fighter (i.e. TRACK 3, the wingman) from firing its missile. At the same time, TRACK2 is kept under supervision by the system.

Solving the Second Scenario

Although the second scenario is rather different from the first one, the system will act in the same fashion. This means that it will attempt to apply its rules in order to determine the best reaction to be done and execute it in a timely way.

Initially, the first premise of Rule 001 will force the system to search for a TRACK context candidate. We will assume that FINDOUT indicated the presence of an emitter in point “A” of Figure 12. This will lead MONITOR to create a new TRACK context node (“TRACK1”) and a new EMITTER branch (“EMITTER1”) under that new TRACK context node. Since the detected emissions were from an emitter cataloged by the system as “EM009”, FINDOUT will assign this parameter to EMITTER1.

Rule 002 is valid for this scenario, so whenever the system detects a track, the first and second premises of Rule 001 will pass. This will speed up the process, leading MONITOR directly to the third premise, which uses Rule 055 in order to assess whether a specific track is an imminent danger. Since Rule 055 makes its judgment based on the THREATLV parameter, all TRACK context nodes will have it checked.

Again, Rule 053 is the first to state about THREATLV and it requires a definition of the STATUS parameter that is not given by any track rule, thus forcing MONITOR to go for the next available context level. Unlike the first scenario, where Rule 054 had to be used in order to establish the third level context nodes, this time this context level was already created (emitter), so the system is able to fire Rule 509.

That rule leads the system to the creation of a new context node (AAA2) while also defining its STATUS as “foe”'. After verifying Rule 053’s premise, MONITOR realizes that this and all other TRACK and EMITTER rules do not establish the

THREATLV parameter for non-friend tracks, so it goes directly to the next context available (the newly created AAA).

The sequence between Rule 202 and Rule 213 is dedicated to the THREATLV assessment for the “AAA1” and “AAA2” AAATYPE. There, the program uses some parameters in order to estimate how dangerous to the system’s aircraft these AAA may be. MODE parameter indicates the status of the AAA weapon system, which gives an indication whether or not it is able to shoot at the aircraft. EMTSS correlates the strength of the received signal with the distance from the emitter, which is a coarse estimation on how likely is the aircraft to be under the AAA lethal range. Finally, the effective altitude of the AAA is a known parameter, so the program compares it with its own altitude in order to realize whether the aircraft is inside the AAA’s vertical range. After considering all these parameters, Rule 208 will return a THREATLV assignment of “3” for TRACK1 and MONITOR will assess TRACK2's THREATLV.

The sequence “Rule 055 ⇒ Rule 053 ⇒ Rule 504 ⇒ Rule 400” will assign a THREATLV of “2” for TRACK2. Although an acquisition radar site provides no danger to the aircraft by itself, the assigned number is related to the threats that may be associated with this radar.

TRACK 3 assessment follows the same pattern of the previous two TRACK nodes. After passing through Rule 055, Rule 053 and Rule 507, MONITOR will use Rule

335 to assign a THREATLV of “3” for TRACK3 and will proceed with TRACK4's assessment.

TRACK4 is a very dangerous THREAT, employing sophisticated electronics for accuracy, multiple WEAPONGD phases for jamming resistance and backup modes for reliability. Nevertheless, its assessment also follows the same pattern of the previous two TRACK nodes, and the sequence “Rule 055 ⇒ Rule 053 ⇒ Rule 506 ⇒ Rule 321” will assign a THREATLV of “3” for TRACK4.

All 4 TRACK context nodes had their respective THREATLV assigned, thus satisfying the third premise of Rule 001. Since no THREATLV greater than 3 is found (refer to Rule 055), Rule 001's first action statement does not hold and MONITOR proceeds with the second action statement.

In order to execute the best actions for decreasing all THREATLV parameter, MONITOR attempts to apply the action rules that have their respective premises valid. In this particular case, Rule 622 forces the system to deviate from TRACK4.

It is interesting to note an “ad hoc” characteristic of this system that may be implemented in a different way in a real case system. The output of the backward chaining process produce a discrete value for THREATLV; in other words, a track can not have “3.5” as a valid parameter value. Instead, it could have either a “3” or a “4” as THREATLV. Because of the uncertainty related to these judgements and the nature of the actions to be taken, I would prefer to have an output that allowed a prioritization among the threats so I could separate an “almost 4” from a “barely 3”. To compensate this issue, “Wise Pilot 1” takes the CF value of each hypothesis to make a prioritization process when needed, as it is in the case of Rule 622. There, we could perceive three tracks with THREATLV of “3”, all eligible for the action stated in that rule. However, “Wise Pilot 1” considers TRACK4 hypothesis as more likely (CF of .9) so it applies Rule 622 to that track, instead of applying it to TRACK1, TRACK3, or to all of them (which would be hard to implement).

After applying Rule 622, MONITOR goes on and some ITERATION will occur between the first and the second snapshots. For comparison purposes, I will assume that no deviation was made (in spite of the last iteration’s result) and no further action was triggered in the thirty seconds period before the beginning of second snapshot's ITERATION.

As we can see in Figure 13, the situation has evolved to a higher risk scenario for Blue aircraft. However, one of the advantages of an automated system is its capacity of keep tracking of the changes in real time while providing a constant feedback to the pilot. This feedback allows him to be aware of the risk situation without preventing him to pay attention to other important issues (i.e. navigation, cockpit management, fuel control, etc.), so the increase of risk on this ITERATION does not constitute a dramatic and unexpected change in the evolving scenario.

There are two really unexpected changes in this snapshot. The first is the detection of a twin gun system associated with the ACQ radar, and the second is the MAW's report of a missile launching from a non-tracked threat. Therefore, the program will create two new TRACK nodes (TRACK5 and TRACK 6).

Again, the first two premises of Rule 001 are skipped and MONITOR will assess the THREATLV of all TRACK nodes (premise 3). TRACK1 has passed from sector 3 to sector 5 and, consequently, is now behind the aircraft's flight path. In addition, its signal strength has decreased. Since its radar is still in acquisition mode, Rule 210 will assign a THREATLV of “2” to TRACK1.

TRACK2 did not change, and the same sequence of rules used in the first snapshot's iteration will assign a THREATLV of “2”. TRACK3 now has a stronger signal and it is in tracking mode. The increase in its threat status can be perceived intuitively through the new parameters and is confirmed by the system, which will use Rule 328 to increase the THREATLV to “4”. Like TRACK2, TRACK4 did not change the parameters that were used by Rule 321, so the same THREATLV of “3” will be assigned to it.

TRACK5 was created because the aircraft's sensors detected a “EM005” emission, which is related with the twin gun system also found in TRACK1. Therefore, MONITOR will follow the sequence “Rule 055 ⇒ Rule 053 ⇒ Rule 505 ⇒ Rule 310” to assign a THREATLV of “3” to TRACK5.

