3-2000

A Method of Focusing the Attention of the Decision-Maker on Uncertain Information

John F. Moesner IV

Recommended Citation

Moesner, John F. IV, "A Method of Focusing the Attention of the Decision-Maker on Uncertain Information" (2000). *Theses and Dissertations*. 4832.
https://scholar.afit.edu/etd/4832

A METHOD OF FOCUSING THE ATTENTION OF THE DECISION-MAKER ON UNCERTAIN INFORMATION

THESIS

John F. Moesner IV, Captain, USAF

AFIT/GCS/ENG/00M-17

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GCS/ENG/00M-17

A METHOD OF FOCUSING THE ATTENTION OF THE DECISION-MAKER ON UNCERTAIN INFORMATION

THESIS

Presented to the Faculty
Department of Electrical and Computer Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
In Partial Fulfillment of the Requirements for the
Degree of Master of Science in Computer Systems

John F. Moesner IV, B.S. Computer Systems
Captain, USAF

March 2000

Approved For Public Release; Distribution Unlimited

AFIT/GCS/ENG/00M-17

A METHOD OF FOCUSING THE ATTENTION OF THE DECISION-MAKER ON UNCERTAIN INFORMATION

John F. Moesner IV, B.S. Computer Systems
Captain, USAF

Approved:

Gregg H Gunsch — 3 MAR 00
Dr. Gregg H. Gunsch — Date
Committee Chair

Mark E. Oxley — 3 Mar 00
Dr. Mark E. Oxley — Date
Committee Member

Thomas F. Reid — 3 Mar 00
Maj Thomas F. Reid — Date
Committee Member

Acknowledgments

This research endeavor would not have been possible without the support and encouragement of a number of people. As a result, I would like to express my sincere gratitude to these individuals. Foremost, I would like to thank my thesis advisor, Dr. Gregg Gunsch. His persistence, vision, and motivation helped me to plow through the many instances of pure chaos and confusion. I thank Dr. Mark Oxley and Maj Thomas Reid for serving on my thesis committee and providing me with invaluable commentary and insight.

In addition, I would like to express my appreciation to my fellow classmates. To my classmates: Thank you for your support and friendship throughout our AFIT experience. In addition, I would like to thank Lt Phillip Polk for his outstanding efforts on the Web-based survey that he took the time to create for me.

Most importantly, I wish to thank my wife, Jennifer, and my children, Emily, Molly, and Jacob, for their love, understanding, and patience. I would never have endured this latest of life's challenges without their eternal support. I feel truly blessed to be part of their lives.

John F. Moesner IV

Table Of Contents

List of Tables

List of Figures

AFIT/GCS/ENG/00M-17

Abstract

Military decision-makers need to be able to synthesize large amounts of information quickly and make accurate and timely decisions. However, all too often, when a decision-maker is bombarded with important information that has an uncertainty associated with it, that information is often neglected. One method of dealing with this type of information overload is through proper data orientation. By reducing the clutter of irrelevant information, the proportions of useful and relevant data can be increased. At the same time, the attention of the decision-maker is directed to the critical tasks or centers of gravity.

The WATCHDOG Decision Support Tool is an attempt to develop such a data orientation method. This tool can aid the decision-maker in sorting through large amounts of uncertain information and guiding his or her focus to the appropriate areas of concern. Conceptually, this tool would be used in the "orient" phase of OODA (Observe, Orient, Decide, and Act) decision cycle. By expediting the "orient" phase, the overall decision process would also be accelerated. The decision-maker's situational awareness would be amplified and the quality and the timelines of a decision would be increased. This thesis effort resulted in the development of the WATCHDOG tool. Experimentation was performed on ten test volunteers with a common underlying experience. Results indicated that the tool proved to be useful in support of the decision-maker when sorting through complex data with high degrees of uncertainty.

A METHOD OF FOCUSING THE ATTENTION OF THE DECISION-MAKER ON UNCERTAIN INFORMATION

1 Introduction

"War is the province of uncertainty: three-fourths of those things upon which action in war must be calculated, are hidden more or less in the clouds of great uncertainty."

C. von Clausewitz, *On War*

1.1 Background

The primary responsibility of the military commander is to make the timely and accurate decisions that will win the battle. In order to do so, he must have all pertinent data about the battlefield environment at his immediate disposal. In the past, these commanders relied on a staff to provide this data. In addition, the staff was also responsible for providing potential Courses of Action (COAs). To ensure that the commander did not waste his time reviewing trivial items, the staff had a team of analysts. These analysts were responsible for aggregating all of the available information, analyzing it, and determining what was important and what was not. By doing so, only the most crucial data ever reached the battlefield commander. With the fat trimmed, the commander was able to quickly synthesize the details and come to the best decision.

Now, much of the information gathering process is automated. With the advent of computers and vast distributed networks, raw data about the battlespace can be gathered quicker than before. An unprecedented amount of intelligence data is available in near real-time, at the touch of a button. Now, a commander can have more battlefield data without the inherent delay previously associated with the human staff process. Now, the commander can receive much more information then he can possibly ever assimilate effectively.

This data on demand service has re-introduced an old but modernized dilemma: information overload. Simply put, too much information is available. A commander can become inundated with a flood of available data. Timely and critical decisions depend on the commander receiving only the most vital and essential pieces of the battlefield puzzle for quick analysis and synthesis. "In forming the mental model, subtle yet critical aspects of the battlespace may be missed, leading to incorrect decisions. Humans have limits of attention that may cause them to process cues that are not the most relevant [Sol91]." Moreover, data that is inherently uncertain can be gathered from the battlefield. Data collection will come from various sources from different functional areas that may contradict each other. Likewise, critical elements may be absent. A commander may need that crucial piece of information from which to base a decision. Furthermore, the reliability of said data may also be in question. A possible solution to this dilemma would be to provide an automated decision-making support tool that employs data orientation to manage the uncertainty. This tool would assist in determining and highlighting the most significant elements. It would sort through the vast amounts of information and pinpoint significant areas of interest. The goal of this research is to explore a decision support methodology that can guide the focus of the analyst to the appropriate areas of concern depending on a predetermined objective.

1.2 Problem Definition

The bane of today's information warrior is information overload, plain and simple. A common methodology that is able to collect raw data, organize it, analyze and assimilate it, and impart it in a form that enhances the military decision-makers' understanding of the battlespace is clearly needed. It is a well known fact that if a commander can improve his OODA (Observe, Orient, Decide, and Act) decision cycle, he can improve the quality and timeliness of his decisions. Ultimately, he can enhance his overall situational awareness. Automated decision support tools can be used to compress increasing volumes of data. Using these tools, information from various systems could be re-oriented in such a manner that provide the maximum amount of data that can be interpreted in the least amount of time with the greatest amount of ease. "Effective military and nonmilitary INFOSYS (Information Systems) help the staff get the right information to the right location in time to allow commanders to make quality decisions and take appropriate actions [FM100-6]." Furthermore, with proper visualization, we may be able to capitalize on the human brain's ability to parallel process enormous amounts of graphical information. Visualization and data orientation, synergized with human intuition, can magnify our efforts in finding those nodes that are common between several different systems. These "centers of gravity" can uncover potential vulnerabilities of an adversary's systems.

Decision-makers cannot fully exploit the information provided because of information overload. A new methodology is needed that can sort through inherently uncertain data, determine what is significant, reveal critical centers of gravity, and present that information in a way that is easily assimilated. This thesis will explore a method that will focus the attention of the decision-maker on uncertain information.

1.3 Objective and Hypotheses

The objective of this thesis is answer the following questions:

- Can a method that can focus the attention of the decision-maker on uncertain information be developed? What is the basic structure of such a method?
- Can a tool or program be created that implements the above method?
- As data becomes more complex, will the benefits of this tool become more apparent?

As result of these questions, a primary and a secondary hypothesis can be formulated. The primary hypothesis is as follows:

> A method that can focus the attention of a decision-maker on uncertain information can be developed.

The secondary hypothesis is as follows:

> As the data set becomes more complex,
> the benefit of the WATCHDOG tool will be greater.

1.4 Approach

The following tasks are designed to meet the proposed thesis objective. These steps represent significant segments of this thesis. They are not entirely distinct and are not independent of the other processes and considerations.

Define the Problem. Determine the nature of the uncertainties involved in decision-making. Assess general assumptions and scope of problem. Provide an initial approach for thesis direction. Provide preliminary thesis objective.

Background Research. Conduct a literature review of journals, publications, past research efforts, Internet sources, and current information tools available. Develop a preliminary taxonomy of methods that can manage uncertainty. Establish the capabilities of current research and tools. Outline absent or underdeveloped capabilities that would provide a logical progression from current research to future developments.

Define System Methodology. Discuss initial design considerations and motivations. Design a general framework for evaluating data orientation techniques. Further refine objectives and outline how to accomplish them. Formulate hypotheses. Set up criteria to validate system effectiveness.

Design and Implementation of the System. Elaborate on initial design consideration. Discuss further refinements for implementation. Establish specific methods, functions, and procedures for execution of data orientation techniques. Define a standard testing methodology to validate candidate system.

Conduct Testing & Analysis. Perform independent verification and validation of system. Gather examination data and assessment metrics. Aggregate test results from system. Resolve discrepancies in test results. Retest if necessary.

Conclude Findings. Scrutinize test results, recognize trade-offs, and establish conclusion. Validate or refute thesis hypotheses.

1.5 Scope

Bits of the decision-making methodology already exist. However, a crucial piece of this methodology is absent. What is missing is the correlation of the gathered raw, uncertain data into meaningful information and the presentation of that information in a manner that is easily assimilated. This corresponds to the "orient" phase of the OODA loop. In order to take full advantage of human parallel processing and automated computer processing, two facets of this orientation dilemma should be addressed. The first will involve applying information visualization to uncertainty, which will be addressed by Capt Evan Watkins [Wat00]. The second facet will be devising a technique of efficient reasoning and data orientation that can integrate data with inherent uncertainty and draw attention to centers of gravity. This thesis will focus on the latter facet of exploring methods that can be used in the development of an effective data orientation methodology that can be used as a decision support system for the military decision-maker.

Aside from the core problem of information overload, there is the problem of dealing with the many types of uncertainty. Initially, I identified several types of uncertainty that would be important to the military decision-maker. Identified were uncertainty due to contradicting or conflicting data, uncertainty due to absent critical data, uncertainty due to questionable sources, uncertainty due to risk, uncertainty due to age of data (staleness), and uncertainty due to data that has been sanitized due to a higher security classification. Ideally, all of these types of uncertainty should be addressed by a decision support tool. However, for the purposes of this research effort, a non-specific type of uncertainty will be addressed. Only uncertainty in general will be studied.

1.6 Assumptions

The following initial assumptions were made concerning this thesis:

- The decision support tool will need to be interactive.
- All domain data gathered will be uncertain and already in a format acceptable by the tool.
- Data orientation should be the prime focus of this thesis, not information visualization.
- This thesis will only focus on uncertainty in general, not on a particular type of uncertainty
- Course of action generation will be out of the scope of this thesis.

- The research is not restricted to any one particular system (e.g. DIODE, Dynamic Information Operations Decision Environment). However, a "toy" problem should be developed that is fundamentally simple yet can handle different domains [DIO99].

1.7 Thesis Organization

The remainder of this document presents a detailed description of my research, beginning in Chapter 2 with the background literature review. Chapter 3 provides insight into the methodology behind the creation of the decision support tool, specifically what I set out to do in this research and how I planned to accomplish it. Chapter 4 discusses the high-level design considerations as well as the finer implementation details of the coded tool. Chapter 5 presents the testing of the decision support tool including test considerations and procedures, the test results, and an analysis of the testing. Finally, in Chapter 6, I present my conclusion: the implementation of a decision support tool validates that a method can be developed that can guide the attention of a decision-maker to uncertain data.

2 Background Literature Review

"If we knew what we were doing, it wouldn't be called research, would it?"

Albert Einstein

2.1 Overview

This chapter presents the background knowledge needed to develop a method of focusing the attention of a decision-maker on uncertain information. The first section introduces uncertainty by defining the various types of uncertainty, the many causes of uncertainty, and the methods of dealing with uncertainty. The second section identifies the need for information systems that can support the decision-maker. The third section discusses decision-making models specifically the methods used to handle uncertainty in military decision-making including one that has been adopted by the Department of Defense. The fourth section discusses general decision-making concepts and basic decision analysis. Finally, the fifth section covers past research efforts in the realm of situational awareness assistance. Sources of background research include journals, publications, Internet sources, past research efforts in similar avenues of decision making and uncertainty, as well as various books on decision-making and decision analysis.

2.2 Understanding the Fog of War

The Prussian military theorist Carl von Clausewitz wrote about the smoke and confusion that takes part when any battle starts. This confusion or uncertainty during battle is called the "fog of war." Ever-present and inescapable, the fog of war is integral to the military decision-maker and is something that every good commander needs to take into account. Several sources were researched concerning the fog of war; however, the most revealing and inspiring sources are discussed below.

Russell & Norvig [RN95] attribute uncertainty to *incompleteness* or *incorrectness* of the environment. The authors stress that probability is the best way of handling uncertainty. In addition, there are four different mechanisms designed for reasoning with uncertainty: *default reasoning, rule-based, Dempster-Shafer,* and *fuzzy logic.* Smithson [Smi89] proposed a taxonomy of ignorance of which uncertainty was a contributor. Smithson suggested that *ignorance* is caused by two primary causes: *error* and *irrelevance. Uncertainty* is listed several levels down as a cause of incompleteness along with *absence. Incompleteness* and *distortion* are causes of error. Watkins [Wat00] further extends this taxonomy to include *unknowable* information and *omission* of information as a third and fourth cause of *ignorance.* Soltz [Sol93] discusses that the fog of war arises from *information inaccuracy* and *informational complexity. Information inaccuracy* is caused by uncertain data from any number of sources while *information complexity* is caused by a dynamic and complicated battlespace. Soltz also proposes that information complexity can be addressed by data reduction and analysis techniques. As you can see, just from these few sources, there are several opinions about the fog of war and uncertainty. There are many points of view just over the types of uncertainty let alone how to handle it. It stands to reason that a comprehensive taxonomy of uncertainty is outside the scope of this thesis. However, for the purposes of this research, the following points have been gleaned from the various background literature:

- All data in the battlespace is inherently uncertain

- Two causes of uncertainty are: Ignorance and Unreliability
- Two methods are available that can handle uncertainty: Information Visualization and Data Reduction/Orientation

The following paragraphs discuss the fog of war in further detail.

2.2.1 Artificial Intelligence: A Modern Approach [RN95]

Russell & Norvig state that uncertainty arises from *incompleteness* and/or *incorrectness* in an environment. They suggest several ways of handling uncertain knowledge. First, they suggest simple diagnosis using first-order logic to cope with a domain. However, this fails for three main reasons:

- *Laziness* – Too much work to list the complete set of antecedents and consequences.
- *Theoretical Ignorance* – Some domains, like medical science, have no complete theory for the domain.
- *Practical Ignorance* – May be uncertain because all necessary tests have not or cannot be run.

Second, probability can provide a way of summarizing the uncertainty that comes from our laziness and ignorance. Probability theory assigns a numerical degree of belief between 0 and 1 to each rule and provides a way of dealing with them. Third, by assigning a degree of truth to the rules, then we could handle the uncertainty using fuzzy logic.

The presence of uncertainty radically changes the manner in which we make decisions. *Preferences* can be used to choose between different possible outcomes. By comparing the utility, the quality of being useful, of the different outcomes, we can show which state or outcome will be a better or more useful outcome. This type of reasoning is called *utility theory*. If we combine probabilities to the utility of each state, we are employing decision theory to decide the better outcome. We would be rational if and only if we choose the action that yields the highest expected utility, averaged over all the possible outcomes of actions. This is called the principle *of Maximum Expected Utility (MEU).*

There are two types of probability statements: *unconditional* and *conditional.* Each statement can be distinguished by its dependence on experience. *Unconditional* or *prior probability* means that in the absence of any other evidence, the proposition will stand as indicated. For example, P(Sunny) = 0.7, means that in the absence of any other weather information, we assign a probability of 70% to the random variable "Sunny." Once we obtain some evidence concerning previously unknown propositions, we use *conditional* or *posterior probabilities.*

The authors elaborate that probability is the right way to reason about uncertainty. They summarize more of the basics of probability such as the *axioms of probability*, the *joint distribution function*, and *Bayes' rule*. The axioms of probability specify constraints on reasonable assignments of probabilities to propositions. A program that violates the axioms will behave irrationally in some circumstances. The *joint probability distribution* specifies the probability of each complete assignment of values to random variables. It is usually far too large to create or use. *Bayes' rule* allows unknown probabilities to be computed from known ones. In the general case, combining many pieces of evidence may require assessing a large number of conditional probabilities. Conditional independence brought about by direct causal relationships in the domain allows Bayesian updating to work effectively even with multiple pieces of evidence.

Other sciences (e.g., physics, genetics, economics) have long favored probability as a model for uncertainty. Pierre Laplace (1819) said, "Probability theory is nothing but common sense reduced to calculation." James Maxwell (1850) notes that "... the true logic for this world is the calculus of Probabilities, which takes account of the magnitude of the probability which is, or ought to be in a reasonable man's mind." Stephen J. Gould (1944) expressed that a "... misunderstanding of probability may be the greatest of all general impediments of scientific literacy."

One of the most well-studied qualitative uncertainty reasoning mechanism is *default reasoning*. *Default reasoning* treats conclusions not as "believed to a certain degree," but as "believed until a better reason is found to believe something else." Conclusions are reached by default unless some new evidence presents itself. Some default reasoning schemes are designed to handle reasoning with default rules and retraction of beliefs.

Rule-based methods for uncertainty reasoning have also been researched. These methods build on the success of logical rule-based systems, but add a "certainty factor" to each rule. This approach of was first used in developing the Mycin medical diagnosis expert system. In Mycin, certainty factors were a form of confidence in the diagnosis based on the symptoms, and could even be established statistically. Furthermore, from the use of certainty factors, a mathematical system for propagating the certainties and combining evidence from multiple sources was created.

Another approach to reasoning with uncertainty is the with *Dempster-Shafer* Theory. *Dempster-Shafer* is designed to deal with the distinction between uncertainty and ignorance. Instead of computing the probability of a proposition, it computes a measure of belief in the function. A common interpretation of *Dempster-Shafer* applies a confidence interval to a proposition.

The last reasoning with uncertainly technique established by the authors is the application of the *fuzzy set theory*. *Fuzzy sets* and *fuzzy logic* allow for a way to represent vagueness. *Fuzzy sets* do not have sharp boundaries and *fuzzy logic operations* account for functions with multiple memberships.