TRACK6 is the next to be assessed. As it is expected for extremely dangerous situations (i.e. a missile has been launched). The system reserves two “first-to-be-checked” rules for checking whether a TRACK is a launched SAM. Rule 050 is the first track context rule to be visited, and its rationale is based on the fact that a missile launch detected inside the enemy territory with no previous knowledge about its existence has a strong likelihood of being a portable IR missile. Rule 052 uses the same rationale, but since the detected launching is not inside enemy territory, the rule assigns a “unknown” value for SAMTYPE.

Although both assignments will receive maximum THREATLV score (Rule 300), a slight difference will be evident in the action rules. A “portable” launched SAM will trigger an immediate action whenever the AI system is fully activated, while a “unknown” launched SAM will trigger the same action only if the autopilot system is fully engaged.

In short, the “portable” is treated like an “almost certain” threat, while the former will rely on the pilot's judgment on other evidences (i.e. smoke from the sector of launching, hints from wingmen, etc). In addition, the system considers that a fully engaged autopilot system means that the pilot is not available for a fast decision (or that he wants the system to act anyway).

Going back to MONITOR's work, the sequence "Rule 050 $\Rightarrow$ Rule 300" will assign a THREATLV of "5" to TRACK6. Since all TRACK nodes have been assessed, premise 3 of Rule 001 is then satisfied and MONITOR goes for the first action statement.

TRACK3, TRACK4, TRACK5, and TRACK6 are all considered as imminent danger, so first action statement of Rule 001 holds and MONITOR will go for the first applicable action rule concerning imminent danger (THREATLV greater than 3). In that particular case, the first rule that applies is Rule 606 and the aircraft will automatically perform its two actions.

The first action is to launch "FLARE1". This statement brings an implicit virtue of the system, the modularity. Flares are like incandescent balls ejected from the aircraft in a particular sequence, burning fast at a high temperature in order to mimic the jet's pipe and deceive the missile's infrared seeker.

However, some factors must be taken in account when launching flares in order to improve its efficacy. The aircraft's flare launcher is a programmable electronic device that has many pre-assorted sequences of launching, each to be used in very specific situations. A wrong sequence may prove inefficient or even act as a "homing path" for the missile.

In short, the type of IR seeker, the sector from it was launched, the aircraft's velocity, and other more detailed parameters are considered by the EW officer when establishing the most suitable sequence of launching (i.e. how many flares, the interval between then, etc) for each situation.

All pre-established sequences are loaded in a list which entries have the form "FLARE##", each relating to a specific situation. The system's modularity is evident by the fact that it allows its users to make modifications in the sequences just by changing the list, with no need for adjusting the code.

As we can see through this scenario, FLARE01 is the sequence that will be used when a portable missile is launched from sector 6 (rear sector). However, flare sequences are often valueless when executed alone; they must be combined with a high "G" maneuver in order to be effective. That is the purpose of the second action statement of Rule 606.

As it is in the flare case, each particular situation will have a most suitable maneuver, and the system will also keep them in a modifiable list. MNV6 combines a series of high "G" turns and throttle settings that will deny the pipe's infrared signature for the missile's seeker, which will be being seduced to another direction by the concurrent FLARE01 sequence.

A transcript of flare sequences and maneuvers involves fairly technical considerations that are beyond the scope of this work. However, there is a slight detail that is important to be taken in account: if a turn has to be made, there should be a preferred direction to execute this turn. This preferred direction is given by the parameter SAFLFSEC.

In order to execute action 2, MONITOR will activate FINDOUT to retrieve the safest lateral front sector. SAFLFSEC is assigned by a system's internal algorithm that checks which sector has less tracks with THREATLV of “5”. In case of draw, the tracks with THREATLV of “4” will make the difference. If there is still a draw, the safest sector will be defined by the CF-based prioritization I have mentioned before (page 194).

In “Wise Pilot 1”, maneuver actions are exclusive so only the first one is to be applied, which explains why the rules are sorted in accordance with its respective THREATLV. Because of that characteristic, no further actions will be made by the system and this ITERATION will finish with the execution of Rule 606’s action clauses. The aircraft will launch a preprogrammed sequence of flares and will execute a standardized maneuver to sector 2 (left side turn).

APPENDIX C:

DDN TECHNIQUE APPLIED TO THE SCENARIOS

For the purposes of this Thesis, the “Wise Pilot 2” DDN system was build with the help of computer software Netica™, a modeling tool that uses a belief updating algorithm described by Spiegelhalter (Spiegelhalter, 1993) and Neapolitan (Neapolitan, 1990).

Pictures in this appendix are Netica™'s screens of the actual system, implemented in accordance with Chapter 5. The complete structures were too big for being displayed in just one page without sacrificing the resolution. In order to assure a complete understanding of the system, figures 34, 36, 38, 40, and 41 were designed only for providing a “complete view” of the entire structure[18]. To provide a more detailed view, figures 35, 37, 39, 41, and 43 show the nuances of the influence diagram part of the system, which corresponds to the complete structure after having the Altitude node and all BN absorbed.

[18] In those views, the Altitude node was already absorbed for improving the clarity of the picture.

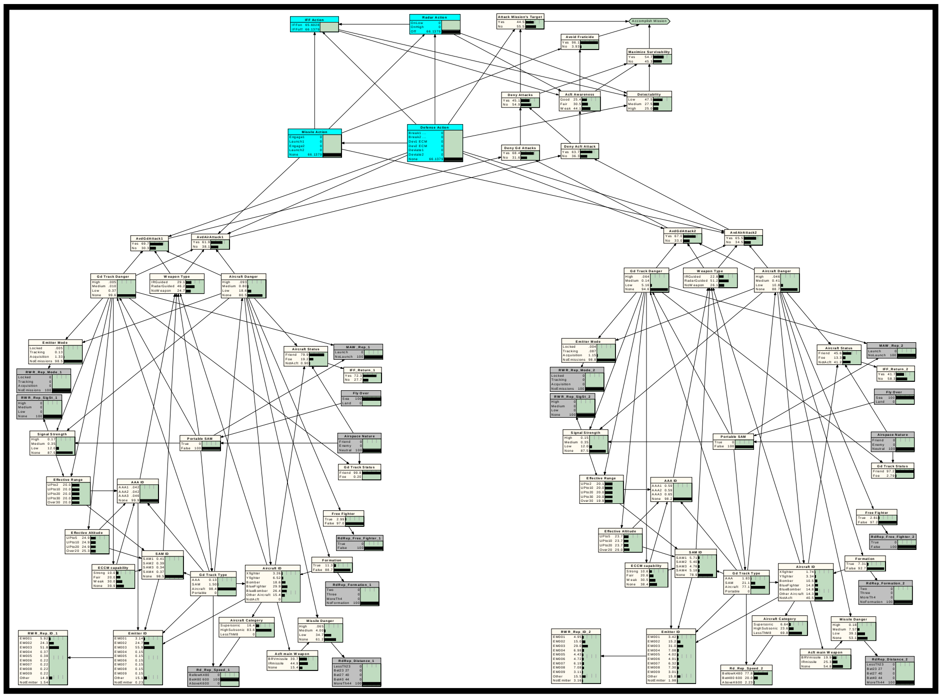


Figure 34 DDN System for 1st Scenario, First Snapshot

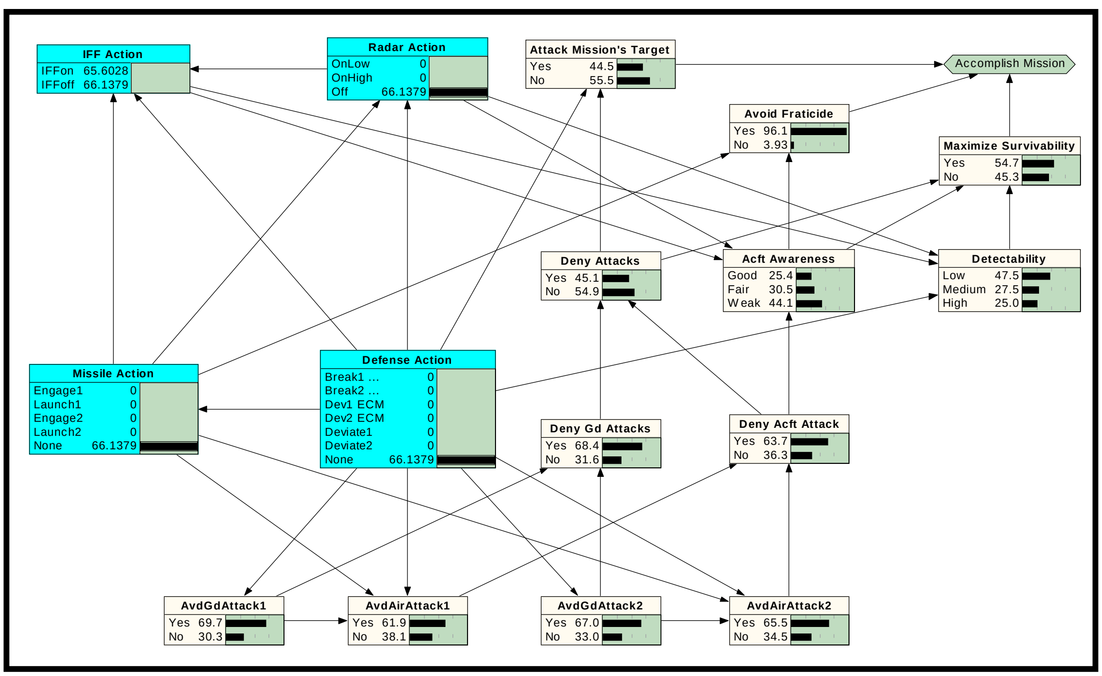


Figure 35 Influence Diagram of the DDN System for 1st Scenario, First Snapshot

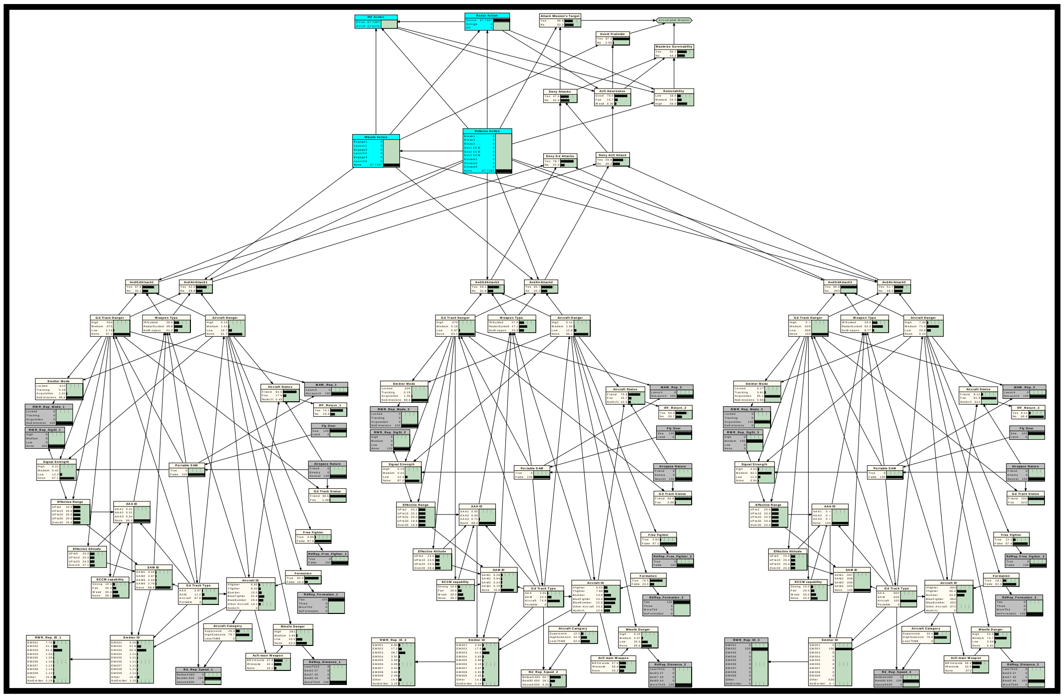


Figure 36 DDN System for 1st Scenario, Second Snapshot

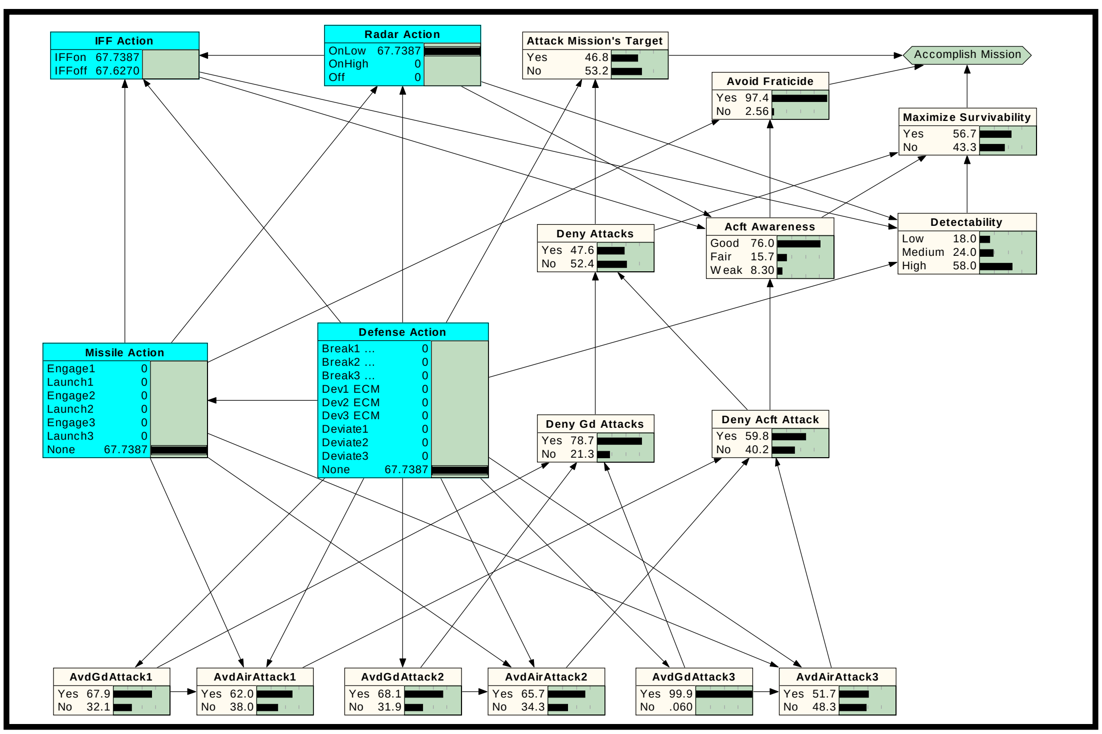