2.2.2 Ignorance and Uncertainty: Emerging Paradigms [Smi89]

Smithson takes a different spin than most with regard to uncertainty. He proposes Taxonomy of Ignorance that places uncertainty as a cause or contributor of ignorance. This is counter to other beliefs of how uncertainty can be broken apart. A diagram of Smithson's Taxonomy of Ignorance is provided in Figure 2-1. The following bullets provide brief descriptions of each entry of this taxonomy.

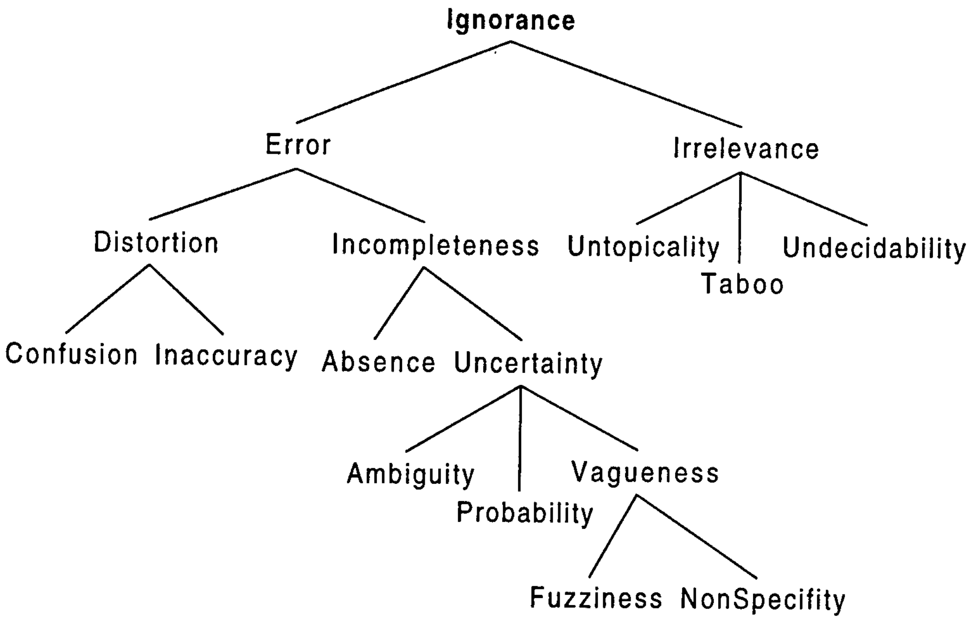


Figure 2-1: Taxonomy of Ignorance

- *Ignorance* - the basic lack of knowledge or information
- *Irrelevance* – having no current applicability hence might be intentionally ignored
- *Untopicality* – those issues that are not currently of interest and therefore appropriately not discussed.
- *Taboo* - socially inappropriate knowledge and activities as established by culture and value systems
- *Undecidability* – not being able reach a decision
- *Error* - incorrect views, information, or processes caused by being *incomplete* or *distorted*
- *Distortion – a* misrepresentation

- *Confusion* - indicates mistaken substitution
- *Inaccuracy* - results in a degree of distortion, degree of misperception or misunderstanding
- *Incompleteness* - state of not having all components
- *Absence* - information is simply missing
- *Uncertainty* - the specificity of the issue cannot be achieved; "one of the most manageable kinds of ignorance"
- *Vagueness* – unclear in form or expression
- *Probability* – involves chance
- *Ambiguous* - possesses more than one interpretation
- *Fuzziness* - the lack of clarity
- *Nonspecificity* - an inexplicit state of information

An interesting side issue, Watkins, in his pursuit of establishing a complete taxonomy of uncertainty [Wat00], adds a third and a fourth contributor of ignorance: *Unknowable* and *Omission.* Watkins points out that *unknowable* information is a significant contributor. Examples of unknowable information would include the outcome of tomorrow, someone else's thoughts, or the exact damage assessment prior to an attack – they are unknowable. However, *unknowable* does not imply something that is not known that could be learned such as something *taboo* or *untopical. Unknowable* is that information, foresight, or knowledge which is not possible to know. *Omission* pertains to the exclusion of some piece of information. This exclusion can be either the deliberate or accidental.

2.2.3 Graphical Tools for Situational Awareness Assistance for Large Battlespaces [Sol93]

In his masters thesis, *Graphical Tools for Situational Awareness Assistance for Large Battlespaces*, Soltz briefly discusses the fog of war, and as well a way of addressing this dilemma within a synthetic environment. In this section, we will discuss Soltz's perspective on the fog of war and its causes and address his actual thesis efforts in a later section.

As the battlespace has become more complex, large staffs and other management mechanisms have been created in order to assist commanders dealing with the fog of war and confusion inherent in any battlespace. However, this uncertainty will always be somewhat prevalent. Soltz states that confusion arises from two complementary problems: information accuracy and information complexity. Information accuracy can arise from deliberate enemy deception, observational error, conflicting data, and errors within the information gathering mechanisms. Information accuracy is a problem caused by uncertain data. Confusion caused by informational complexity can occur in any large battlespace that is constantly updating and complicated with numerous facets and characteristics. Because there are so many dynamic aspects, it becomes very difficult to ascertain every feature of the state. Soltz proposes that information complexity can be addressed using techniques for data reduction and analysis since it has already been proven in the field of computer science.

2.3 Identifying a Fundamental Need

This section covers identifying a fundamental need for a method of decision-making assistance. A brief article in *Forbes* highlights this need from results of Desert Storm [Ros97]. Desert Storm demonstrated that the use of precision weapons and superior intelligence gathering tools helped reduce the fog of war to such a minimal level that it almost was not a factor. The article recognizes a need for change because of the rapid pace of our technological edge. This section also highlights two other sources that recognize the fundamental need for a method.

2.3.1 Field Manual 100-6 : Information Operations [FM100-6]

Field Manual 100-6 is the U.S. Army bible on information operations. Here, the Army recognizes that INFOSYS (Information Systems) allow the commander to view and understand his battlespace, communicate his intent, lead his forces, and disseminate pertinent information throughout his chain of command. Importantly, effective military and nonmilitary INFOSYS help the staff get the right information to the right location in time to allow commanders to make quality decisions and take appropriate actions. This resonates the fundamental need for decision support tools.

Chapter 5 of FM 100-6 defines the role of INFOSYS to provide the infrastructure that allows the Army to interface with the GII (Government Information Infrastructure). INFOSYS form the architecture that

- Supports the staff process
- Supports the decision-making process
- Provides the relevant common picture that helps synchronize force application
- Links sensors, shooters, and commanders
- Supports C2-attack and C2-protect capabilities

The infrastructure of military and nonmilitary INFOSYS combine to provide the commander with a global reach capability. Chapter 5 also recognizes that as technology advances, information technology will--

- Help leaders form a more complete picture of the battlespace
- Generate the potential for faster, higher quality decisions
- Support more rapid maneuvers in terms of both time and space
- Increase a unit's flexibility and agility

Chapter 5 also acknowledges that while the fog of war has thinned, it will never completely disappear. The fog of war and uncertainty will be compounded by artful opponents (military or otherwise) and exacerbated by the consequences of unintentional actions from other sources within the unit.

Chapter 6 of FM 100-6 defines information dominance as a temporary tactical condition achievable through a deliberate process. Information dominance necessitates that the commander must seize the opportunity to gain the advantage through effective battle command. One of the two features essential to this process is Commander's Critical Information Requirement (CCIR). CCIR means that the commander must control information, or he runs the risk of being overwhelmed or disoriented by it. It can control the glut of information and separate the true signals from the noise. Similar to the Intelligence Preparation of the Battlefield (IPB), CCIR must be precise to ensure responsiveness and dynamic to survive. As you can see, CCIR also aligns with the fundamental need for a decision support tool.

Chapter 6 also comments on how Army operations are profoundly affected by information and IO in the critical function of battle command. Battle command relies increasingly on the ability to process information. It relies on the ability to move the information rapidly to critical points in the operational area. Here, again, we hear the familiar requirements of a decision support tool that can focus the attention of the user on critical points or watchspaces. Of the three basic elements of battle command (*leadership*, *decision-making*, and *controlling*), the element of *decision-making* is facilitated through much-improved information technologies.

2.3.2 Information Operations, Air Force 2025 [Osb96]

The Air Force 2025 study clearly identifies the need for a common methodology. This methodology needs to be able to collect raw data, organize it, analyze and assimilate it, and impart it in a form that enhances the military decision-makers' understanding of the battlespace. This has become the primary motivation behind this research.

To win the war in 2025, the U.S. Armed Forces will need an information operations architecture that provides timely, reliable, and relevant battlespace information to the battlefield commanders. This robust system will collect the raw data, organize it into usable information, perform preliminary analysis on it, and display this critical information in a form that enhances the military decision-maker's understanding of the situation. To win the war in 2025, the U.S. Armed Forces will need this system to make faster and more accurate decisions than their adversaries.

2.4 Decision Models

This section discusses two basic decision making models that are used in military decision-making: C3EVAL Model and the OODA Decision Cycle. The C3EVAL Model was a predecessor of the OODA Decision Cycle which is widely accepted by the decision-makers of the armed forces. The following sections will discuss each decision-making model in further detail.

2.4.1 Science of Command and Control: Coping with Uncertainty [GM88]

In the book, the *Science of Command and Control: Coping with Uncertainty*, AFCEA presents the fundamental problems of uncertain data associated with military decision-making in the battlespace. Here, the problem of uncertainty is identified, however a proper means of data presentation is not addressed. The following paragraphs discuss their major points.

Overcoming uncertainty on the battlefield is a function of command. Current combat modeling does not follow three non-Newtonian lessons of Clausewistz:

- Center of Gravity (CoGs) – the hub of all power and movement, on which everything depends (Clausewitz)

- Lines of Operation – connect the force base(s) of operation with its operational objective

- Culminating Point – the point at which the continuation of an offensive operation risks over-extension, counterattack, and possible defeat.

A major theoretical challenge is to represent CoGs. Also, the key is to achieve decisive objectives before the culminating point is reached. Another concept that relates to the smoke and confusion of battle is the "Fog of War." The fog of war is the effect of all uncertainties of combat operations; the friction of combat.

The author discusses the structural perspectives of C2 (Command and Control) embedded systems. A hierarchy of four focuses, defining features, and levels is proposed (See Table 2.1).

Table 2-1: Focus Hierarchy

Focuses	Defining Features	Levels
Nodes	Data	Micro
Links	Structure and Information	Meso
Processes	Rules and Transactions	Meta
Functions	Goals of the C2 Systems	Macro

The book also proposes a decision cycle called the C3EVAL model. This model is a nodal analysis model in which C2 interacts with combat flow.

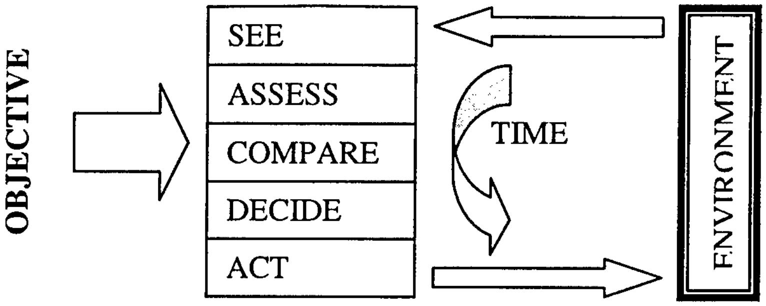


Figure 2-2: C3EVAL Model

From Figure 2-2, you can see that any major decision begins with an objective. The decision process begins by seeing the environment, gathering input about it. Next, the information about the environment is assessed. It is then compared to how it was predicted to react. Subsequently, the decision-maker makes an actual decision based on the primary objective, gathered inputs, and assessments. Finally, the decision-maker acts on a particular course of action. Over a period of time, the decision is acted on in the environment and the environment reacts in the appropriate manner. Feedback is made available to the decision-maker where he/she re-evaluates the situation again. This decision cycle continues until the objectives have been achieved.

2.4.2 OODA Decision Cycle [Boy87], [Joi96]

In 1987, Boyd introduced the OODA (Observe, Orient, Decide, Act) Decision Cycle in his Discourse on Winning and Losing. This provided a fundamental decision-making framework and it was eventually adopted into Joint Pub 3-13.1, *Joint Doctrine of Command and Control Warfare* by the Department of Defense. Both describe this basic framework and the four processes.

The OODA Decision Cycle is decided into four phases or processes: Observe, Orient, Decide, and Act (Figure 2-3). The Observe phase consists of gathering raw data about the battlespace. Next, the Orient phase takes that raw data from the previous phase and converts it into meaningful information. From that information, courses of action or alternative decision are created in the Decide phase and a decision is chosen. Finally, the Act phase is where the decision is actually executed. This is a continual process so it does not stop after the Act phase. Like the C3EVAL Model, the decision is acted upon the battlespace and the decision-maker will gain feedback from the battlespace. Once that feedback returns to the decision-maker, the decision cycle begins again in the Observe phase of a new decision loop.

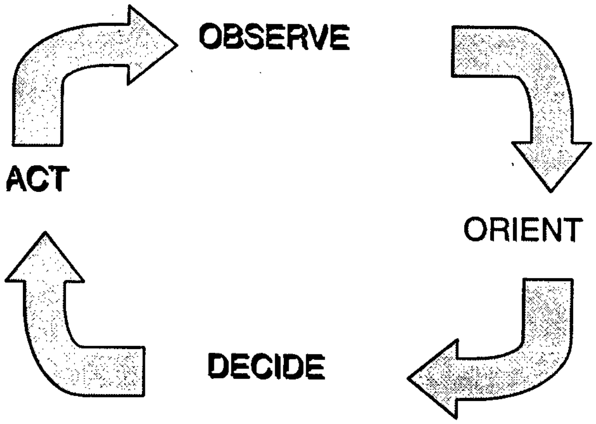


Figure 2-3: OODA Decision Cycle

The implications that can be derived from the OODA Decision Cycle are profound. First, if we improve any phase of our own OODA Cycle, we can improve the overall cycle. Decisions can be made at a faster rate and in turn corresponding reactions can be implemented just as fast. Second, if we improve our OODA cycle to the point that our overall decision process is faster than that of our adversaries, we will be able to outmaneuver and out-decide them. This is referred to as "getting inside the enemy's OODA loop."

2.5 Decision Analysis

This section builds from the uncertainty and decision cycle elements of the previous sections. Here we discuss the systematic method of decision-making called decision analysis.

2.5.1 Making Hard Decisions [Cle96]

Clemens provides an introduction to uncertainty, probability, and decision analysis. Three chapters were of particular interest. Chapter 1 offered an introduction to decision analysis. Chapter 2 discussed the elements of decision problems. Chapter 7 introduced the basics of probability.

Chapter 1 states that decisions are hard because of four reasons: Complexity, Uncertainty, Multiple Objectives - One Direction, and Different Perspectives - Different Conclusions. There are several reasons why we study decision analysis. First, decision analysis results in better decisions and is more preferred than relying on lucky outcomes. Second, decision analysis improves the chances of a better outcome. Third, with decision analysis, one is less likely to experience unpleasant surprises. Fourth, it provides structure and guidance for systematic thinking in different situations. Finally, decision analysis provides an understanding of the problem thoroughly.

Decision analysis requires subjective, personal judgments. An awareness of human cognitive limitations is critical in developing the necessary judgmental inputs. Clemens proposes the following Decision Analysis Cycle:

1) Identify the Decision Situation & Understand Objectives.
2) Identify the Alternatives.
3) Decompose & Model the Problem.
   - Structure
   - Uncertainty
   - Preferences
4) Choose best Alternative.
5) Sensitivity Analysis (What If?)
6) Is Further Analysis necessary?
   - Yes – go to Step 1, 2, & 3.
   - No – Implement Chosen Alternative.

Chapter 2 describes the elements of decision problems. These elements are values and objectives, the decision to make, uncertain events, and the consequences. Values are what matters to you while objectives are the specific things that you want to achieve. An important tool in decision making is the influence diagram. The diagram consists of Value Nodes and Decision Nodes. The Value Nodes represent a number or a calculation and there must be at least one Decision Node. The influence diagram is not necessarily a flow chart since it does not have to be sequential. What it does show is the influence value of alternatives, which eventually leads to the final decision and best solution.

Chapter 2 also discusses other basic elements of the decision problem. Decisions can be divided into simple and complex decisions. Simple decisions have a small number of alternatives while complex decisions have infinitely many alternatives. Probability can be represented in discrete units to measure uncertainty.

Probability is quantifiable. However, words have different meaning to different people. Therefore, it can be difficult to communicate exact values. We can use fuzzy terms and linguistic qualifiers to express membership possibility distributions and alleviate some of the vagueness that goes along with verbal communication. Several methods are available that aid in decision making. In Fuzzy Set Theory, membership indicators (functions) of a set are assigned to attributes. Fuzzy Set Theory is flexible since the various indicator numbers do not have to add to one unlike classical statistics. Bayesian Nets (Belief Networks) uses "prior probability" to handle uncertainty. They are based on Bayesian Theory, not classical statistics and can update or adjust their probabilities

Finally, Chapter 7 covers some more probability basics. The objective of this chapter was to show some ways that probability modeling could be useful in decision problems. We can interpret probability statements in terms of uncertainty. Chapter 7 was also concerned with the chance event. This refers to something about which a decision-maker is uncertain. It usually has more than one outcome. The probability basics were also discussed in Chapter 7:

2.6 Situational Awareness Assistance, Dealing with Uncertainty

Now that we have researched the basics of uncertainty, probability, decision-making and decision analysis, we turn to research applications that apply these concepts of dealing with uncertainty in order to improve the decision-maker's situational awareness. There have been two past research efforts at the Air Force Institute of Technology (AFIT) that focused on improving the situational awareness of the user within a synthetic, virtual environment. In 1993, Soltz [Sol93] designed autonomous agents that were capable of monitoring the battlefield and determining areas of interest to improve situational awareness of the commander. Later, in 1996, Wells took a different approach [Wel96]. His research was designed to focus the user attention through the use of a collaborative workspace.

2.6.1 Graphical Tools for Situational Awareness Assistance for Large Battlespaces [Sol93]

Soltz in his masters thesis, *Graphical Tools for Situational Awareness Assistance for Large Battlespaces*, similarly identified a problem of situational awareness as they pertain to large virtual battlespaces. Although designed to simplify the real-world battlespace, virtual battlespaces can become complicated and overwhelming as they try to capture, interpret, and portray the raw but essential data for the real-world battlespace. As the virtual battlespaces grow in size and complexity, assessing situations within them becomes more difficult. Determining where to focus attention, assimilate, and assess information as it comes in becomes a problem.

To combat this problem, Soltz created Sentinels, autonomous agents that provided situational awareness assistance for users within a large virtual environment. These agents were situated over user-defined areas-of-interest or "watchspaces" as analysis modules, programmed to notify the user if an event with moderate or high risk had occurred. Sentinels employed fuzzy logic because fuzzy logic can recognize a pattern of activity and mimic human judgments concerning the significance of the pattern. Inputs would be taken in and fuzzy logic would be used to combine the uncertainties of given actions within the watchspace. Fuzzy set operations would be performed and a single relative number would be produced. This number would then be converted to a color and bar length and presented visually to the user to represent the watchspace assessment value (risk) associated within the entire watchspace.

This thesis successfully showed the practical application of the Sentinel situational awareness assistance system in an actual synthetic environment. By using a fuzzy logic set theory, the Sentinel system was able to combine and abstract many factors concerning a particular watchspace. This would draw the attention of the user to that watchspace and allow the user to quickly ascertain the relative situation. As a result, large amounts of information concerning a virtual battlespace can now be presented in a timely and efficient manner thereby greatly enhancing the situational awareness of the user.

2.6.2 Collaborative Workspaces within Distributed Virtual Environment [Wel96]

Wells investigated ways "to improve situational awareness within large-scale virtual environments." He determined that the fundamental problem is that of isolation. Commanders are isolated with no way to communicate with each other in the synthetic battlespace. Battlespace awareness can be enhanced through communications technology in a Computer Supported Cooperative Workspace (CSCW). By using collaborative communication, it offers the widest range of communication options to all commanders in the battlespace. Multiple commanders can work together. The likelihood that vital information will be overlooked is decreased since commanders are no longer alone and many commanders will be viewing the same information simultaneously. "Two heads are better than one."

2.7 Summary

The primary thrust of this chapter was to provide background research on several topics related to this research effort. The key issues learned from this chapter include understanding the "fog of war," what is it and how to combat it. In addition, the need for information systems that can aid the decision-maker has been firmly identified. We examined the military's use of a decision-making models for coping with uncertainty and learned about decision analysis, a continuos and systematic decision-making. Finally, we explored past research attempts in dealing with uncertainty through situational awareness tools. The next chapters will expand on these key issues that were learned from this background research.

3 Methodology

"The confidence of ignorance will always overcome indecision of knowledge."

Anonymous

"To succeed in life, you need two things: ignorance and confidence."

Mark Twain

3.1 Overview

The previous chapter described and summarized the concepts of uncertainty, decision analysis, and some of the current research thrusts for decision support applications in uncertain environments. This chapter will focus on the development of a decision support tool. This tool will not make the actual decisions for a user but will focus the attention of a decision-maker on critical but uncertain multi-criteria information. Once developed, this simple decision support tool will be used to validate my primary hypothesis stated in paragraph 1.3.

This chapter describes the considerations made during the data orientation process of the decision support tool. Briefly, further considerations for the development of the simple system as well as its intended domains will be discussed. The finer design and implementation details of the decision support tool will be addressed in Chapter 4; this chapter is dedicated to a higher-level discussion for answering the following questions:

- What did I set out to do?
- How do I hope to accomplish this?

3.2 Domain

As stated above, the primary focus of this thesis is to develop a method of focusing the attention of the decision-maker on uncertain information. Decision Support Tools can aid decision-making in a wide variety of domains. As mentioned in Chapter 2, applications that use a type of automated reasoning in some form or another have been found in many fields. It stands to reason that if one were to design such a tool, it should also handle a wide variety of domains. Therefore, I intend to create a decision support tool that can be employed across a broad range of opportunities and applications. It should be independent of a particular domain and still be able to sort through large amounts of data to assist decision-making in multi-criteria problems.

With the decision support tool, I will consider a "toy" problem that will be easy to understand and fundamental to many potential users. With this in mind, I chose to augment the basic decision support tool with a car-buying domain, specifically buying a minivan. In general, purchasing a car is something with which everyone has some experience. Therefore, it should be easy to relate. It also involves making a decision based on several quantitative and qualitative factors, yet these factors are modest in actual number and quite tangible. Data from *Consumer Reports* [CON99] and *Popular Mechanics* [Old99] magazines, specifically for 1999 minivans, will be used and decision techniques similar to those used in PC *Computing Magazine's* "Decision-Maker" column [Car99] will be tailored to the automobile realm. With minimal augmentation to the data set, the toy problem of purchasing a minivan will be used to validate the basic model. In this capacity, this simple decision support system will serve as a prototype to prove the concept of the attention focusing method.

3.3 Uncertainty

Most real world problems that a decision-maker is faced with contain an enormous number of criteria, much of it uncertain. In many cases, this can be too much for a single person to handle. One such real world, multi-criteria realm, is the military intelligence community. Here, critical decisions in the name of national security must be made with a high degree of certainty. Specifically, within the Dynamic Information Operations Decision Environment (DIODE), information from various sensors can provide an analyst with the raw data that can be sorted and oriented so that meaningful information can be extracted. This data from the sensors may be contradicting, missing, or possess a certain level of uncertainty. Therefore, in order to introduce the concept of uncertainty to the minivan buying scenario, I assigned certainty values to the minivan data in a similar manner that the military intelligence community might assign a confidence value to a particular piece of gathered information.

All data is inherently uncertain. Primarily, there are four causes of uncertainty: *ignorance*, *unreliability*, *unknowable*, and *omission*. This thesis does not address a particular aspect of uncertainty but attacks uncertainty in general. This thesis does address the need for a new methodology of focusing the attention of a decision-maker. There are two possible methodologies. The first is better information visualization and the second is data orientation. This thesis will concentrate on the latter, data orientation.

3.4 Information Overload

Now that I have identified what I intend the decision support tool to process, I will discuss how it will process it. As mentioned in Chapter 1, the fundamental problem is that there is too much raw data. Meaningful information is abundant but obscured. We need a consolidated methodology that is able to "collect raw data, organize it, analyze it and assimilate it [AF2025]." This is the prime motivator behind this research effort.

There are two methods of dealing with information overload. First, we could simply reduce the clutter of information. This can be accomplished by actually throwing away the excess or extraneous information, or simply by ignoring the irrelevant information. Either way clutter is reduced. Second, we can direct the focus of attention of the decision-maker. By actively guiding the decision-maker to the important hot spots, or centers of gravity, we can focus him on the most important information at hand.

I believe that both methods can be accomplished at the same time. By reducing the clutter, we increase the proportion of relevant and useful information. The decision-maker is forced to focus on what is important. All that remains is the appropriate meaningful information. Therefore, this thesis will attempt to accomplish both. It may not reduce the amount of raw data available to the user, but it will increase the proportion of valuable and meaningful information. By doing so, the tool essentially throws away unnecessary raw data then updates and refreshes the data set. The data is oriented in such a manner that the user can do nothing but focus on the most important decision factors.

3.5 OODA Loop – Orient Tool

"Data orientation" is the title that I am giving to the name of the process that the decision tool will undertake. This was primarily derived from the OODA Decision Cycle or Loop that was discussed in Chapter 2. I set out to create a tool that should be able to take already meaningful information and re-orient it into something that the decision-maker would find useful. Furthermore, I believe that the decision-maker should be focusing on areas of importance that more than likely had uncertainty associated with them. Previous research efforts have labeled such areas as "Hot Spots," "Centers of Gravity," and "Watchspaces." The last term seemed most appropriate to me. As a derivative of the term watchspace, it felt appropriate to name the decision support tool WATCHDOG, to look over these watchspaces for the decision-maker.

As you can see from Figure 3-1, conceptually, the WATCHDOG Decision Support Tool would be one of a series of programs that could aid the battlefield commander.

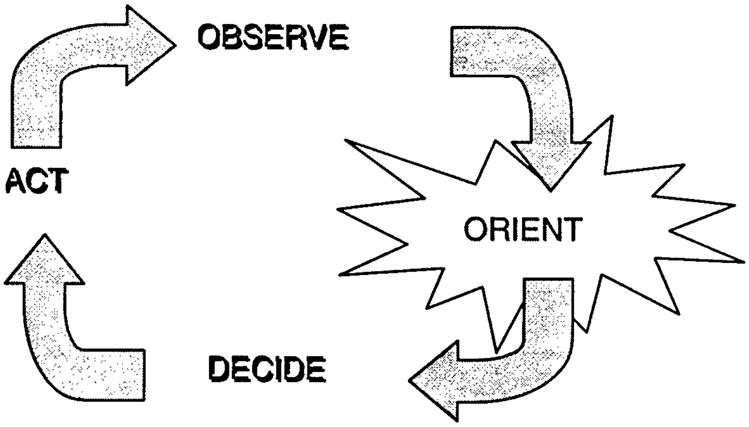


Figure 3-1: OODA Decision Cycle

WATCHDOG corresponds only to the “Orient” phase. It should not be used as a course of action (COA) generator. That would correspond to the “Decide” phase. WATCHDOG is simply a decision support tool. It is not meant to take the place of a decision-maker, but rather to aid the analyst in sorting large amounts of uncertain information and quickly orienting the data into something more meaningful.

3.6 Structured Analysis of a Decision Support Tool

It is my intention to use a Structured Approach to analyzing, designing, and implementing. This will involve three steps: a Structured Analysis step, a Design step, and an Implementation step. First, the Structured Analysis step is a high-level look at the problem. It will entail performing a Requirements Analysis, which results in a System Specification. Second, the Design step will take the Systems Specification and add more details, refining the problem by allocating functionality into convenient program modules. Third, the Implementation step will transform the modules from the previous step into executable code on the desired platform. The Structured Design and the Structured Implementation steps will be discussed further in the next chapter. Chapter 3 will concentrate on the Structured Analysis Step. In this chapter, a Context Diagram and a Data Flow Diagram will also be presented.

3.6.1 System Description

In order to derive the requirements of the decision support tool, I had to figure out how I wanted the tool to operate. The following paragraphs provide an overall system description of how I initially wanted the WATCHDOG Decision Support Tool to operate.

The tool will gather inputs from the user. The inputs will be in the form of decision preferences with regard to the domain. These preferences will be collected interactively through a menu driven system. The objects of the domain data will have multiple characteristics upon which multiple user decision objectives will be based. In addition, these objects will also have a corresponding certainty value associated with them.

Next, the model will attempt to find the best solution based on the preferences provided. The model will disqualify elements from the solution space that do not meet the given preference. The model will take the first preference and "trim" the data accordingly. As the model looks for the best answer it will check for objects that have exceeded a user-specified uncertainty threshold. Furthermore, it will maintain both a "trim list" of objects that have definitely met the user's preferences as well as maintain an "uncertainty list" consisting of objects that may meet the user's preferences but have been deemed too uncertain.

The search for a solution will be an iterative process. That is, the tool will continue to query the user for preferences and keep performing preference and uncertainty checking against each preference factor until all of the factors are exhausted. Through each iteration of data disqualification, the solution space of best answers will be further reduced.

Upon resolution of decision factors, the model will display its results. It may return one object or many objects. In either case, the tool will also return the list of possible solutions in the form of the uncertainty list. In addition, it will caution the user which elements have exceeded the uncertainty threshold.

It is prudent to note that the top response is not necessarily the "best" answer. All answers in this category should be considered equal. The decision to choose the optimal solution is ultimately left up to the user. With these pareto-optimal problems, the user will be able to take the corresponding uncertainties associated with the answers into account before making a final decision. Again, this tool should be considered to be a decision support tool and not a decision-making tool.

3.6.2 Requirements Analysis

From the above system description, we can derive the following requirements:

- Data elements will have multiple characteristics.

- Each characteristic will have a magnitude and a corresponding certainty value associated with it.

- User Attribute and Uncertainty Threshold Preferences will be obtained interactively through a menu-driven system
- The data from the solution space will be iteratively trimmed until the preferences are exhausted.
- Upon preference checking resolution, the tool will display its results.
- The associated uncertainty values will also be displayed.

3.6.3 Context Diagram

The following is the high-level Context Diagram (Figure 3-2) of the WATCHDOG Decision Support Tool as derived from the System Requirements:

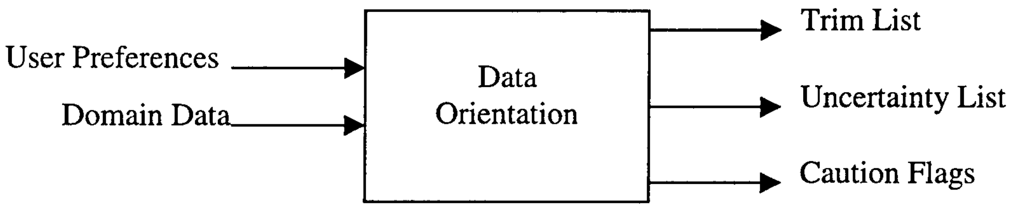


Figure 3-2: Context Diagram

At a very high level, the WATCHDOG tool will accept the User Preferences and the Domain Data as input. It will then perform a Data Orientation process. WATCHDOG will produce as output a Trim List, an Uncertainty List, and possible Caution Flags.

3.6.4 Data Flow Diagram

The following is the Data Flow Diagram (Figure 3-3) for the WATCHDOG Decision Support Tool derived from the System Requirements:

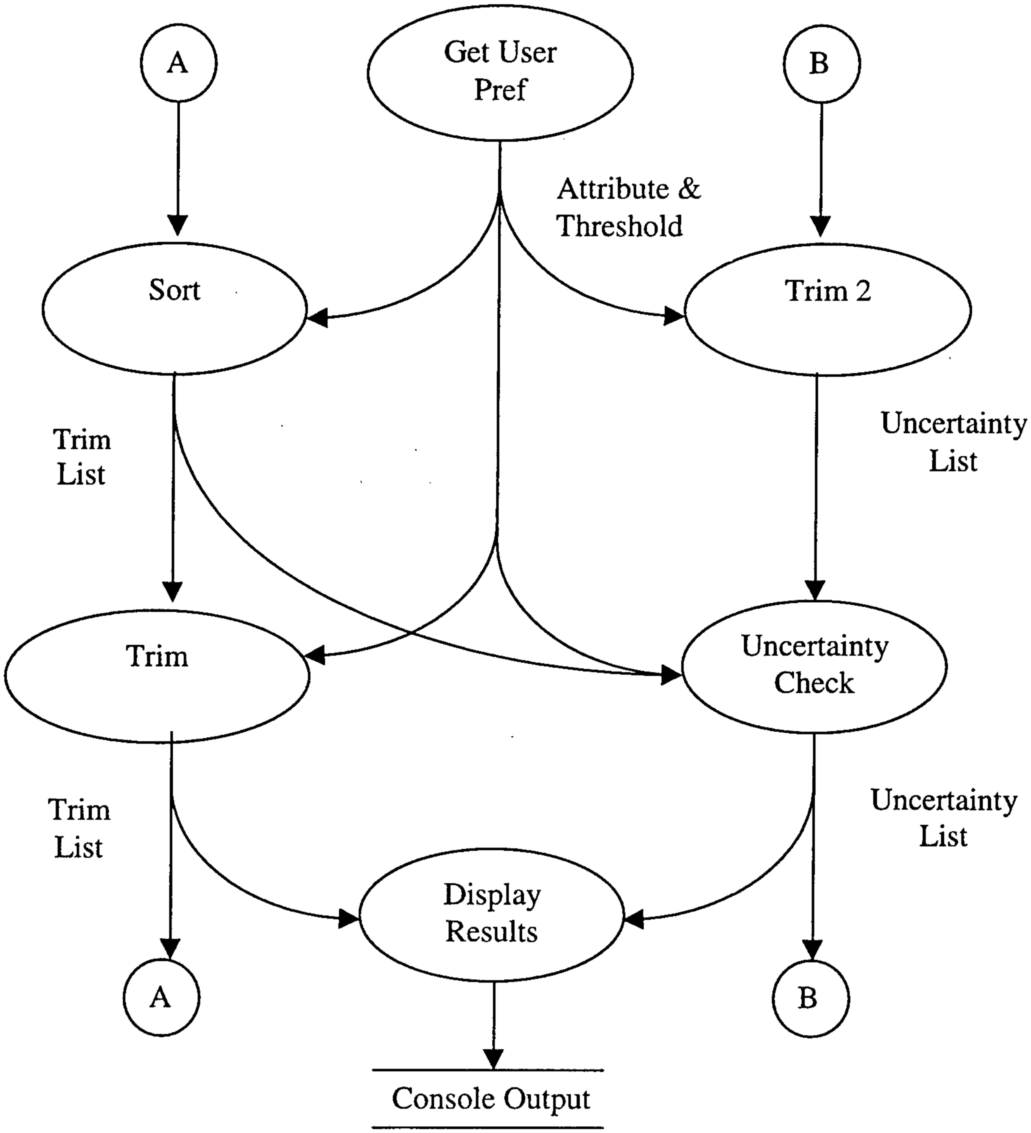


Figure 3-3: Data Flow Diagram

In Figure 3-3, we can clearly see two distinct path of logic for a particular run-level. The left path begins with the top “A” terminal and demonstrates the path of logic that the Trim List flows. It ends with the “A” terminal on the bottom. Likewise, the right path or the Uncertainty List path begins with the top “B” terminal flows down to the bottom “B” terminal. These terminals represent a method of illustrating continuity between run-levels. As a path of logic ends at the bottom terminal, it completes one specific run-level. Subsequently, the next run-level will then begin anew at the top terminals. This pattern repeats as often as necessary. The specific routines and data flow are described below.

The Get User Pref routine will first ask the user for the most important preference. The preference, consisting of the Attribute and its Uncertainty Threshold is first passed to the Sort routine. The Sort routine receives the Trim List from the top terminal A. The Sort routine sorts the Trim List based on the Attribute. It then routine passes the Trim List to the Trim routine and to the Uncertainty Check routine. The Trim routine accepts the Trim List from Sort and proceeds to trim the list based on the preferences passed from the Get User Pref routine. Once complete, the new Trim List is passed to the Display Results routine and also to the next run-level through the bottom terminal A.

On the other path, the Trim 2 routine will take the Uncertainty List from the top terminal B and trim those objects that certainly fail based on the preferences provided by the Get User Pref routine. Trim 2 then passes the updated Uncertainty List to the Uncertainty Check routine. This routine takes the sorted Trim List from the Sort routine and extracts those objects that are too uncertain, based on the Attribute and Uncertainty

Threshold. Uncertainty Check then adds these objects to the Uncertainty List. This new Uncertainty List is then passed to the Display Results routine and to the next run-level through the bottom terminal B.