Figure 37 Influence Diagram of the DDN System for 1st Scenario, Second Snapshot

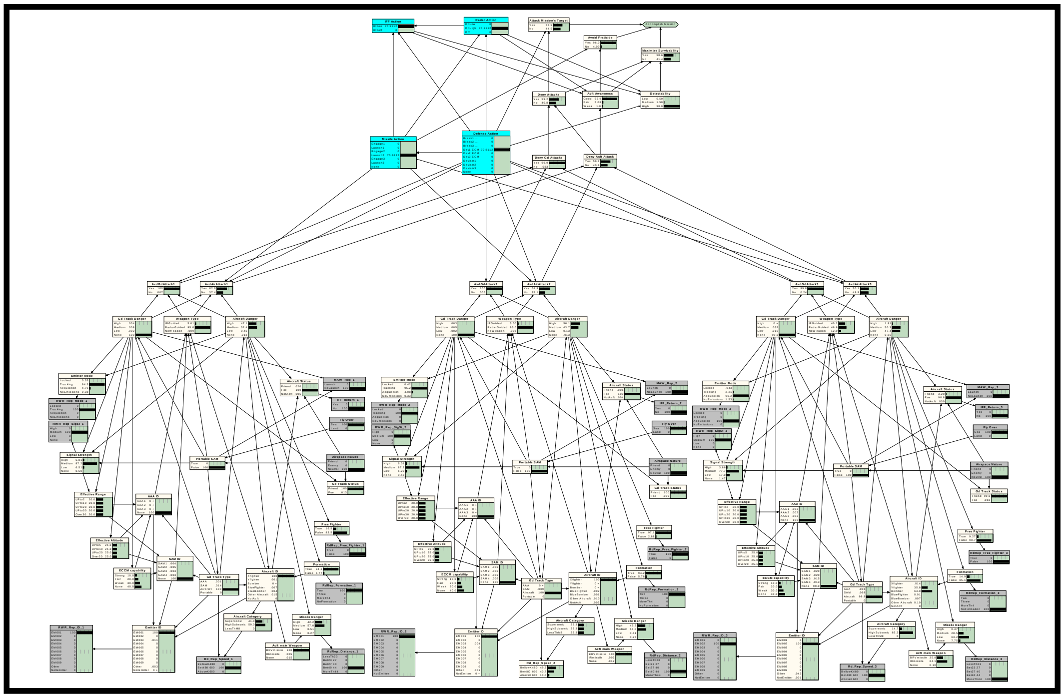


Figure 38 DDN System for 1st Scenario, Third Snapshot

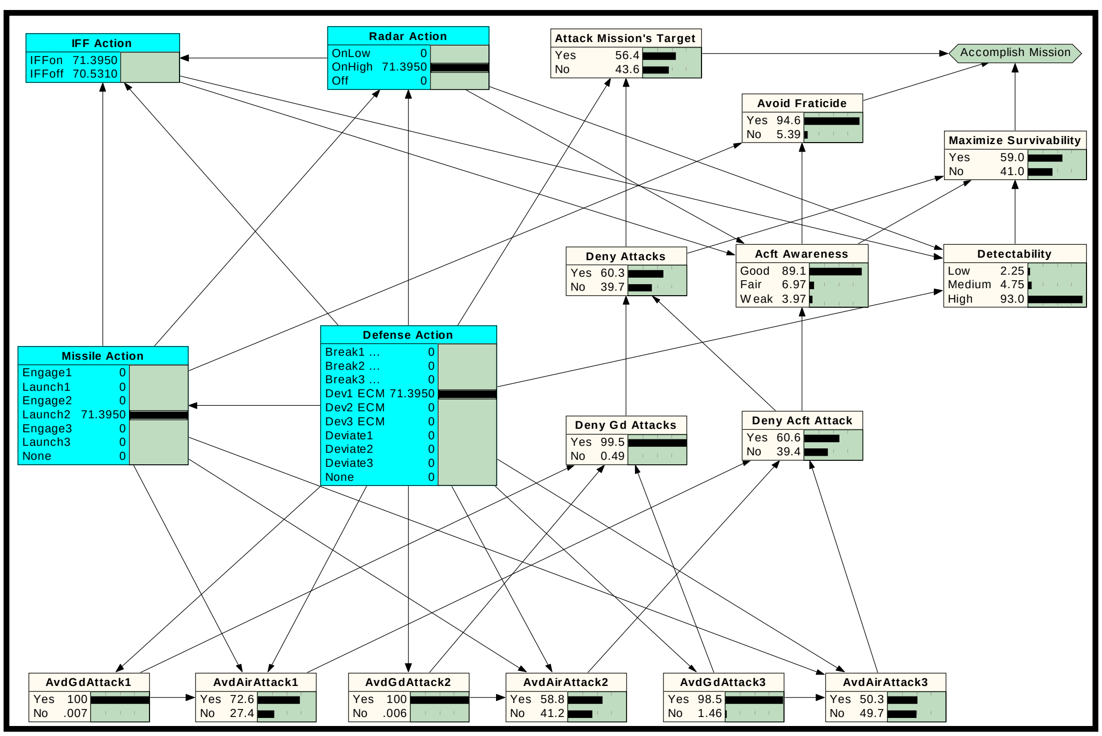


Figure 39 Influence Diagram of the DDN System for 1st Scenario, Third Snapshot

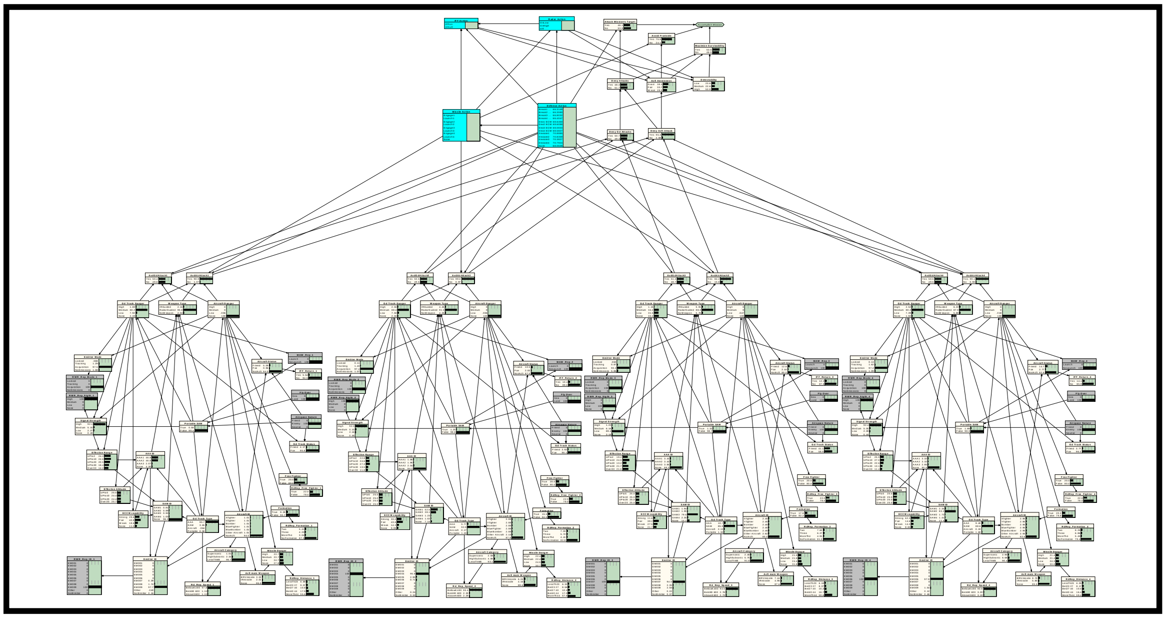


Figure 40 DDN System for 2nd Scenario, First Snapshot