Finally, the Display Results routine will display the final End Product to the console output. This End Product will consist both the current Trim List and Uncertainty List.

3.7 Summary

This chapter presented a high-level overview of the methodology used for this research. The purpose of this methodology was to answer the following questions:

- What did I set out to do?
- How did I hope to accomplish this?

The main objective of this thesis is to validate the following question: Can a method that can focus the attention of the decision-maker on uncertain information be developed? I set out to validate this hypothesis by creating a decision support tool called WATCHDOG. In this chapter, I described how the WATCHDOG tool would accept data from various domains. I also described how uncertainty would be introduced into this system in a similar fashion that is used in the military intelligence community. I briefly discussed the need to overcome information overload and how this tool would be one of two methods that could handle information overload. I further examined how conceptually this tool would be used in the "orient" phase of OODA decision cycle and that it could be one of series of tools that can aid the decision-maker by expediting the overall decision process.

This chapter elaborated on the structured analysis approach that I would use to determine how the tool should work. Factors affecting the structured analysis of this research were presented as well as some of the motivations driving it. I provided a high-level system description of how I thought the tool should operate and then derived some basic system requirements from that description. In addition, a high-level context diagram was developed from this description as well as a data flow diagram, which further refined the system. Ultimately, all of the above factors helped to establish the objectives and expectations of this thesis in the form of several testable hypotheses as described in paragraph 1.3.

4 Design And Implementation

"A good plan, violently executed now, is better than a perfect plan next week."

George S. Patton, General (1885-1945)

4.1 Overview

The WATCHDOG Decision Support Tool is an attempt to develop a demonstration system that validates my hypothesis: a method can be developed that can aid a decision-maker in the process of handling information overload and guiding him to uncertain information. WATCHDOG is the name of the actual program, which is written in the JAVA Programming Language. This chapter highlights many of the detailed design decisions made while creating this tool.

4.1.1. Basic Terminology

In order to provide a better understanding of how the WATCHDOG Decision Support Tool operates, I feel that it is prudent to first discuss a few basic terms. This program implements recursion when searching for a solution. Because the program also implements backtracking, it is important to keep track of the different levels of recursion. Consequently, the Run or the Run Level is used to represent the level of recursion. Moreover, because of the recursive programming and backtracking, the state of any given level also must be maintained. This was accomplished through the creation and implementation of a State Vector. A State Vector is a dynamic data structure that contains a number of variables that characterize a particular state at a particular instance. A State Vector entails the Sort Vector, the Trim Vector, the Uncertainty Vector, the

Choice Vector, the Uncertainty Level Vector, the Percent from the Top, the Cutoff, and the Run Level, all of which will be discussed below. Figure 4-1 is a graphical representation of a State Vector.

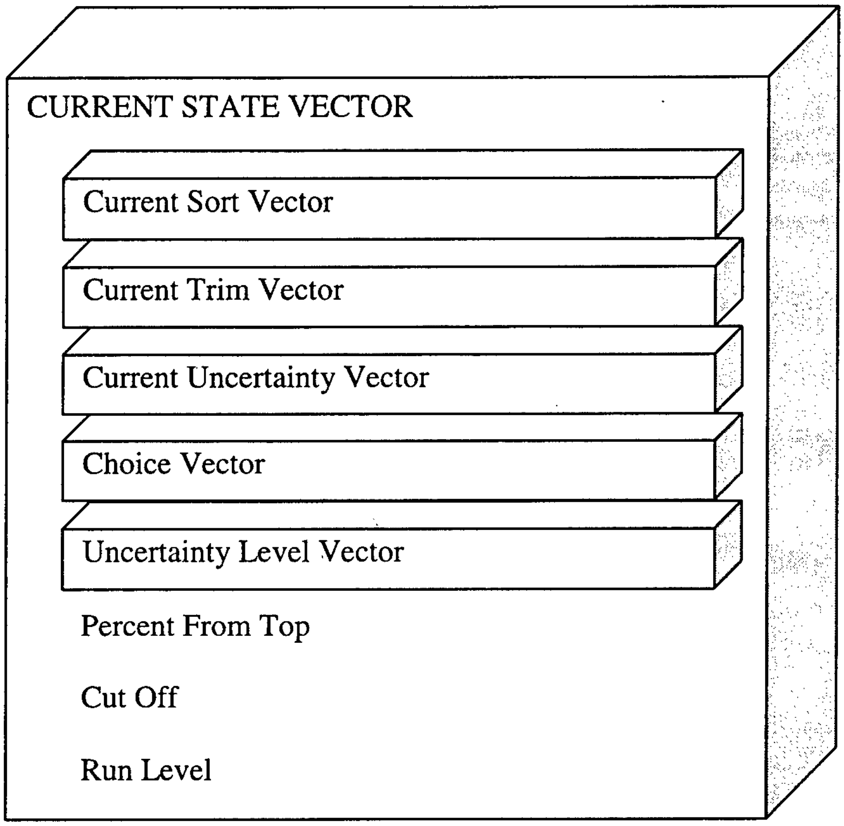


Figure 4-1: Current State Vector

Vectors are very flexible data structures since they are dynamic. They can grow or diminish as needed. For the purposes of this document, list and vector will be used synonymously

The Sort Vector is the list of objects from the domain data that are ordered based on a particular attribute. The Trim Vector is the list of objects from the domain data that have met the decision-maker's preferences. It can be thought of as the "list of definites."

The Uncertainty Vector is the list of objects from the domain data that may have met the decision-maker's preferences but where associated certainties for a particular attribute have been deemed too uncertain. Given further time and resources to mitigate their uncertainties, these objects may still be possible solutions. The Uncertainty Vector can be thought of as the "list of maybes."

The Choice Vector is the list of choice preferences provided by the decision-maker. Each preference consists of an Attribute and an Uncertainty Threshold. The Attribute represents the name of the attribute which the user believes is important or significant while the Uncertainty Threshold represents the lowest possible certainty value that the user is willing to accept for that particular attribute. These attribute/threshold pairs are stored in the Choice Vector in the order of most important preference to least important preference.

Another aspect of the WATCHDOG tool that needs to be understood is the format of the domain data objects. Each object of the domain data will have multiple attributes. The more attributes that the domain objects have imply the more complex the domain data will be. Each attribute will have two characteristics: a Magnitude and a Certainty. The Magnitude represents the actual value that the attribute takes on with the appropriate units while the Certainty represents the amount of assurance that you have with the associated attribute. For example, if we are provided with "Toyota Sienna $26,769 (60%)" we can say that we have a 60% assurance that the price of the Toyota Sienna is $26,769. Furthermore, Certainty also relates to the proximity of the indicated value to the true value. An attribute with a high certainty means that the true value lies closer to the indicated value than an attribute with a low certainty. As indicated by Figure 4-2, the solid horizontal line on both graphs represent the indicated value or magnitude while the area below the curved line represents the possible range of values that the true value can take. This does not establish floor or ceiling values for the indicated value, but it does give you a good idea of the range that the true value may be from the indicated value.

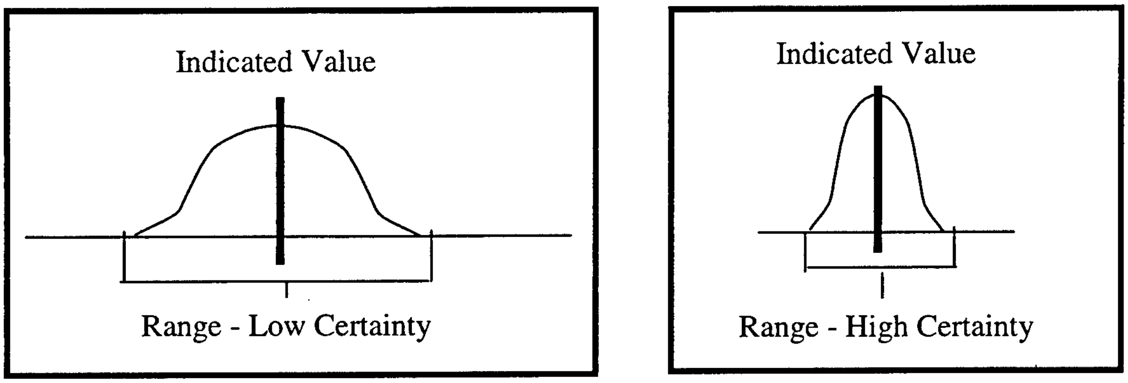


Figure 4-2: Range of Possible Values

Furthermore, it is important to note that the Certainty Values as illustrated in Figure 4-2 were not derived from some kind of Gaussian or normal distribution function. They did not have to be represented by an area below a curve. The Certainty Value could have been represented by a bar graph, pie graph, or some other illustration that would show a range of possible values. The Certainty Values are simply my method of representing a generic, non-specific type of uncertainty without regard to how they were obtained. The Certainty Values are not probabilistic and are not values over an ensemble but they may be viewed as a degree of membership of a particular attribute.

The Cutoff is the boundary value that the program uses to establish which objects should be on the Trim Vector and which ones should not with regard to the currently selected attribute. This Cutoff is established with Percent From Top value. Percent From

Top is the percentage removed from the top-most value and is where the Cutoff value should be set.

4.1.2. Basic Operations

This section will discuss the basic operations of the WATCHDOG Decision Support Tool. It is a further refinement of the initial system description provided in Section 3.6.1. In order to help the decision-maker to sift through large amounts of uncertain data, the program needs to know what attributes are important to the user. The program will ask the decision-maker for a series of preferences, one at a time, of what he/she believes are paramount. These preferences are the most important attributes and their associated uncertainty thresholds are called User Thresholds. A User Threshold is the lowest possible certainty that the decision-maker is willing to accept for a particular attribute.

Once it accepts all of the decision-maker's preferences, the program will sort through the domain data based on user preferences. A Cutoff value will then be established. Next, the program will trim the objects from the domain data that certainly do meet the decision-maker's preference. Those objects that are equal to or above the Cutoff value will be placed on the Trim Vector. The program will also maintain a list of objects that may meet the decision-maker's preferences but are too uncertain. These objects will be placed on the Uncertainty Vector. The program will provide the decision-maker with the "best-effort" results.

"Best effort" means that through recursion and backtracking, the program will search for the best possible set of solutions. It will start conservative and attempt to find a solution with a high degree of certainty. Failing to find a solution, the program will then relax the Uncertainty Level and try to find a solution again. It will keep relaxing the Uncertainty Level until it finds a solution set or until it meets the User Threshold. If the program cannot still find a solution, it will push below the User Threshold, starting with least important preference, in order to provide the next-best alternative of answers. The program will notify the decision-maker at this point if it has to exceed the User Threshold.

4.2 Watchdog Program Flow

This section describes the overall WATCHDOG program flow. Within the WATCHDOG program is the Main method or top-level routine. From the Main method, all other methods are spawned. The first step of the Main method is Initialization. With Initialization, all various variables and data structures are set to their starting positions or values. Once Initialization has occurred, the domain data is then loaded and the objects are allocated into the appropriate data structures. The Main method then displays the Main Menu for the user with the following options: Launch Watchdog (method), Load New Domain Data, List The Domain Data, Reset Debugging Levels, or Quit. Based on the user's menu choice, the corresponding actions are then executed. Upon completion of that action, the program returns to the Main Menu. This sequence continues until the user selects the Quit option upon which the program terminates. Figure 4-3 illustrates the overall program logic of the Main method. Further details about specific aspects of the Main method are provided in the following paragraphs.

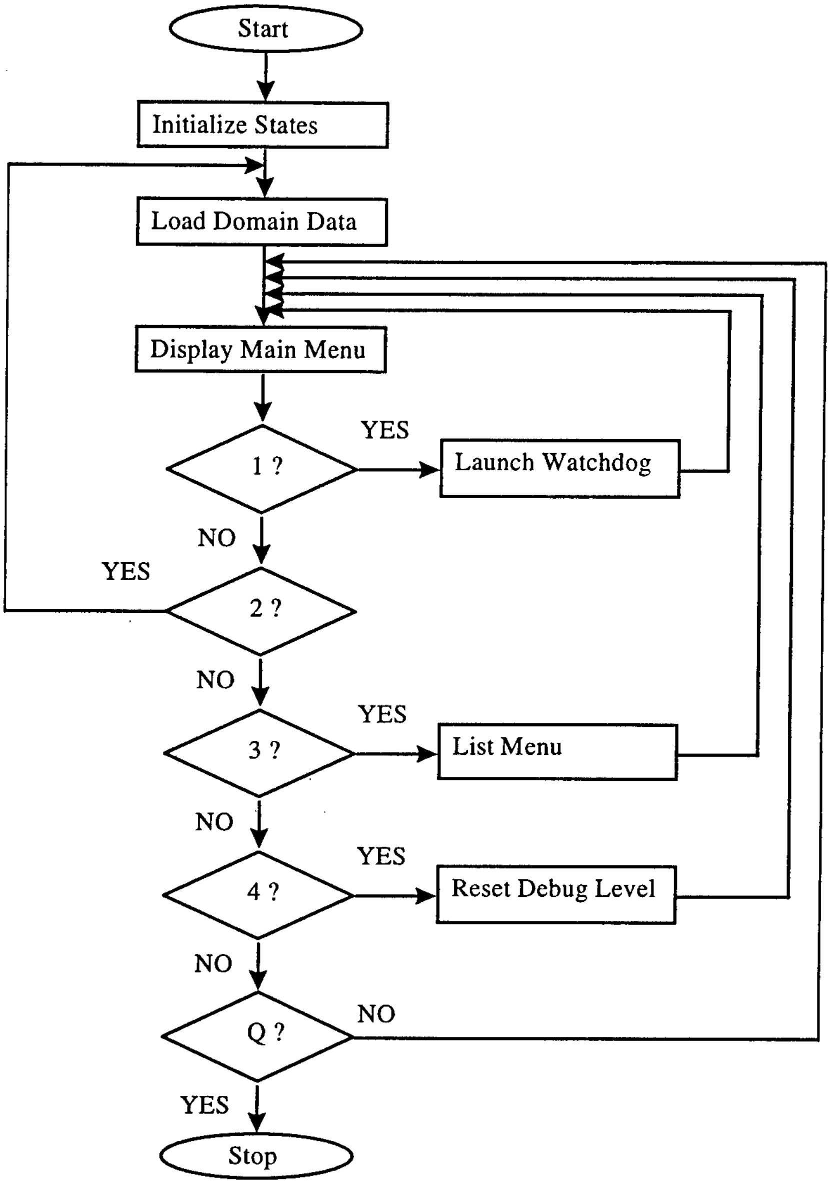


Figure 4-3: The Main Method Flowchart

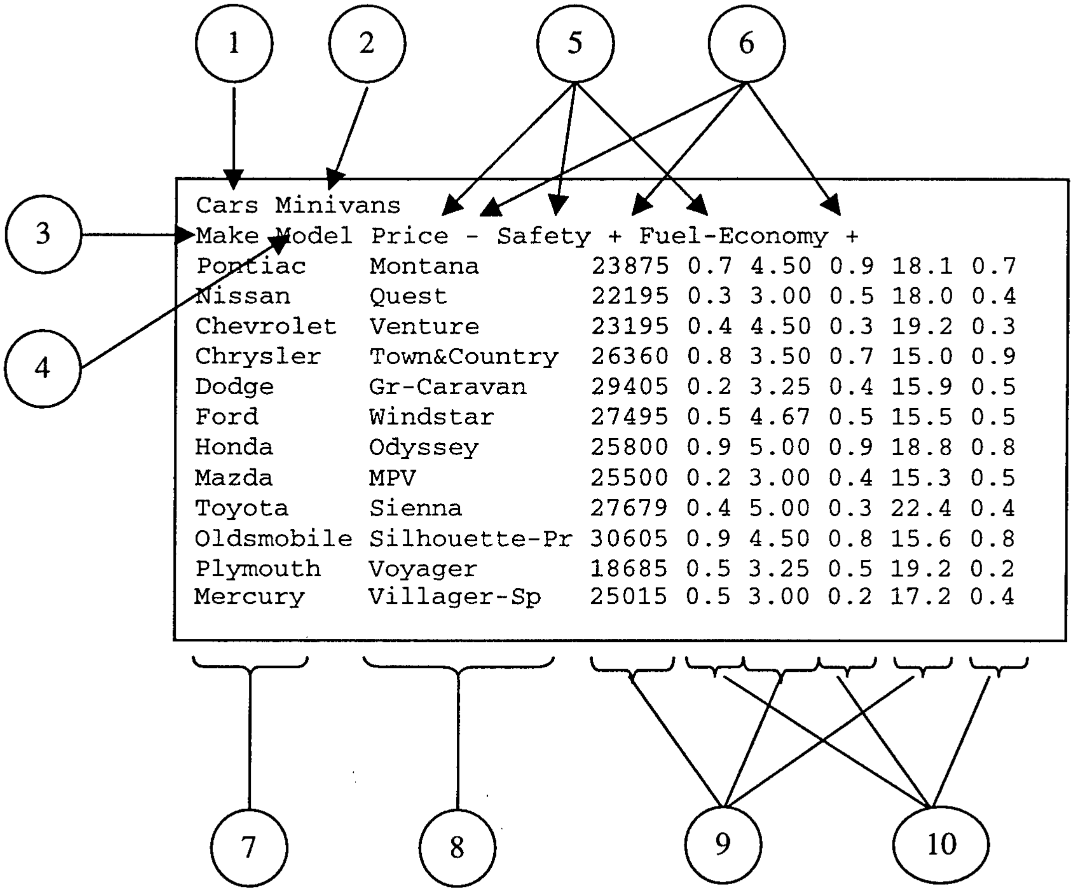
4.2.1 Load Domain Data

After initialization, the decision-maker will be prompted for the name of a data file to load. To have this file loaded, the decision-maker must enter the correct file name and extension along with the correct directory path of the desired data file. Once a valid file name is entered, the objects of the data file will be loaded into the appropriate data structure line-by-line. It is important to note that in order for the program to load the data objects correctly, the data file must be in the standard format described below.

The first line of the data file must contain the class and sub-class of the domain data. For example, in Figure 4.4, the class is "Cars" (Item 1) while the subclass is "Minivans" (Item 2.) Although, not currently used, this functionality is made available for future use. The Load Domain Data method takes advantage of the built-in JAVA language word tokenizer. Therefore, in order to read in the objects of a line from a text file, a space must delimit any two objects in order for Java to tokenize and recognize these objects as separate entities.

The second line of the data file will contain names of the attributes. The first two objects of this line are the first and second identifiers. In Figure 4.4, Items 3 & 4 illustrate the first and second identifiers, "Make" and "Model." At first, it does not appear to be obvious why two identifiers are used to associate with only one object. Frequently, objects in the real world have a first and second name with which they can be associated or identified. For example, most people in the western world can be identified with a first and last name, e.g. Thomas Jefferson. Minivans, the domain of the toy problem, can be identified with a make and model, e.g. Toyota Sienna. This identification convention can be applied to a variety of different domains, so it would also be appropriate for a decision support tool that is supposed to employ any object of all types. The next data objects that are read in are the actual names of the attributes and their corresponding sorting order. The attribute names will be strings of up to 15 characters in length and the sorting orders will be strings of one character length possessing either a value of "+" or "-." Each attribute should be characterized with a sorting order. A sorting order is the appropriate direction of sorting for a given attribute, either ascending or descending. Whether a particular attribute should be minimized or maximized is left up to the owner of the domain data. There is not a set number of attribute-sort order pairs that can be read in. Although the program is designed to handle zero pairs, to be effective there should be at least one attribute/sort order pair to assess. Figure 4.4, Item 5 shows that the attribute names are "Price, " "Safety, " and "Fuel-Economy" while Item 6 shows that their corresponding sorting orders are "-," "+," and "+."

Finally, any line after the second line contains the actual data of the individual objects of the domain. There is no limit to the number of objects in the domain, although to be practical there should be at least two objects within a domain to compare. The first two objects of these lines will be the first and second identifiers for each particular element as shown by Figure 4.4, Items 7 & 8. In a similar fashion to the second line, the objects that follow the identifiers are read in pair-wise. These paired objects will be double-float integers that represent the magnitude value (Figure 4.4, Item 9) of the actual data associated with an attribute as well as its corresponding certainty value (Figure 4.4, Item 10). This certainty value represents the confidence or degree of certainty that can be associated with its corresponding attribute data. Furthermore, there should be the same numbers of domain data pairs as there are attribute name-sort order pairs from line number two.



```
Cars Minivans
Make Model Price - Safety + Fuel-Economy +
Pontiac    Montana        23875 0.7 4.50 0.9 18.1 0.7
Nissan     Quest          22195 0.3 3.00 0.5 18.0 0.4
Chevrolet  Venture        23195 0.4 4.50 0.3 19.2 0.3
Chrysler   Town&Country   26360 0.8 3.50 0.7 15.0 0.9
Dodge      Gr-Caravan     29405 0.2 3.25 0.4 15.9 0.5
Ford       Windstar       27495 0.5 4.67 0.5 15.5 0.5
Honda      Odyssey        25800 0.9 5.00 0.9 18.8 0.8
Mazda      MPV            25500 0.2 3.00 0.4 15.3 0.5
Toyota     Sienna         27679 0.4 5.00 0.3 22.4 0.4
Oldsmobile Silhouette-Pr  30605 0.9 4.50 0.8 15.6 0.8
Plymouth   Voyager        18685 0.5 3.25 0.5 19.2 0.2
Mercury    Villager-Sp    25015 0.5 3.00 0.2 17.2 0.4
```

Figure 4-4: Format of Domain Data

4.2.2 Display Main Menu

Once the initial domain data is read in, the Main Menu will be displayed. A number of options are available for the user to choose. The decision-maker's selection will determine the next course of action for the program. Once that course of action has finished executing, the program will return to the Main Menu. This Main Menu loop will continue until the user has selected the Quit option. The following paragraphs are descriptions of the Main Menu options.

4.2.2.1 Menu Option 1 – Launch Watchdog

This option will be covered in a later section.

4.2.2.2 Menu Option 2 – Load New Domain Data

If menu option 2 is selected, the user desires to load a new domain data set. The main menu will call the Load Domain Data method and the user will be able to select a different set of data to compare. All local variables will be re-initialized in preparation for a new program execution.

4.2.2.3 Menu Option 3 – List Domain Data

If menu option 3 is selected, the user desires to see the domain data listed. Another menu is displayed and the user is given the option of viewing various lists of all of the domain data. One option will allow the user to view a list of the domain objects in the order that they were loaded. The other options will allow the user to view a list of domain objects sorted by any single attribute.

4.2.2.4 Menu Option 4 – Reset Debugging Levels

If menu option 4 is selected, the user desires to reset the debugging levels. Currently, there are three levels of debugging: Terse, Verbose, and Cacophony. The Terse level debugging will enable only the statements that concern major processing events such as processing status, backtracking status, and results of program run. The Terse level is essentially normal operations, statements that would normally be seen if all of the debugging statements were removed. The Verbose level debugging will enable the process entry and exit statements along with the names of the parameters being passed in and out in addition to all of the statements from the Terse level. Finally, the Cacophony debugging level will display the actual values of all of the parameters in addition to the statements from the previous two levels. The default debugging level is initially set at the Terse level.

All of the functions of the Main Menu were tested to ensure that they were working properly. Several requirements were sought for compliance. First, I checked to see if the corresponding course of action was taken once the test subject selected a menu option. Second, I tested to see if the selected course of action functioned properly. Third, with the exception of the "Quit" option, I checked to make sure that upon completion of the requested course of action, the program would return to the main menu. The results of the Main Menu Testing are as follows:

Table 4-1: Main Menu Testing Results

#	Menu Option	Correct Corresponding Function Executed?	Functioned Properly?	Upon Completion, Returned to Main?
1	Launch Watchdog	Y	Y	Y
2	Load New Domain Data	Y	Y	Y
3	List Domain Data	Y	Y	Y
4	Reset Debugging Levels	Y	Y	Y
Q	Quit	Y	Y	N/A

4.3 Launch Watchdog

The Launch Watchdog method is meant to be called recursively. That is, if the program identified a set of possible solutions, this method will call itself any number of times in order to find a solution space that satisfies the next preference. The stopping condition will be one of the following: the decision-maker wishes to quit, the program has found at least three objects that meet the decision-maker's preferences above the Uncertainty Level, or the program has found less than three objects and the User

Threshold has been reached. By using this form of direct recursion [HC97], the program is able to self-maintain its state values at each level of recursion, thereby alleviating much of the coding effort of the programmer. The system will take care of the bulk of the work [DL91]. Furthermore, recursion simplifies the code thus increasing the overall clarity of the actual program. The flow of program logic is illustrated by Figure 4-5, while the following paragraphs in 4.3 discuss in detail the various methods and functions within Launch Watchdog. Please note from Figure 4-5 the various flows of logic, especially the looping constructs. These looping constructs are implied in the Trim method and Adjust function paragraphs, however, neither actually performs these loops. It is important to understand that it is the interplay between the two that provides the "adjust until satisfies" behavior.

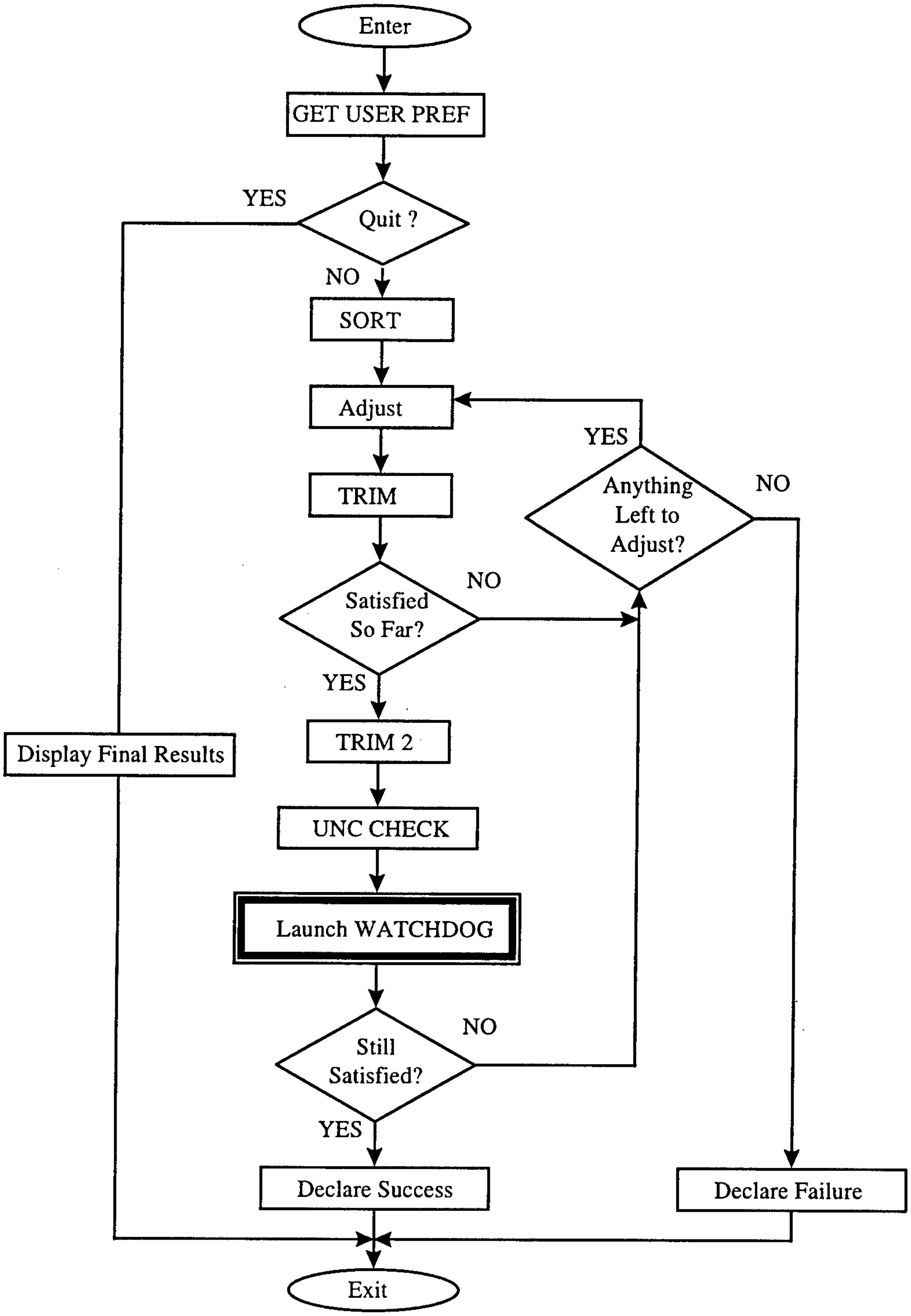
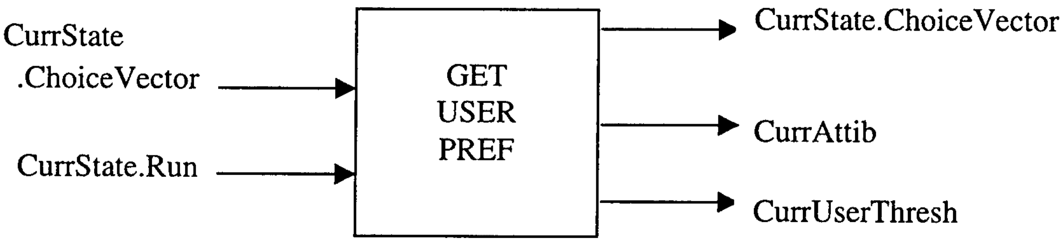
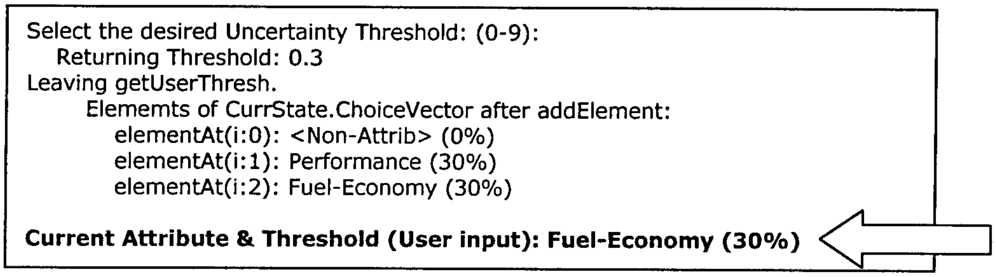


Figure 4-5: The Launch Watchdog Method

4.3.1 Get User Preference



The first method or process in the Launch Watchdog method is the Get User Preference method. As its name implies, the purpose of this method is to obtain the user preferences. These user preferences will be used to sort the domain data. The Get User Preference method can obtain the next user preference by one of two ways: either through console query or by retrieving the preference from the Choice Vector. If the program is not backtracking, it will query the user for the next most important attribute (Figure 4-6). Next, the program will query the user for the uncertainty threshold associated with the attribute. This User Threshold represents the lowest degree of certainty that the user is willing to accept when making decisions regarding that attribute. Both the attribute and its Uncertainty Threshold are then added to the end of the Choice Vector. This vector will be used to keep track of previously made choices while backtracking.

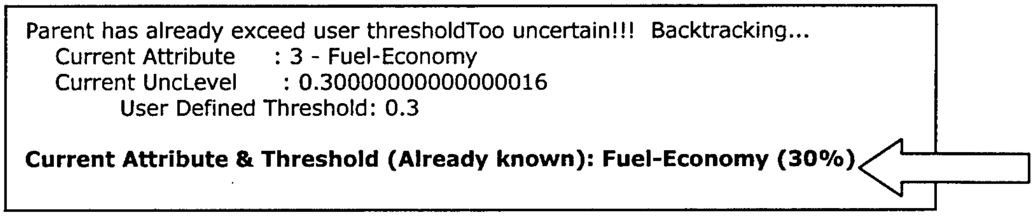


```
Select the desired Uncertainty Threshold: (0-9):
   Returning Threshold: 0.3
Leaving getUserThresh.
      Elememts of CurrState.ChoiceVector after addElement:
         elementAt(i:0): <Non-Attrib> (0%)
         elementAt(i:1): Performance (30%)
         elementAt(i:2): Fuel-Economy (30%)

Current Attribute & Threshold (User input): Fuel-Economy (30%)
```

Figure 4-6: Get User Pref (Console Input)

If the program is backtracking, instead of querying the user again for a choice that is already known, the program accesses the Choice Vector and retrieves the appropriate attribute and corresponding User Threshold (Figure 4-7). When backtracking, the program will know which user choice is the correct choice by the Run Level which corresponds to the same position in the Choice Vector.



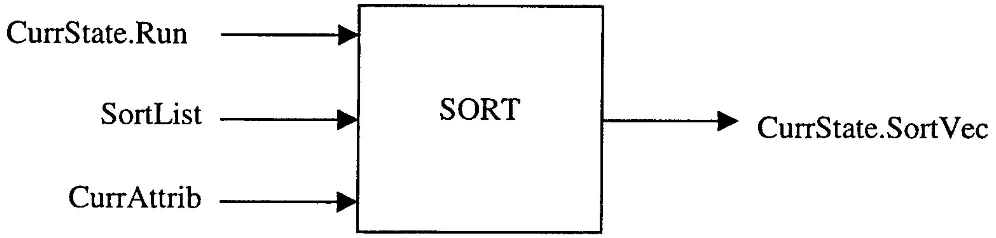
```
Parent has already exceed user thresholdToo uncertain!!!  Backtracking...
   Current Attribute     : 3 - Fuel-Economy
   Current UncLevel      : 0.30000000000000016
         User Defined Threshold: 0.3

Current Attribute & Threshold (Already known): Fuel-Economy (30%)
```

Figure 4-7: Get User Pref (Already known)

Furthermore, anytime an attribute and threshold pair is retrieved, the program will notify the user through console output of the event. The console output statements will also distinguish between a preference just received (Figure 4-6) and a preference that is already known (Figure 4-7). Once the preference is obtained, both the Attribute and User Threshold are passed back to the Launch Watchdog method.

4.3.2 Sort



Once the program has a Current Attribute, the domain data is ready to be sorted. The Sort method receives the Current State Vector, which contains all of the local variables for that particular run, the Initial List of domain objects, and the Current Attribute. The Sort method will order the Initial List based on the Current Attribute and the Sorting Order of that Attribute. Then it produces a new list of domain objects. This new list then becomes the Current Sort Vector within the Current State Vector. This Current State Vector is then passed back to the Launch Watchdog method.

The Sort method employs a recursive quick sort algorithm that is able to call itself repeatedly until the list is completely ordered. The quick sort algorithm will partition the list into smaller pieces and then sort those pieces based on the Current Attribute. Each piece is further subdivided until the whole list is finally sorted. Aptly named, quick sort is one of the faster sorting methods available. Quick sort on average requires $O(N \ln N)$ time, while worst case it approaches $O(N^2)$ [FEL 97].

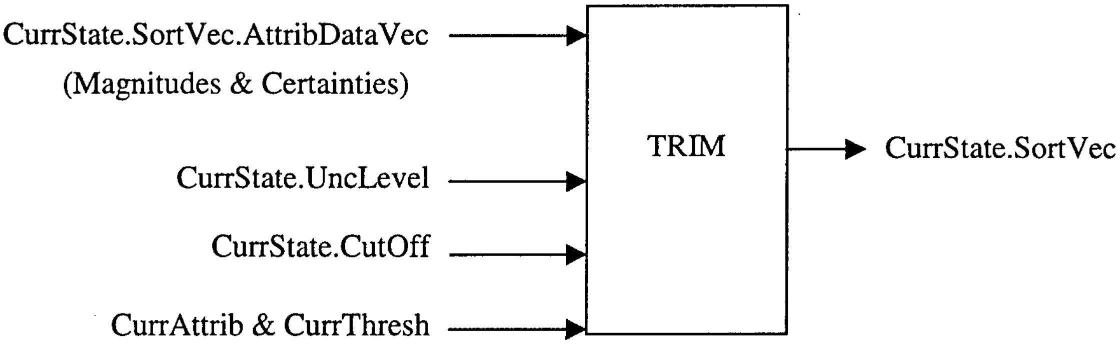
4.3.3 Adjust

At this point, several looping variables may need adjusting. These include the Current Uncertainty Level and the Percent From Top. However before these variables are adjusted, the program will attempt to find the top-most object. The program will attempt to find the top-most object on the sorted list that has not exceeded the Current Uncertainty Level. This Uncertainty Level value begins at 0% and is gradually adjusted until a top object is found or the Uncertainty Level has exceeded the User Threshold. Once a top object has been found, the program will attempt to establish a search interval. This search interval is a certain percent removed from the top element. This percent is stored in the Percent From Top variable in the Current State Vector. A Cutoff Value is then generated based on that Percent From Top. This Cutoff Value represents the lowest value of attribute data that the program is willing to accept. For example, if the top most element had a rating of 5, the program will first attempt to establish a search interval that is 10% from that top element and establish a Cutoff. In this case 10% of 5 is 0.5, so the Cutoff would be established 0.5 units removed from 5 which would be 4.5. Therefore, the search interval will be between 5.0 and 4.5. If the program returns and cannot find enough solutions, it will adjust the percentage by 10% more. The program will continue to process in this fashion until at least three solution objects can be found or if the Percent From Top exceeds 30%.