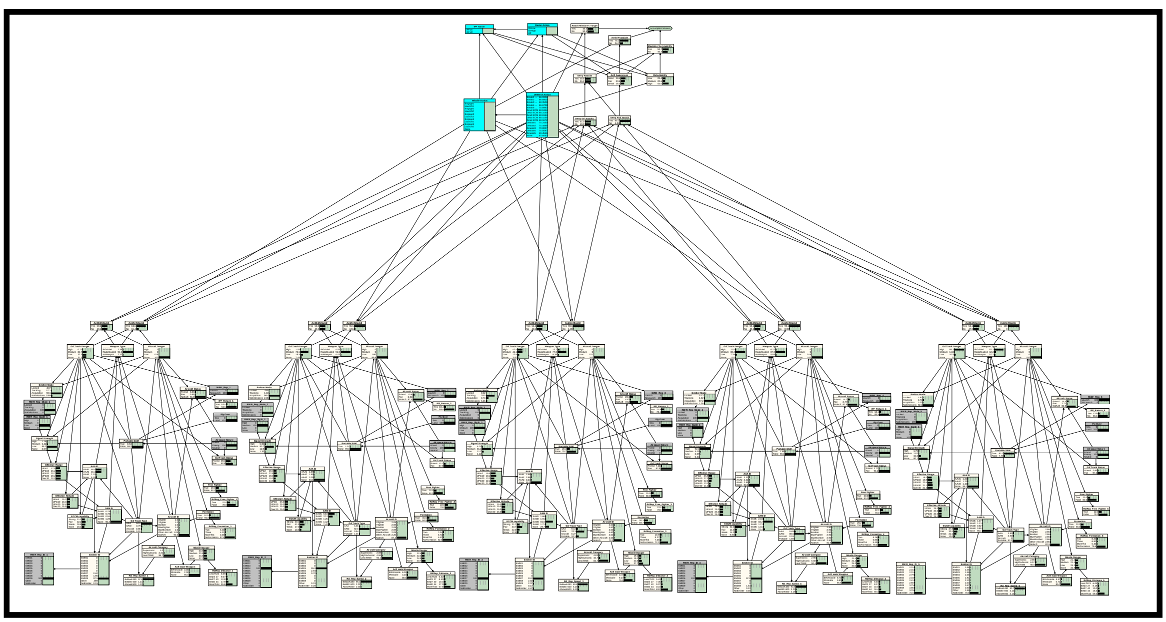


Figure 41 DDN System for 2nd Scenario, Second Snapshot

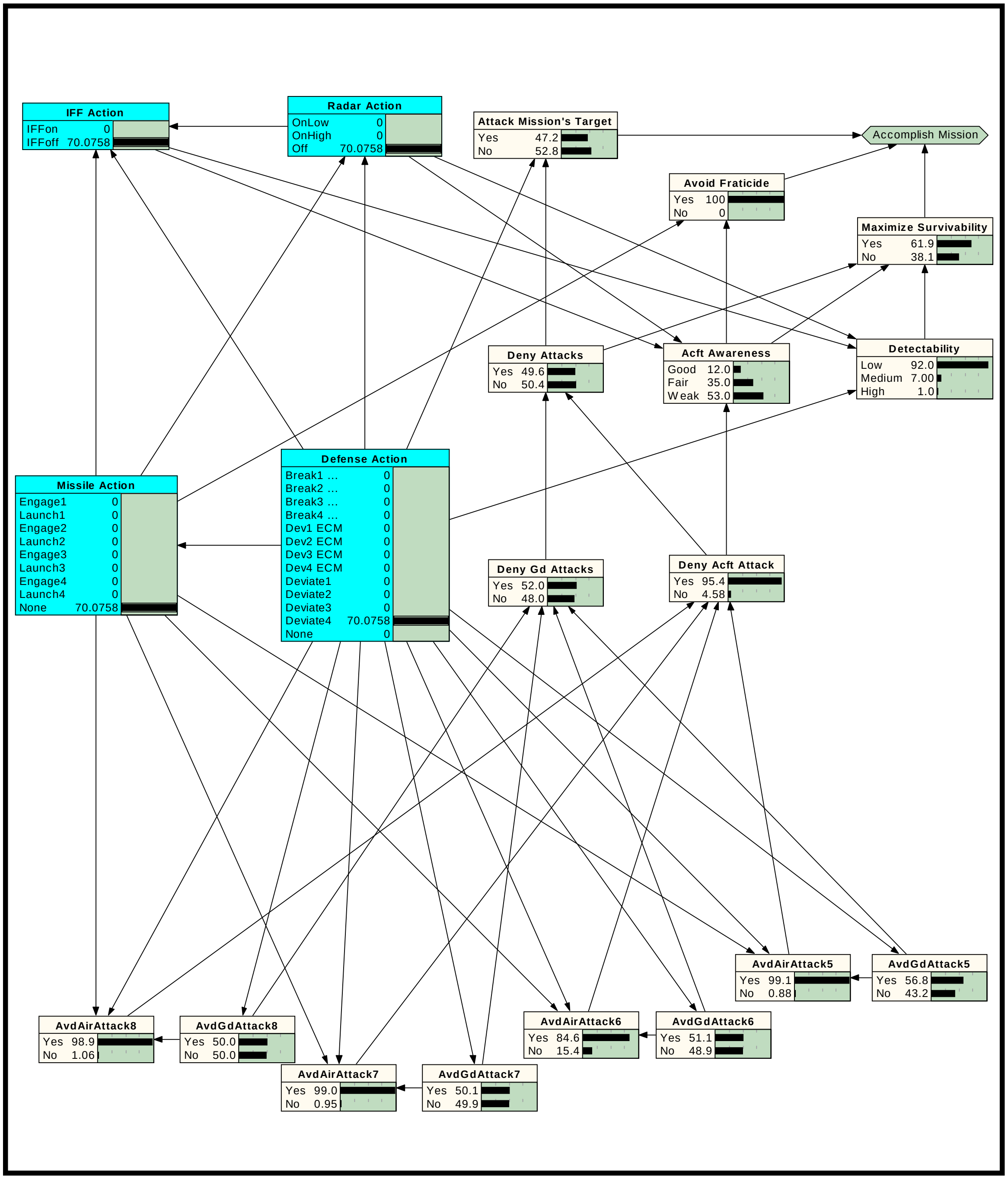


Figure 42 Influence Diagram of the DDN System for 2nd Scenario, First Snapshot

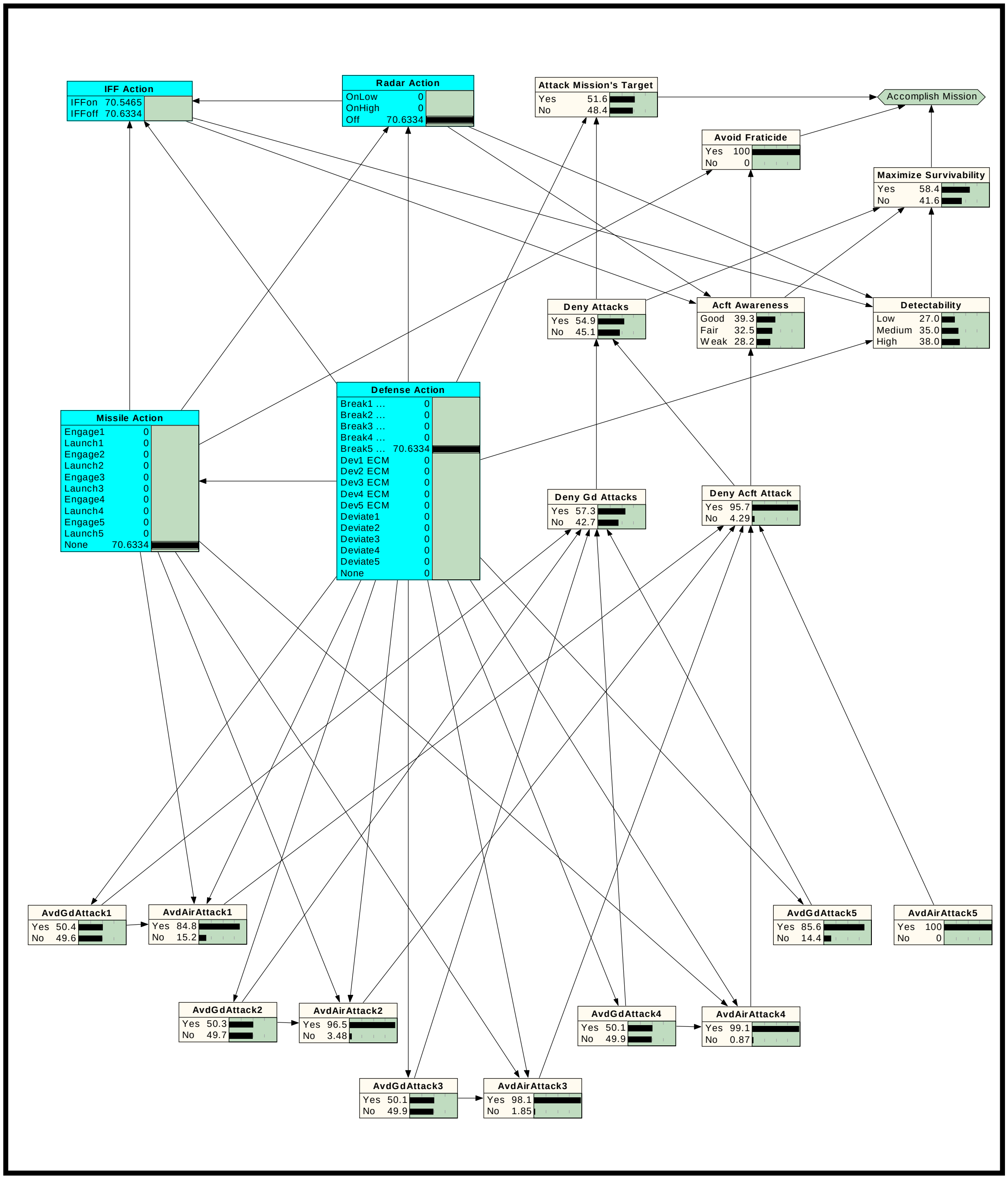


Figure 43 Influence Diagram of the DDN System for 2nd Scenario, Second Snapshot

BIBLIOGRAPHY

BIBLIOGRAPHY

Adams, J. B. (1976) “A Probability Model of Medical Reasoning and the MYCIN model”, in ***Mathematical Biosciences***, no. 32, 177-186.

Agogino, A. M., & Ramamurthi, K. (1990) “Real Time Influence Diagrams for Monitoring and Controlling Mechanical Systems”, chapter in ***Influence Diagrams, Belief Nets, and Decision Analysis***, John Wiley & Sons, Inc., Chichester, UK.