At this juncture, if Launch Watchdog has returned from the Trim method further adjustments can be made. Both the Current Uncertainty Level and Percent From Top can be adjusted if satisfying conditions have not been met in the Trim method. The satisfying conditions for the Trim method are that there are at least three objects in the solution set or that the Current Uncertainty Level is not greater than or equal to the Current Threshold. Of course, an adjustment of the Percent From Top will result in a correction in the Cutoff Value. If adjustments are necessary, the program will first adjust the Percent From Top and establish a new Cutoff Value. It will begin with 0% removed from to a maximum of 30% removed from the top object. If conditions are still not satisfactory, the Current Uncertainty Level will be adjusted until either satisfactory condition has been met. If the Uncertainty Level has reached the User Threshold, a failure condition exists. Launch Watchdog will notify the calling level of this failure.

If the Launch Watchdog has returned from a called level that has failed, it will attempt to adjust the Current Uncertainty Level of this Run Level in an attempt to relax the current levels constraints. It will keep adjusting until the Current Uncertainty Level reaches the User Threshold or if it has found at least three objects. If it fails to find a solution, Launch Watchdog will report this failure to its calling level. However, if the current Run Level is the first Run Level or the top-most levels, there are no other levels to report. Therefore, in an attempt to find the next-best solution, Launch Watchdog will push past the User Threshold here and set the User Threshold of the least most important attribute to zero. This will allow the program enough flexibility to find the next best solution. The alternative would be to keep asking the user to relax his/her User Threshold until a solution can be found. By implementing in the manner it is now, the program will always return a solution.

4.3.4 Trim



Now that a top object has been found and a Cutoff Value has been established, the program will call the Trim method. The Trim method will traverse the Current Sort Vector and try to find objects that are above the Cutoff and the Current Uncertainty Level. It will attempt to find at least three objects. If there are enough objects that satisfy these conditions, they are placed on the Trim Vector. The Trim Vector is then stored as the Trim Vector of the Current State Vector, which can be accessed by the Launch Watchdog method.

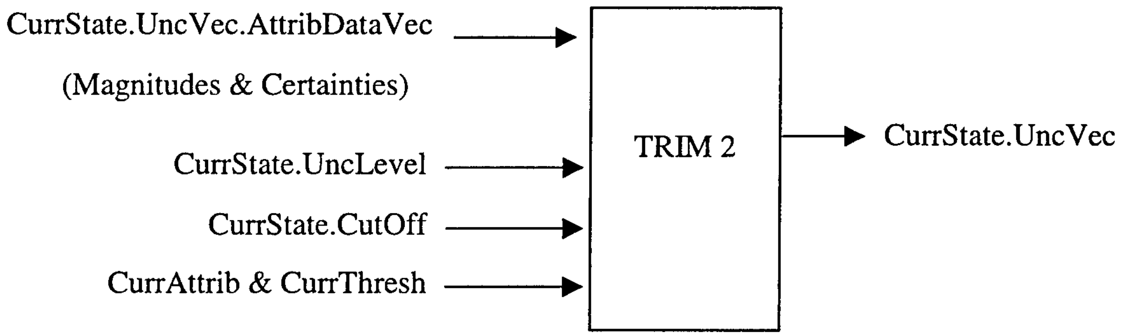
4.3.5 Satisfied So Far?

Finding the proper objects that should be put into the Trim Vector is not an easy task. The program will find objects that meet the satisfactory conditions with a "best effort" level of attempt. In other words, it is not guaranteed to find a solution vector. The satisfactory conditions that the program looks for are that there are at least three objects on the Trim Vector and that the Uncertainty Level is not greater than or equal to the User Threshold. It also looks to see if the Percent From Top value has not exceeded 30%. If these satisfactory conditions have been met, Launch Watchdog will proceed to the Trim 2 method. If not, Launch Watchdog will continue to check if there are any variables available to adjust.

4.3.6 Anything Left to Adjust?

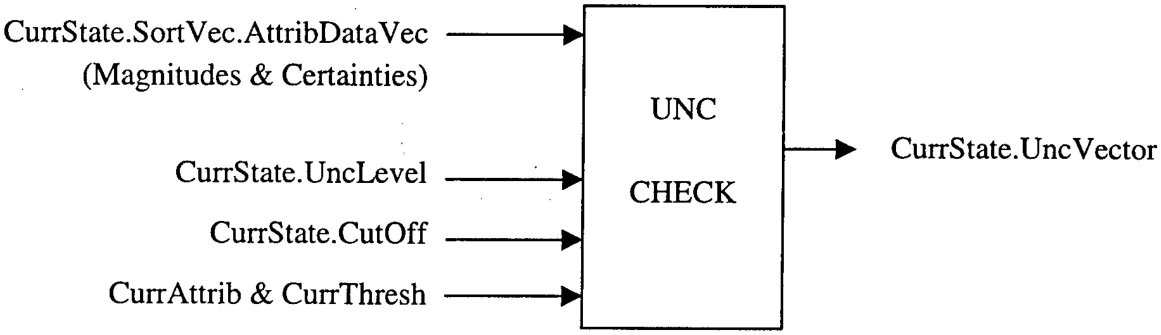
If, after the execution of the Trim method or after a recursive Launch Watchdog call, satisfactory conditions have not been met, the program will check to see if there are any looping parameters that can be adjusted. Again, these parameters are the Uncertainty Level and Percent From Top. If there are looping variables that have not exceeded their threshold, they are adjusted appropriately. If there isn't anything else left to adjust, a failure condition exists for this Run Level and Launch Watchdog will notify the calling level of this failure. Again, however, if the Current Run Level is the top-most level, the appropriate adjustments will be made as described in paragraph 4.3.3.

4.3.7 Trim 2



At this point, the objects that appear in the Uncertainty Vector are those objects whose values are too uncertain concerning previous attributes. Now, the program needs to uncover those objects in the Uncertainty Vector that would certainly fail the current user preference and eliminate them. Therefore, the purpose of the Trim 2 method is to purge those objects from the Uncertainty Vector that would certainly fail the Current Attribute and User Threshold. The program will take the Uncertainty Vector from the Current State Vector. It will then compare each and determine which objects certainly fail. Those objects that certainly fail are removed from the Uncertainty Vector. The Uncertainty Vector is then returned to the Launch Watchdog method in the Current State Vector.

4.3.8 Uncertainty Check



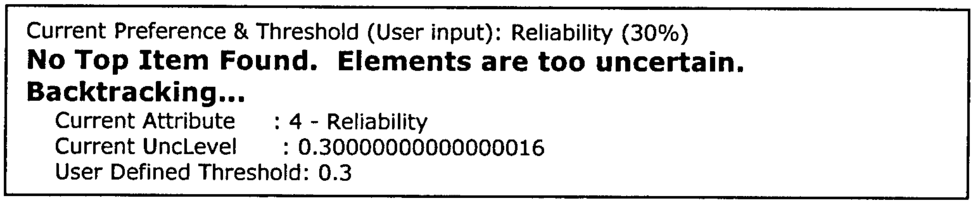
Now that the program has eliminated all values from the Uncertainty Vector that certainly fail, it will traverse the Sort Vector looking for objects that are too uncertain with regard to the current attribute choice. The program accesses the Sort Vector of the Current State Vector and determines which objects in the Sort Vector have exceeded the Uncertainty Level. These objects are added to the Uncertainty Vector. The Uncertainty Vector is then returned to the Launch Watchdog method in the Current State Vector.

4.3.9 Recursive Launch Watchdog Call

Finally, the program has produced an acceptable list of objects that have met the user preference. It is now ready to proceed to the next user preference. The current local variables are loaded into a New State Vector. This New State Vector is then passed to another Launch Watchdog method. Here, the program employs recursion to systematically call one of its methods repeatedly in order to solve a problem. When a "child" or a called level returns control, it will pass back a state vector that becomes the End Product.

4.3.9.1 Backtracking Verification

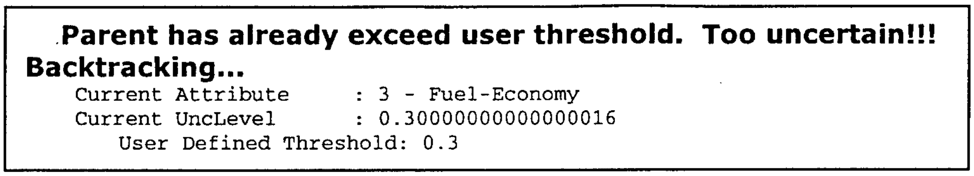
A concern related to proper recursion is proper backtracking. There are three conditions in which WATCHDOG will backtrack. First, if WATCHDOG cannot find a Top Object on the Sorted domain list, then all of the objects on the list are too uncertain. WATCHDOG will attempt to backtrack to a previous run level and try to adjust the previous run level's Uncertainty Level in an attempt to uncover more possible solutions for the Trim Vector. Figure 4-8 illustrates that this backtracking capability has been implemented.



```
Current Preference & Threshold (User input): Reliability (30%)
No Top Item Found.  Elements are too uncertain.
Backtracking...
   Current Attribute     : 4 - Reliability
   Current UncLevel      : 0.30000000000000016
   User Defined Threshold: 0.3
```

Figure 4-8: Backtracking Condition 1

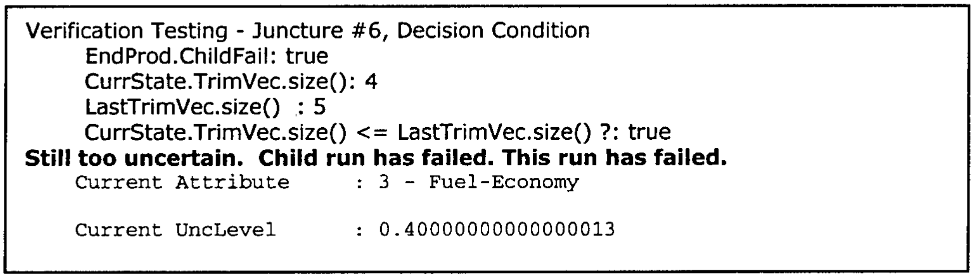
Second, if the program is returning from a child run and the child run has failed because it could not produce a Trim List, and if the Current Threshold level has already been reached, then the current run has also failed. WATCHDOG will need to backtrack to the level that called it. Figure 4-9 illustrates that this backtracking capability has been implemented.



```
Parent has already exceed user threshold.  Too uncertain!!!
Backtracking...
    Current Attribute       : 3 - Fuel-Economy
    Current UncLevel        : 0.30000000000000016
       User Defined Threshold: 0.3
```

Figure 4-9: Backtracking Condition 2

Third, if a child run has failed and the current run has tried to relax its Uncertainty Level in attempt to find possible solutions to no avail, then WATCHDOG will also need to backtrack to a previous run. Figure 4-10 illustrates that this backtracking capability has been implemented.



```
Verification Testing - Juncture #6, Decision Condition
      EndProd.ChildFail: true
      CurrState.TrimVec.size(): 4
      LastTrimVec.size() : 5
      CurrState.TrimVec.size() <= LastTrimVec.size() ?: true
Still too uncertain.  Child run has failed. This run has failed.
    Current Attribute       : 3 - Fuel-Economy

    Current UncLevel        : 0.40000000000000013
```

Figure 4-10: Backtracking Condition 3

4.3.10 Still Satisfied?

After control passes from a child program back to the parent, several conditions must be checked. The program will first check to see if the returning child has failed in finding a solution. If the current level has already reached its User Threshold, then the program knows immediately that the current level has failed as well. Therefore, if the returning child has failed, the program will next check the Current Uncertainty Level to see if the User Threshold has been reached. If so, then a failure will be declared and the program will backtrack to its calling program. In that instance, all objects of the domain data are too uncertain and the program will attempt to push past the User Threshold as described in paragraph 4.3.3

Moreover, if the child has failed but the Current Uncertainty Level is still within an acceptable range, the program will re-initialize the End Product's values. It can adjust the uncertainty level a bit more and search for a new top object as described in paragraph 4.3.3.

Conversely, if the child program has not failed and successfully returns a solution, the solution is passed back to the parent's calling program in the final State Vector called the End Product.

4.3.11 Quit Loop.

The Launch Watchdog method will keep asking the user for the next most important preference. It will keep cycling until the user has selected the Quit option upon which the program will return to the Main Menu, ready for another program execution.

4.3.12 Display Final Results

Finally, if an End Product has been returned from the original calling level, the program will declare a final success. It will then display a brief summary of all of the preferences that it checked the domain data against. The preference name, its associated User Threshold, and the Uncertainty Level will be displayed. Next, all of the objects in the Trim Vector of the End Product will be displayed in no particular order. Objects in the Trim Vector are those objects of the original data set that have satisfied the user's Attributes within the Uncertainty Levels. If the program was forced to push past the User

Thresholds, a warning flag will be displayed. Finally, the program will focus the user's attention on those objects in the Uncertainty Vector. These objects may have met the user's preferences; however, they have exceeded the Uncertainty Levels and should be given further consideration.

4.4 Verification of Major Processes and Program Features

To ensure that the coded implementation of the program worked properly, I examined the execution of the major processes of Get User Preference, Sort, Trim, Trim 2 and Uncertainty Check. I verified to see if these major processes were called under the proper circumstances and I also ensured that these processes executed correctly.

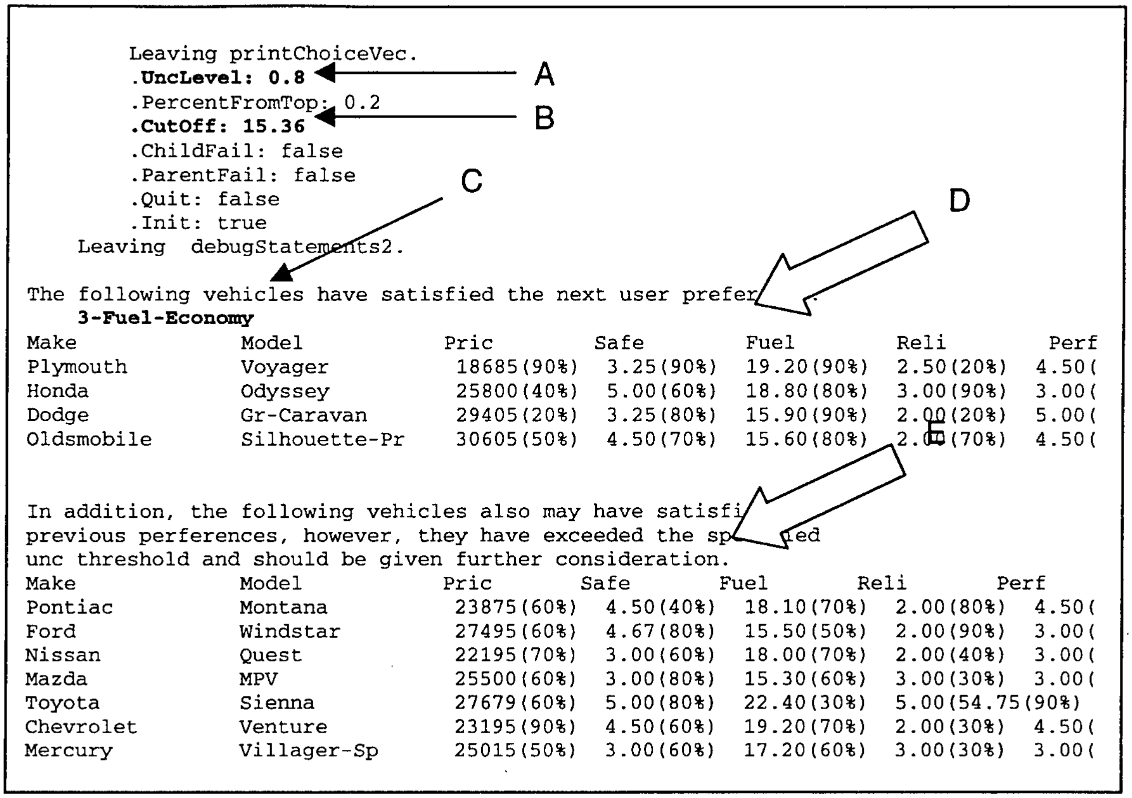
There were several items of concern that were scrutinized when verifying the results from the Trim, Trim2, & Unc Check methods. First, all items on the Trim Vector should be above the established cut-off point. Furthermore, these items should also be above the Uncertainty Level. As you can see from Figure 4.11, the current test subject preference is Fuel Economy. All of the objects on the first list, the Trim Vector, are above the established Cutoff Value of 15.36. Furthermore, these same objects have a Certainty Value equal to or above the Current Uncertainty Level of 80%.

The second item that is scrutinized is that the objects that appear on the Uncertainty List have Certainty Values below the Current Uncertainty Level. Also illustrated by Figure 4.11, it is apparent that the objects on the Uncertainty Vector have Fuel Economy Certainty Values that are below the Current Uncertainty Level of 80%. The Plymouth, Honda, Dodge, and the Oldsmobile, which are in the Trim Vector, do not appear in the Uncertainty Vector. Similarly, the objects in the Uncertainty Vector do not appear in the Trim Vector.

A third point of interest that was tested was that no element on the Trim Vector should appear on the Uncertainty Vector and vice versa. Figure 4.11 illustrates this point as well.

A fourth matter of concern was that there were not any duplicate entries on either the Trim Vector or the Uncertainty Vector. Again, Figure 4.11 demonstrates that this matter of entry duplication was not a problem with WATCHDOG.

Finally, a fifth matter that was examined was that objects that are certainly below the Cutoff Value. These objects have certainty values above the Uncertainty Value. Although it is not obvious from Figure 4.11, the Chrysler Town & Country is not listed on either list since its Fuel-Economy, 15.00 mpg, is below the Cutoff Vale of 15.36. Also, the Certainty Value for the Chrysler's Fuel-Economy is 80% which is equal to or above the Uncertainty Level of 80%.



```
        Leaving printChoiceVec.
        .UncLevel: 0.8
        .PercentFromTop: 0.2
        .CutOff: 15.36
        .ChildFail: false
        .ParentFail: false
        .Quit: false
        .Init: true
    Leaving  debugStatements2.

The following vehicles have satisfied the next user prefer    .
    3-Fuel-Economy
Make             Model            Pric        Safe        Fuel        Reli        Perf
Plymouth         Voyager           18685(90%)  3.25(90%)  19.20(90%)  2.50(20%)  4.50(
Honda            Odyssey           25800(40%)  5.00(60%)  18.80(80%)  3.00(90%)  3.00(
Dodge            Gr-Caravan        29405(20%)  3.25(80%)  15.90(90%)  2.00(20%)  5.00(
Oldsmobile       Silhouette-Pr     30605(50%)  4.50(70%)  15.60(80%)  2.00(70%)  4.50(


In addition, the following vehicles also may have satisfi
previous perferences, however, they have exceeded the sp   ied
unc threshold and should be given further consideration.
Make             Model            Pric       Safe       Fuel       Reli       Perf
Pontiac          Montana           23875(60%)  4.50(40%)  18.10(70%)  2.00(80%)  4.50(
Ford             Windstar          27495(60%)  4.67(80%)  15.50(50%)  2.00(90%)  3.00(
Nissan           Quest             22195(70%)  3.00(60%)  18.00(70%)  2.00(40%)  3.00(
Mazda            MPV               25500(60%)  3.00(80%)  15.30(60%)  3.00(30%)  3.00(
Toyota           Sienna            27679(60%)  5.00(80%)  22.40(30%)  5.00(54.75(90%)
Chevrolet        Venture           23195(90%)  4.50(60%)  19.20(70%)  2.00(30%)  4.50(
Mercury          Villager-Sp       25015(50%)  3.00(60%)  17.20(60%)  3.00(30%)  3.00(
```

Figure 4-11: Console Output of Interim Results

<u>Points of Interest</u>

A – Current Uncertainty Level – 80%

B – Established Cut-Off Value with respect to the Current User Preference – 15.36

C – Current User Preference – "Fuel Economy"

D – Trim List

E – Uncertainty List

4.5 Summary

This chapter focused on the design and implementation of the WATCHDOG Decision Support Tool. The data structures and the basic operations of the tool were discussed including the finer details of the major processes and program features that are integral to WATCHDOG. Along with the finer details of implementation, the decisions that went into the design of the overall tool were also highlighted. In particular, this chapter explained how the need for an interactive search system required recursive programming and the ability to backtrack. Furthermore, examples of the actual execution of the implemented system were also reviewed.

5 Testing & Results

"I have not failed. I've just found 10,000 ways that won't work."

Thomas Edison

5.1 Overview

The testing of the Watchdog Decision Support Tool consisted of two primary parts: Verification and Validation. The purpose of Verification testing was to ensure that the system performs in the manner it was programmed to do. Verification testing involved testing all paths of program logic as well as ensuring all major system requirements are being met. However, occasionally a program may be built in a manner in which it was specified but not in the manner it was intended. Therefore, the purpose of Validation testing was to ensure that the system performed in the original manner in which it was intended. For this program, Validation testing was more qualitative in nature and served to demonstrate the true value added of this tool. Validation testing involved using 10 test subjects, running three different scenarios with and without the decision support tool, and collecting their opinions in order to evaluate the effectiveness of WATCHDOG.

5.2 Verification Testing

The goal of Verification testing of WATCHDOG was to guarantee that the system performed in the manner in which it was programmed to do. Verification involved testing the main menu sub-system, testing all paths of logic, and testing the major processes or methods. All of this testing was performed to ensure that the system design requirements had been fulfilled and to guarantee that the program operated in the manner it was programmed to operate.

The main menu testing involved ensuring that the four menu options operated properly. They were checked to see if the appropriate option was actually executed, that the option executed properly, and that the program control returned to the main menu. All menu options performed properly.

WATCHDOG possessed twenty-four separate paths of logic. Conditions that would induce particular paths were initiated and eventually all paths were tested. All twenty-four paths were tested with successful results. However, one path was overlooked initially: when no answers could be found to satisfy the user's preferences. This phenomenon was discovered during Test One of the Validation Testing. Subsequently, the code was corrected and then re-verified.

Finally, Verification Testing involved testing the functionalities of the major processes and features. I tested the major processes of Get User Preferences, Sort, Trim, Trim2 and Uncertainty Check all executed under the correct circumstances and that they performed properly. I also examined the recursion and backtracking capabilities of WATCHDOG during this phase of Verification testing.

5.3 Validation Testing

The purpose of Validation Testing was to demonstrate the benefits that the WATCHDOG Decision Support Tool can provide. Primarily, I wanted to show that WATCHDOG had value added when sorting through a large amount of data with multiple attributes. Validation Testing consisted of two separate tests. Both tests consisted of 10 test subjects evaluating a number of different scenarios of domain data, with and without the aid of WATCHDOG. A survey of questions was given to each test subject. Their responses were aggregated and analyzed in order to determine WATCHDOG's effectiveness.

5.3.1 Test One

In Test One, the test subjects were asked to evaluate three different minivan data scenarios, with and without the aid of the WATCHDOG tool. They were divided into two groups: Group A and Group B. Both groups were given the same set of three minivan data scenarios. The data in each scenario was slightly different from the other scenarios. The only difference between the groups was the order in which they evaluated the minivan data. By dividing the subjects into two groups, it was my intention to show that the subjects from Group A would gain more benefit from using the WATCHDOG tool than the subjects from Group B.

It was the job of the test subjects to figure out which attribute or attributes were important to them, then make a decision on which minivan to buy based on those attributes. The subjects would later use WATCHDOG to help them to sort through minivan data and try to find possible solutions based on the attributes that they would tell WATCHDOG were important. WATCHDOG would ask the test subject for a series of attributes and uncertainty thresholds, one at a time. These uncertainty thresholds were the lowest possible certainty that the subject would be willing to accept for a particular attribute. Each time the tool accepted an attribute/threshold pair from the test subject, WATCHDOG would attempt to sift through the minivan data and try to find solutions based on that pair. Once it found an interim solution, WATCHDOG would ask the test subject for the next most important attribute. Please note that for the first test, WATCHDOG did not display any solutions until the test subject was ready to quit. The reason for this was that objects on the different lists may change due to the possible relaxing of thresholds. Once the test subject entered all of the attributes that were important, he/she could display the final results by quitting.

Group A was asked to evaluate each scenario one at a time without the aid of the WATCHDOG tool, and decide which minivans they would buy. They would select one minivan from each of the three separate scenarios. Next, they were asked to re-evaluate each scenario but with the aid of the WATCHDOG tool and then decide which minivan they would buy.

Group B was tested slightly differently. Group B was asked to evaluate the first scenario without the WATCHDOG tool and decide which minivans they would buy. After that, instead of evaluating the next scenario right away, they were asked to re-evaluate the first scenario with the aid of the WATCHDOG tool. After Group B evaluated the first scenario both with and without the aid of the WATCHDOG tool, they proceeded to evaluate the other two scenarios in the same manner.

Furthermore, each test subject was given a test booklet to answer a few questions while they were evaluating the scenarios. These questions helped to evaluate the effectiveness of the WATCHDOG decision support tool as a whole.

Additionally, because Group B subjects evaluated each scenario manually and then re-evaluate the same scenario with the aid of the tool back-to-back, it was my expectation that they should be able to develop a methodology of evaluation quicker than the Group A subjects. I expect that this structured methodology would put Group B subjects in a better frame of mind in order to assess the next set of scenario data. Therefore, I expected that the benefits of the WATCHDOG tool would not be as apparent to Group B as it would be for Group A who would not have the benefit of learning this methodology as they went along.

5.3.1.1 Test One Domain Data [CON99], [Old99]

The WATCHDOG Decision Support Tool is designed to handle a number of different domains. However, for the purposes of the Validation Testing, the domain was limited to something that is familiar to a wide range of people: minivans. Data from *Consumer Reports* and *Popular Mechanics* magazines was used in the scenario data. In these magazines, twelve minivans, each with front-wheel drive, a V6-engine and four doors were comparison tested. This minivan comparison test data was used to populate the minivan data for the validation testing scenarios.

For Test One, I limited the number of attributes to five. I realized that there are a number of other attributes that a person can take into consideration when purchasing a vehicle. However, for the most part, I believed that people would only consider a handful of factors when making a decision. After deciding on the number of attributes, I turned my attention to choosing which five attributes to include in the domain data.

The overall vehicle ratings from *Consumer Reports* were based on performance, comfort, convenience, safety, and fuel economy. Although reliability was not a part of this score, it does affect whether a vehicle is recommended. Comfort and convenience information was not readily available and furthermore seemed too qualitative compared to the other attributes. Pricing information was available and should always be considered when buying a car. As a result, I selected what seemed to be the five most appropriate attributes of Price, Safety, Fuel Economy, Reliability, and Performance. The following are brief descriptions of each attribute:

PRICE - Pricing data is measured in dollars and is based on the base price of the minivan model.

FUEL ECONOMY - Fuel Economy data is measured in miles per gallon (mpg). It is not specific to city or highway performance but rather the actual fuel economy of the test vehicles used in *Popular Mechanics* comparison tests. The Fuel Economy rating was an average taken from the suburb driving tests as well as test track driving tests.

SAFETY - The Safety rating is based on a scale of 1 (Worse) to 5 (Better). The safety rating for minivans was determined by a combination of factors. These factors include analysis of front and side crash tests as conducted by the government's National Highway Traffic Safety Administration (NHTSA) and the private Insurance Institute for Highway Safety (IIHS). The presence of design features such as safety belts, side air-bags, anti-lock braking systems (ABSs), and traction aids also contributed to a vehicle's overall safety rating. The actual Safety rating was an amalgam of all of these factors.

RELIABILITY - The Reliability rating is based on a scale of 1 (Worse) to 5 (Better). The reliability of a car is actually a forecast of how well a car will hold up based on data gathered from previous years of cars of the same make. The frequencies of repairs of sixteen separate "trouble spots" such as the engine, the transmission, and the suspension, were evaluated and rated. These individual ratings help to determine a Reliability Summary of the vehicle as a whole.

PERFORMANCE - The Performance rating is based on a scale of 1 (Worse) to 5 (Better). It represents a combination of factors including a vehicle's braking ability, handling, obstacle avoidance, horsepower, and acceleration.

Table 5-1 summarizes the characteristics associated with the individual minivan attributes. The Min/Max column represents whether the attribute is a minimizing or maximizing attribute. Minimizing attributes are attributes where the smaller the magnitude the better (e.s. Price – It is better to have the lower priced vehicle). Maximizing attributes are just the opposite (e.s. Reliability – A vehicle with a higher reliability rating is better and therefore more reliable.)

Table 5-1: Test One Minivan Attribute Summary

Attribute	Max/Min	Units
Price	↓	$
Safety	↑	R:1-5
Fuel-Economy	↑	mpg
Reliability	↑	R:1-5
Performance	↑	R:1-5

5.3.1.2 Test One Scenarios

There were three different test scenarios used for Test One. In the first scenario the data for only one minivan was altered so that the scenario had one clear winner, the Toyota Sienna. In the second scenario, four cars were given certainty values between 70-90% while the remaining cars had certainty values of less 60%. A clear distinction between certain and uncertain minivans was made evident. Of the certain minivans, the Honda Odyssey should have appeared most favorable. The third scenario had completely random certainties assigned to all of the data elements. There were no clear winners and finding a solution would require WATCHDOG to backtrack quite a bit.

5.3.1.3 Test One Results

For the first scenario without the tool, 100% of the test subjects picked the same car, Toyota Sienna, as expected. A wide variety of reasons and personal algorithms were used to select the minivan. Only one test subject was aware of the need to get information that is more confident on the other vehicles.

For the second scenario without the tool, 80% picked the Honda while 20% picked the Toyota. Again, a wide variety of reasons were used to select the minivans.

For the third scenario without the tool, various minivans were picked with no clear winners, also as expected. Ultimately, the test subjects' personal preferences were the deciding factors that distinguished between minivans.

For the first scenario with the tool, 100% picked the Toyota. Price, Safety, and Reliability were the top attributes that helped the test subjects to decide among the minivans of the first scenario. A majority of the test subjects, 80%, felt comfortable about their decision with the use of the WATCHDOG tool.

For the second Scenario with the tool, 90% picked the Honda while 10% picked the Toyota. Price, Reliability, & Safety were the top attributes that helped the test subjects to decide among the minivans of the second scenario. As with the first scenario, a majority of the test subjects, 80%, felt comfortable about their decision with the use of the WATCHDOG tool.

For the third scenario with the tool, a wide variety of minivans were selected (40% Toyota, 20% Honda, 20% Dodge, 10% Plymouth, 10% Chevy). Price, Reliability, & Safety were the top attributes that helped the test subjects to decide among the minivans of the third scenario. Although a majority of the test subjects felt comfortable about their decision with the use of the WATCHDOG tool, the percentage dropped from 80% to 60%.

Overall, 20% of the test subjects said that WATCHDOG was very useful, 60% said that it was useful, while only 20% said that it was of average use. When asked if they would use WATCHDOG again if they were in the market to buy minivan, 90% said they would use the tool again. Additionally, on a scale between 1 and 5, 30% said they were very satisfied (5) with the tool, while 30% said they were satisfied (4). Also, 20% said they had an average satisfaction level (3) with the tool while 20% said they were not satisfied (2) at all. With respect to ease of use, 30% of the test subjects felt the tool was very easy (5) to use, 40% felt it was easy to use (4), while 30% felt it had an average level of ease (3). The best liked aspects of WATCHDOG were noted to be its speed, ease of use, and the promise that it has of focusing the test subject on uncertain data. The least liked aspects of the tool were the seemingly counter-intuitive processing, the system crashing on the third scenario, the awkward interface, and no way to specify ranges for magnitude and confidence.

5.3.1.4 Test One Trends

I observed several trends from the results of Test One. First, several subjects changed their answers after using the WATCHDOG tool. Second, as the data sets became more complex, the comfort level of the test subjects with the tool decreased. Third, trends in the survey results indicated that the test subjects might not have been focusing on the certainty values of the data domain attributes.

During the test, one subject changed his decision after using the tool in the second scenario. In a like fashion, three subjects also changed their decision in the third scenario. Out of these four subjects, three of them felt comfortable with their decision after using WATCHDOG. This data would seem to indicate that as the data becomes more complex, the test subjects relied more on the tool. I conjecture that the subjects used WATCHDOG to help sort through the complex data and consequently changed their answers based on the results that the tool provided.

However, the test results also showed indications that as the data became more complex, the test subjects were less comfortable in using WATCHDOG. This is contradictory to my hypothesis. This decrease in comfort level with the tool may have been attributed to one of two factors. The first factor may have been that the test subjects lost confidence in the tool after it crashed in the third scenario. Seven test subjects had WATCHDOG halt processing on them while they were evaluating the third scenario with the tool. Of these seven, three subjects indicated that they were more comfortable about their choice without the use of the tool. Although it was designed to backtrack if it could not find a solution, WATCHDOG would terminate processing if it reached the test subject's thresholds. In many cases, test subjects tended to select high uncertainty thresholds that did not allow for much leeway in finding a possible solution. This contingency was not expected when designing the system.

The second factor that may be attributed to the decrease in comfort level with WATCHDOG is the counter-intuitive feel associated with the tool's decision-making. WATCHDOG's decisions may have seemed counter-intuitive since the program will first try to eliminate those items whose attributes are too uncertain before sorting by magnitude. As discussed in Chapter 3, I believed that it was more important that the tool should address the certainty levels before considering the magnitude of the data attributes. This appeared to be contrary to the way many of the test subjects approached decision-making. Many test subjects would first select candidates based on magnitude. Of those that passed this first test, the subjects tended to select the least uncertain item. Because of this fundamental difference in the decision-making process, WATCHDOG appeared to be counter-intuitive. Consequently, this also may be a reason why test subjects became less comfortable with the tool.

The third trend that I noticed was that the test subjects did not necessarily focus on the pertinent elements of uncertain data. Even after WATCHDOG displayed its final results with explicit statements that indicated which minivans needed more consideration, many test subject disregarded the elements on the Uncertainty List. It was my intention that with the tool, the test subjects would focus on the alternative solutions provided in the Uncertainty List and converge on the relevant issues (uncertain data). Furthermore, I had hoped that the subjects have realized that by possibly dedicating more resources to alleviating some of this uncertainty, they might reap greater rewards and uncover a better minivan. Instead, some of the test subjects indicated that they made their decision based on the elements that WATCHDOG provided in the Trim List. I can attribute this trend to the test subjects not fully understanding the purpose of WATCHDOG. Also, they may have not understood the difference between the two lists, or I may not have fully elaborated how the decision support tool was supposed to be used. Moreover, the survey questions that asked the test subjects what information would they want to know more about may have been too ambiguous.

Finally, the responses from the survey seemed to indicate that to Group B, there was less value added when using the WATCHDOG tool when compared to Group A, as expected. However, it should be noted that the number of test subjects was not statistically significant enough to draw a hard conclusion.

5.3.2 Test Two

After evaluating the results from Test One, it became clear that the use of the WATCHDOG tool did not produce the desired results and brought to light minor flaws with the tool. The WATCHDOG tool was subsequently upgraded and new survey questions were polished. As a result, Test Two was created and administered.