Balakrishnan, V. K. (1991) ***Introductory Discrete Mathematics***, Prentice Hall, Englewood Cliffs, NJ.

Ball, W.W.R. (1960) ***Short Account of the History of Mathematics***, Dover, New York, NY (originally published in 1908).

Baumol, F. H. (1972) ***Economic Theory and Operations Analysis***, Prentice Hall, Englewood Cliffs, NJ.

Bell, D. E., Keeney, R. L., & Raiffa, H. (1977) ***Conflicting Objectives in Decisions***, John Wiley & Sons, New York, NY.

Bellman, R. (1957) ***Dynamic Programming***, Princeton University Press, Princeton, NJ.

Blake, B. (1998) ***Jane's Radar and Electronic Warfare Systems***, 10^{th} Ed., August 1998, Jane's Information Group Ltd., Surrey, UK.

Booker, L. B. & Hota, N. (1988) “Probabilistic Reasoning About Ship Images”, in ***Uncertainty in Artificial Intelligence 2***, pp. 371-379, Elsevier Science Publishing Company, Inc., New York, NY.

Buchanan, B. G. & Shortliffe, E. H. (1985) ***Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project***, Addison-Wesley Publishing Company, Inc., Reading, MA.

Buede, D. M. (1988) "Shafer-Dempster and Bayesian Reasoning: A Response to Shafer-Dempster Reasoning with Applications to Multisensor Target Identification Systems", in ***IEEE Transactions on Systems, Man, and Cybernetics***, November/December 1988, Vol. SMC-18, no. 6, 1009-1011.

Buede, D. M., & Girardi, P. (1997) "A Target Identification Comparison of Bayesian and Dempster-Shafer Multisensor Fusion", in ***IEEE Transactions on Systems, Man, and Cybernetics***, September 1997, Vol. 27 Part A, No. 5, pp. 569-577.

Buede, D. M. (1999) ***Dynamic Decision Networks: An Approach for Solving the Dual Control Problem***, at 1999 Spring INFORMS, Cincinnati, OH.

Buede, D. M. (Forthcoming) ***The Engineering Design of Systems: Models and Methods***, John Wiley & Sons, Inc., New York, NY.

Carnap, R. (1950) "The Two Concepts of Probability", in ***Logical Foundations of Probability***, University of Chicago Press, Chicago, IL.

Charniak, E. (1991) "Bayesian Networks without Fears", in ***AI Magazine***, Winter 1991, 12(4), 50-63.

Clemen, R. T. (1996) ***Making Hard Decisions: An Introduction to Decision Analysis***, Duxbury Press, Belmont, CA.

Cohen, L. J. (1989) ***An Introduction to the Philosophy of Induction and Probability***, Clarendon, Oxford, UK.

Corner, J. L. & Kirkwood, C. W. (1991) "Decision Analysis Applications in the Operations Research Literature, 1970-1989", in ***Operations Research***, 39, 206-219.

Cullen, T. (1998) ***Jane's Land Based Air-Defense 1998-99***, 11th Ed., May 1998, Jane's Information Group Ltd., Surrey, UK.

De Finetti, B. (1972) ***Probability, Induction, and Statistics: The Art of Guessing***, John Wiley & Sons, New York, NY.

Fine, T. L. (1973) ***Theories of Probability***, Academic Press, New York, NY.

Foss, Christopher (1998) ***Jane's Armour & Artillery 1998-99***, 19th Ed., August 1998, Jane's Information Group Ltd., Surrey, UK.

French, S. (1986) ***Decision Theory: An Introduction to the Mathematics of Rationality***, John Wiley & Sons, Chichester, UK.

Gelman, A., Carlin, J. B., Stern, H. S., Rubin, D. B. (1995) ***Bayesian Data Analysis***, Chapman & Hall / CRC press, Boca Raton, FL.

Hacking, I. (1975) ***The Emergence of Probability: A Philosophical Study of Early Ideas about Probability, Induction, and Statistical Inference***, Cambridge University Press, Cambridge, MA.

Haiffa, H. (1982) ***The Art and Science of Negotiation***, Harvard University Press, Cambridge, MA.

Hansson, O., & Mayer, A. (1989) “Heuristic Search as Evidential Reasoning”, in ***Proceedings of the Fifth Workshop on Uncertainty in Artificial Intelligence***, 152-161, Mountain View, CA.

Harmon, P. & King, D. (1985) ***Expert Systems: Artificial Intelligence in Business***, John Wiley & Sons, Inc., New York, NY.

Hayes-Roth, F., Waterman, D. A., & Lenat, D. B. (1983) ***Building Expert Systems***, Addison-Wesley Publishing Company, Inc., Reading, MA.

Henrion, M., Breese, J., & Horvitz, E. J. (1991) “Decision Analysis and Expert Systems”, in ***AI Magazine***, Winter 1991, 12(4), 64-91.

Hilton, R. W. (1981) “The Determinants of Information Value: Synthesizing some General Results”, in ***Management Science***, Vol. 27, no. 1, January 1981, 57-64.

Howard, R. A. & Matheson, J. E. (1981) “Influence Diagrams”, in ***The Principles and Applications of Decision Analysis***, Vol. II, Strategic Decisions Group, Menlo Park, CA.

Howard, R. A. (1989) “Knowledge Maps”, in ***Management Science***, Vol. 35, no. 8, August 1989, 903-922.

Jackson, P., Taylor, J.W.R., Peacock, L., & Munson, K. (1997) ***Jane's All the World Aircraft 1997-98***, 88th Ed., October 1997, Jane's Information Group Ltd., Surrey, UK.

Jensen, F. V. (1996) ***An Introduction to Bayesian Networks***, Springer, New York, NY.

Keeney, R. L., & Haiffa, H. (1976) ***Decisions with Multiple Objectives***, John Wiley & Sons, Inc., New York, NY.

Keeney, R. L. (1980) ***Siting Energy Facilities***, Academic Press, New York, NY.

Keeney, R. L. (1992) ***Value-Focused Thinking: A Path to Creative Decisionmaking***, Harvard University Press, Cambridge, MA.

Kenley, C. R., & Casaletto, T. R. (1990) “Multi-Target Tracking Using Influence Diagram Models”, chapter in ***Influence Diagrams, Belief Nets, and Decision Analysis***, John Wiley & Sons, Inc., Chichester, UK.

Keynes, J. M. (1957) ***A Treatise on Probability***, Macmillan, London, UK (originally published in 1921).

Kirkwood, C. W. (1996) ***Strategic Decision Making: Multiobjective Decision Analysis with Spreadsheets***, Duxbury Press, Belmont, CA.

Kolmogorov, A. M. (1959) ***Foundations of the Theory of Probability***, Chelsea Publishing, New York, NY (originally published in 1933).

Laplace, P. S. (1951) ***A Philosophical Essay and Probabilities***, Dover, New York, NY (originally published in 1820).

Lee, P. M. (1992) ***Bayesian Statistics: An Introduction***, Oxford University Press, New York, NY.

Lipschutz, S. (1976) ***Schaum's Outline of Theory and Problems of Discrete Mathematics***, Schaum Outline Series, McGraw-Hill Inc., New York, NY.

Lovejoy, W. S. (1991) "A Survey of Algorithmic Methods for Partially Observed Markov Decision Processes", in ***Annals of Operations Research***, 28(1): 47-65.

Lucas, P. & Van Der Gaag, L. (1990) ***Principles of Expert Systems***, Addison-Wesley Publishing Company, Inc., Reading, MA.

Matheson, J. E. (1990) “Using Influence Diagrams to Value Information and Control”, chapter in ***Influence Diagrams, Belief Nets, and Decision Analysis***, John Wiley & Sons, Inc., Chichester, UK.

Maxwell, D. T. (1994) ***Composing and Constructing Value Focused Influence Diagrams: A Specification for Decision Model Formulation***, PhD Dissertation, IT&E, George Mason University, 1994.

Miller, A. C., Merkhofer, M. M., Howard, R. A., Matheson, J. E., & Rice, T. R. (1976) ***Development of Automated Aids for Decision Analysis***, Stanford Research Institute, Menlo Park, CA.

Morgan, M.G. & Henrion, M. (1990) ***Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis***, Cambridge University Press, New York, NY.

Neapolitan, R. E. (1990) ***Probabilistic Reasoning in Expert Systems: Theory and Algorithms***, John Wiley & Sons, Inc., New York, NY.

Oliver, R. M., & Smith, J. Q. (1990) ***Influence Diagrams, Belief Nets, and Decision Analysis***, John Wiley & Sons, Inc., Chichester, UK.

Olmsted, S. M. (1983) ***On Representing and Solving Decision Problems***, Ph. D. thesis, EES Department, Stanford University, Stanford, CA.

Pearl, J. (1988) ***Probabilistic Reasoning in Expert Systems: Networks of Plausible Inference***, Morgan Kaufmann Publishers, Inc., San Francisco, CA.

Page, L. B. (1988) ***Probability for Engineering with Applications to Reliability***, Computer Science Press, New York, NY.

Press, S. J. (1989) ***Bayesian Statistics: Principles, Models and Applications***, John Wiley & Sons, Inc., New York, NY.

Quinlan, J. R. (1987) Applications of Expert Systems: Based on the Proceedings of the Second Australian Conference, Addison-Wesley Publishing Co., Inc., Wokingham, U.K.

Ramsey, F. P. (1931) ***The Foundations of Mathematics and Other Logical Essays***, Kegan Paul, London, UK.

Savage, L. J. (1954) ***Foundations of Statistics***, John Wiley & Sons, New York, NY.

Savage, L. J. (1961) "The Foundations of Statistics Reconsidered", in ***Proceedings of the Fourth Berkeley Symposium on Mathematics and Probability***, reprinted in ***Readings in Uncertain Reasoning***, Shafer & Pearl Ed., Morgan Kaufmann, Palo Alto, CA.

Schum, D. A. (1994) ***Evidential Foundations of Probabilistic Reasoning***, John Wiley & Sons, Inc., New York, NY.

Shachter, R. D. (1986) "Evaluating Influence Diagrams", in ***Operations Research***, Vol. 34, 871-882.

Shachter, R. D. (1988) "Probabilistic Inference and Influence Diagrams", in ***Operations Research***, Vol. 36, 589-604.

Shachter, R. D. (1990) "An Ordered Examination of Influence Diagrams", in ***Networks***, Vol. 20, 535-563.

Shafer, G. (1976) ***A Mathematical Theory of Evidence***, Princeton University Press, Princeton, NJ.

Sox, H. C., Blatt, M. A., Higgins, M. C., & Marton, K. I. (1988) ***Medical Decision Making***, Butterworths, Boston, MA.

Spiegelhalter, D. J, Franklin, R., Bull, K. (1989)"Assessment Criticism and Improvement of Imprecise Probabilities for a Medical Expert System", in ***Proceedings of the Fifth Workshop on Uncertainty in Artificial Intelligence***, 335-342, Mountain View, CA.

Spiegelhalter, D. J., Dawid, A. P., Lauritzen, S. L., & Cowell, R. G., (1993) "Bayesian Analysis in Expert Systems", in ***Statistical Science***, 8(3), 219-283.

Von Newmann, J. & Morgenstern, O. (1944) ***The Theory of Games and Economic Behavior***, Princeton University Press, Princeton, NJ.

Von Mises, R. (1957) ***Probability, Statistics, and Truth***, George Allen & Unwin, London, UK (originally published in 1928).

Walker, T. C. & Miller, R. K. (1989) ***Expert Systems Handbook***, The Fairmont Press, Inc., Lilburn, GA.

Watson, S. R. & Buede, D. M. (1987) ***Decision Synthesis***, Cambridge, New York, NY.

Weatherford, R. (1982) ***Philosophical Foundations of Probability Theory***, Routledge & Kegan Paul, London, UK.

Welmann, M. P., & Heckerman, D. (1995) "Real-World Applications of Bayesian Networks", in ***Communications of the ACM***, March 1995, vol. 8, no. 5, pp. 24-30.

Winston, P. H. (1986) "The Commercial Debut of Artificial Intelligence", in ***Applications of Expert Systems***, Quinlan, J. R. ed., Addison-Wesley Publishing Company, Inc., Reading, MA.

Winston, W. L. (1993) ***Operations Research: Applications and Algorithms***, Duxbury Press, Belmont, CA.

Ziemba, W. T., & Butterworth, J. E. (1975) "Bounds on the Value of Information in Uncertain Decision Problems", in ***Stochastics***, Vol. 1, 361-378.

CURRICULUM VITAE

Paulo Cesar Guerreiro da Costa was born on March 03, 1965 in Rio de Janeiro, Brazil, and is a Brazilian citizen. He graduated from Escola Preparatória de Cadetes do Ar, Barbacena, Brazil, in 1982, and received his Bachelor of Science degree from Academia da Força Aérea, Pirassununga, Brazil, in 1986. Captain Paulo Cesar has more than ten years of experience flying fighter aircraft for the Brazilian Air Force, having reached the operational status of Fighter Wing Leader in AMX fighters. He also has served as an electronic warfare officer during the last four years prior to his arrival at the United States, in 1997. Until the beginning of 1999, he has been a full time student at the Center of Excellence in Command, Control, Communications and Intelligence for his M.S. degree in systems engineering. Currently, he is a Ph.D. student at George Mason University.