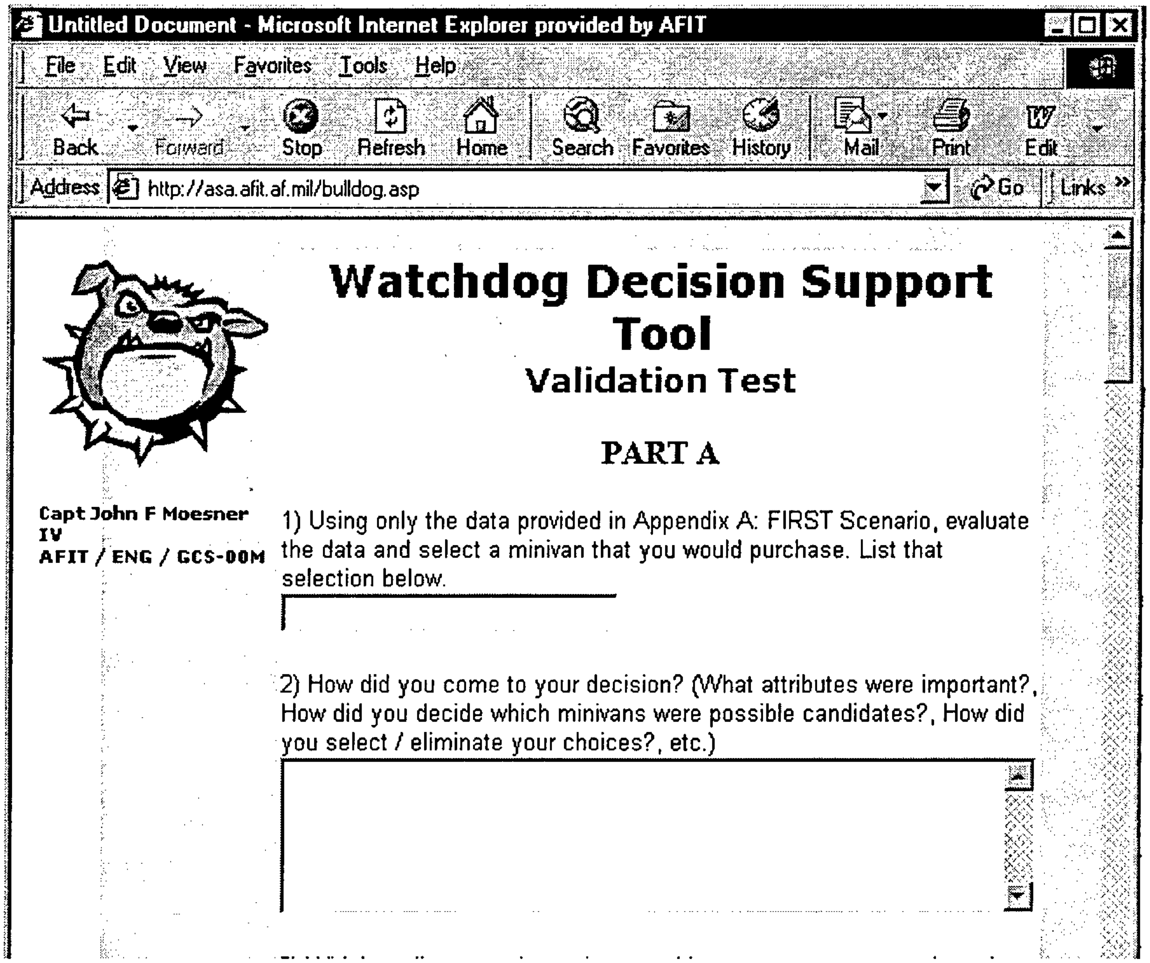
In the previous test, some of the test subjects encountered abnormal system terminations when executing the third scenario. The reason that this problem occurred was that the tool would backtrack as far as it could, but it still could not produce a solution. A graceful exit was not provided. The user-defined thresholds were too constraining for the given data. The domain data may have been too uncertain for the liking of the test subjects. Therefore, I added a new feature that allowed the tool to find a solution with a best-effort level of attempt. WATCHDOG was now able to exceed the test subject-defined threshold in order to find the next best solution. By implementing this new feature, I had hoped to keep the test subjects from losing faith in the tool.

Furthermore, in order to promote a better understanding of WATCHDOG's role as an aid to decision-making, I drafted a more detailed instruction booklet that clarified the tool's purpose. Also, I installed extra program output statements within WATCHDOG that would better elaborate what the program was doing as well as noting to the test subject the significance of the displayed final results. In addition, the tool would now display interim results so the test subjects could better follow WATCHDOG's progress. These results were accompanied by a cautionary warning that the displayed interim results may change due to alteration of lists from backtracking.

As with Test One, ten test subjects were asked to evaluate three different minivan data scenarios, with and without the aid of the WATCHDOG tool. Again, it was their job to figure out which of the attribute or attributes were important and then make a decision on which minivan to buy based on those attributes. However, a new and improved attribute list was implemented. Many more attributes were added in order to make the data set more complex and illustrate WATCHDOG's true potential.

Different from Test One, the test subjects were not divided into Groups A & B. Also different, the subjects were asked to evaluate two instead of three different minivan data scenarios, with and without the aid of the WATCHDOG tool. Again, the data in each scenario was slightly different from each other, one scenario being quite easier than the other but both considerably more complex than the scenarios from Test One.

Moreover, a few survey questions were edited in order to eliminate any possible points of confusion that were present in Test One's survey. Specific questions on the certainty value and what it represents were answered. In addition, clear descriptions of the attributes and their significance were also addressed. With the aid of Lt Phillip Polk, the survey was now web-based (Figure 5-1) which made it easier for the test subjects to enter their answers. It facilitated compiling and aggregating end results for the test administrator. All of these extra nuances were added to help gather better test data.



Watchdog Decision Support Tool

Validation Test

PART A

Capt John F Moesner IV
AFIT / ENG / GCS-00M

1) Using only the data provided in Appendix A: FIRST Scenario, evaluate the data and select a minivan that you would purchase. List that selection below.

2) How did you come to your decision? (What attributes were important?, How did you decide which minivans were possible candidates?, How did you select / eliminate your choices?, etc.)

Figure 5-1: Web-based Survey for Validation Test

5.3.2.1 Test Two Domain Data [CON99], [Old99]

After analyzing the results from Test One, I came to the conclusion that five attributes may not have been an appropriate number of attributes. Some of the test subjects noted that they could assimilate most of the information and evaluate the data without any aid. Therefore, for Test Two, I wanted to incorporate a larger number of attributes. In Test Two, I included the first four attributes of Price, Safety, Fuel Economy, and Reliability. In addition to these four attributes from the first test, I further broke down Performance into Braking Ability, Acceleration, and Horsepower. Furthermore, I added the following attributes: Towing, Sound, Engine Displacement, Satisfaction, Depreciation, Buying Experience, Cargo, Warranty Rating, and Overall Rating for a total of 16 attributes. With these new attributes, I intended to show that the tool would be able help the test subject sort through even more complex amounts of data. The following are brief descriptions of the new attributes:

BRAKING - Braking data represents the stopping distance, measured in feet, of a vehicle as it travels from a rate of 60 to 0 mph.

ACCELERATION - Acceleration data represents the time, measured in seconds, that the vehicle takes to accelerate from 0 to 60 mph.

HANDLING - Handling data represents the speed, measured in miles per hour, that the vehicle takes to negotiate a 525 ft. slalom course of eight cones, separated 75 feet apart.

TOWING - Towing data represents the maximum towing capacity of the vehicle measured in pounds.

SOUND - Sound data represents the noise level, measured in decibels, of the vehicle as the vehicle travels at 60 miles per hour.

ENGINE DISPLACEMENT - Engine Displacement data represents the volume of space, measured in liters, that the cylinders of the vehicle displace.

SATISFACTION - The Satisfaction rating is based on a scale of 1 (Worse) to 5 (Better). It is a forecast based on responses from the 1998 Consumer Report's Annual Questionnaire on whether the reader would buy the same car again. Vehicles with no data available were given default values of 3, while new vehicles were given values of 4.

DEPRECIATION - The Depreciation rating is based on a scale of 1 (Worse) to 5 (Better). It represents *Consumer Report's* prediction of how well a model will keep its value. Vehicles with no data available were given default values of 3, while new vehicles were given values of 4. (FYI: The average depreciation rating for all vehicles tested [not just minivans] was about 1.7.)

BUYING EXPERIENCE - Buying Experience is measured in the percent of customers satisfied with a dealer. Based on answers given on questionnaires of buyers of new cars from 1997 & 1998, buying experience shows the overall satisfaction customers have with the brand, but also with the various models within a brand.

Data was adjusted for the buyer's age, since older buyers tend to be less critical than younger ones.

CARGO CAPACITY - Cargo data represents the maximum cargo space, measured in cubic feet, that a vehicle can hold.

WARRANTY RATING - The Warranty rating is based on a scale of 1 (Worse) to 5 (Better). It represents the value that can be associated with the overall warranty packages of the vehicle, independent of dealer. Data in this category did not come from either *Consumer Reports* or *Popular Mechanics* but was fabricated solely for the purpose of this test.

OVERALL RATING - The Overall rating is based on a scale of 1 (Worse) to 5 (Better). It represents the summary rating of the vehicle as a whole as provided by *Consumer Reports*.

Table 5-2 summarizes the characteristics associated with the individual minivan attributes. The Min/Max column represents whether the attribute is a minimizing or maximizing attribute. Minimizing attributes (↓) are attributes where the smaller the magnitude the better (i.e. Price – It is better to have the lower priced vehicle). Maximizing attributes (↑) are just the opposite (i.e. Reliability – A vehicle with a higher reliability rating is better and therefore more reliable).

Table 5-2: Test Two Minivan Attribute Summary

Attribute	Max/Min	Units
Price	↓	$
Safety	↑	R:1-5
Fuel-Economy	↑	mpg
Reliability	↑	R:1-5
Braking, 60 – 0 mph	↓	ft.
Acceleration, 0 – 60 mph	↓	sec.
Handling, 525 ft. slalom	↑	mph
Towing	↑	lbs
Sound, @ 60 mph	↓	dBA
Engine-Disp	↑	liters
Satisfaction	↑	R:1-5
Depreciation	↑	R:1-5
Buying-Exp	↑	%
Cargo	↑	cu. ft.
Warranty	↑	R:1-5
Overall	↑	R:1-5

5.3.2.2 Test Two Scenarios

There were only two scenarios for Test Two. One was relatively easy while the other was more difficult. The first scenario had 5 minivans with 80-90% certainty with no clear favorable solution. The remaining 7 minivans had a 20-50% certainty range, resulting in another clear distinction between certain and uncertain cars. The second scenario had all 12 minivans with random certainties ranging from 20-90%. There was no clear winner.

5.3.2.3 Test Two Results

For the first scenario without the tool, 90% of the test subjects selected the Toyota while 10% selected the Honda. Not surprisingly, various approaches were used to select the minivan.

For the second scenario without the tool, 80% selected the Toyota, 10% selected the Plymouth, while 10% selected the Chevy. Again, a wide variety of reasons were used to select the minivan.

For the first scenario with the tool, a wide variety of minivans was selected (40% Toyota, 20% picked Plymouth, 10% Pontiac, 10% Honda, and 10% Nissan). Price, Safety, and Overall were the top attributes that helped the test subjects to decide among the minivans of the first scenario. A clear majority of the test subjects was satisfied with their answer with the aid of the WATCHDOG tool. On a scale of 1 to 5, 50% were very satisfied (5), 30% were satisfied (4), while 20% had rated the tool as average (3).

For the second scenario with the tool, a wide variety of minivans was also selected (40% Toyota, 30% Chevy, 10% Plymouth, 10% Honda, and 10% Dodge). Price, Safety, and Overall were the top attributes that helped the test subjects to decide among the minivans of the first scenario. Again, a clear majority of the test subjects was satisfied with their answer with the aid of the WATCHDOG tool. On a scale from 1 to 5, 50% felt very satisfied (5), 40% felt satisfied (4), while only 10% had an average level of satisfaction (3).

Overall, on a scale from 1 to 5, 50% of the test subjects said that WATCHDOG was very useful (5) and 50% said that it was useful (4). When asked if they would use WATCHDOG again if they were in the market to buy minivan, 100% said they would use the tool again. Additionally, on a scale from 1 to 5, 30% said they were very satisfied (5) with the tool while 70% said they were satisfied (4). With respect to ease of use, on a scale from 1 to 5, 40% felt the tool was very easy to use (5) and 60% felt it was easy to use (4). The best liked aspects of WATCHDOG were noted to be that the tool was quick and easy, reduced the solution space to a manageable set, and picked results that the subjects may not have considered. The least liked aspects of the tool were the perceived counterintuitive processing, the desire for better visualization or a Graphical User Interface (GUI), and the lack of floors/ceilings for the magnitudes.

5.3.2.4 Test Two Trends

I observed several trends from the results from Test Two. First, several subjects changed their answers after using the WATCHDOG tool. Second, as the data sets became more complex, the comfort level of the test subjects with the tool increased. Third, trends in the survey results indicated that the test subjects were focusing on the certainty values of the data domain attributes.

Similar to Test One, several test subjects in Test Two changed their decisions after using WATCHDOG. Fives subjects changed their decisions after using the tool in the first scenario and four subjects changed their decisions in the second scenario. All nine of these subjects felt either satisfied or very satisfied with their decisions after using the tool. This data would seem to indicate that as the data became more complex, the test subjects relied more on the tool. I again conjecture that the subjects used WATCHDOG to help sort through the complex data and consequently changed their answers based on the results that the tool provided.

Furthermore, the test results also showed indications that as the data became more complex, the test subjects became more comfortable in using WATCHDOG. This is contradictory to results from Test One but is aligned with my original hypothesis. I believe that this increase in comfort level with the tool can be attributed to three factors. With the new best-effort backtracking function in place, the system no longer terminated abruptly. I believe that this was the first factor which helped raise the confidence in the tool with the eight test subjects, which were repeat subjects from Test One. Also, the improved instruction booklet was the second factor that may have increased the test subjects' confidence in the tool. The additional system information as well as detailed data descriptions helped the test subjects better understand the purpose of the tool and how WATCHDOG finds solutions. The third factor that may be attributed to the increase in confidence in the WATCHDOG tool was the increase in the number of attributes. By increasing the number of attributes from five to sixteen, the process of choosing the "best" minivan was made more difficult. Although there were more attributes to qualify the minivans by, the test subjects had to sort through more data which complicated the decision-making process. Because the data set was so much larger than in the previous test, the benefits of letting WATCHDOG sort through the data became apparent.

The third trend that the results indicated was opposite of the results from Test One. This time, test subjects did indeed focus their attention on the pertinent elements of uncertain data. Improved understanding of the WATCHDOG tool that was gathered from the improved documentation as well as improved program output statements can be attributed to guiding the attention of the subjects appropriately. Furthermore, better-quality survey questions also help to direct the test subjects to the uncertainty issues. Eight of the ten subjects stated that they would gather more data concerning minivans that appeared on the Uncertainty List in order to make a better-informed decision.

Another general observation that again was noticeable was the perceived counter-intuitive nature of WATCHDOG. Although the data trends indicated that the test subjects were more comfortable with the tool than during Test One, many were perplexed by some of the results that the tool presented. A prime example that occurred anytime a test subject selected Price as one of the discriminating attributes. Because the Toyota Sienna's Price had a high magnitude with a high certainty, WATCHDOG would summarily eliminate the Toyota from the list of potential candidates. This surprised many test subjects. Again, this relates back to how the tool sorts and processes data. It looks at certainty first, then magnitude second. Apparently this is converse to how people in general make a selection from a number of choices.

5.4 Summary

To summarize, the testing of the WATCHDOG decision support tool consisted of two parts: Verification Testing and Validation Testing. The purpose of Verification testing was to ensure that the system performs in the manner it was programmed to do. Conversely, the purpose of Validation testing was to ensure that the system performs in the original manner in which it was intended. Verification testing involved testing all paths of logic within the WATCHDOG program while Validation testing consisted of evaluating decision-making scenarios with and without the decision support tool.

Validation testing was qualitative in nature and was designed to test for true benefits of this tool. Validation testing involved two separate tests: Test One and Test Two. Test One used 10 test subjects evaluating three different data scenarios of minivan data and selecting a minivan from each scenario. The test subjects ran each of the three different scenarios with and without the aid of the decision support tool.

Consequently, Test Two was created and administered. The expected results from Test One were not forthcoming for a number of reasons. The WATCHDOG tool was upgraded, the test questions were cleaned-up, and the system description documentation was enhanced.

Test Two was very similar to Test One. It had 10 test subjects but only two different data scenarios. In addition, the minivan data evaluated in Test Two was significantly more complex than the data evaluated in Test one. In both tests, all subjects were asked to fill out a survey of 20-30 questions. These questions asked the subjects what choices they made and how they made them. The results of these surveys were then used to make an overall evaluation of the effectiveness of the WATCHDOG tool.

5.4.1 Testing Conclusions

From the Verification testing, I have learned that the WATCHDOG program works in the manner in which I had intended and as I had programmed it. From the Validation testing, I have learned that this tool is a valid method that can be used to focus the attention of the decision-maker on uncertain data.

Validation Testing required two separate tests. In Test One, test subjects did not notice the benefits of the tool. This was attributed to several possible reasons. As a result, the program was upgraded and any areas of ambiguity were cleared-up. Importantly, I realized that the small size of the data set masked the true utility of the tool. Thus, the number of attribute data was increased from five attributes to sixteen attributes per minivan. Test Two was administered.. The survey data from Test Two illustrates that as the data set became larger and more complex it became more difficult to evaluate the minivan data. The benefits of using WATCHDOG over manually sifting through the data set were highlighted as the test subjects ran through the scenarios. Furthermore, the decision support tool was able to focus the attention of the test subjects on the appropriate uncertain data. These two Validation tests would seem to indicate that the more complex the data set, the more value added the tool has since the decision-making becomes more difficult.

Regarding the counter-intuitive dilemma, I have demonstrated that the test subjects are more appreciative of WATCHDOG with a complex problem. However, many subjects commented on and some disagreed with how the tool sorted and eliminated candidates. It did not operate exactly how they expected. The test subjects used varying and different approaches to decision-making but survey results as well as interviews with test subjects indicated that people selected candidates by magnitude first and then eliminated those candidates that were too uncertain. Most subjects seemed to concentrate on magnitude first, uncertainty second, which is opposite of how WATCHDOG processes. I propose that this is not a bad thing.

As I stated in Chapter 3, WATCHDOG is a decision support tool. It is an aid for the decision-maker. It should not make decisions for the decision-makers. The tool should focus test subjects on uncertain data. Then with a little more effort in resolving some of the uncertainty, the test subject might find a better alternative solution. By not concentrating on the magnitude, WATCHDOG can support the decision-maker in a complementary manner. The benefit of the tool is that it will not be hindered by the human bias toward magnitude. The tool will intentionally weed out items deemed too uncertain. These items may be objects that would normally be overlooked by the decision-maker. One alternative would be to have the decision-maker make a decision first, then use WATCHDOG to possibly uncover items that may have been neglected otherwise. By using the tool in this manner, the decision-maker may be able to capitalize on both human intuition as well as the decision support tool in a synergistic manner. Moreover, I believe that if the decision-maker used WATCHDOG on a regular basis, he would see the usefulness of the tool as productivity went up. The decision-maker would be trained how to use the tool to its fullest through everyday use. However, we would not be able to see this gain without extensive experimentation.

6 Summary, Conclusion, & Recommendations

"May every young scientist remember... and not fail to keep his eyes open for the possibility that an irritating failure of his apparatus to give consistent results may once or twice in a lifetime conceal an important discovery."

Patrick Blackett

6.1 Research Summary

The objective of this thesis research effort was to develop a method of focusing the attention of the user on uncertain information. While there are several methods available for dealing with information overload that could have been investigated, this thesis concentrated on developing a tool that would reduce a solution space into a manageable set of choices. By decreasing the amount of non-essential information, we can increase the proportions of useful information thereby focusing the decision-maker's attention on what is really important.

In order to meet this objective, I set out to create a decision support tool that would guide the decision-maker to important "centers of gravity" or "watchspaces." Entitled the WATCHDOG Decision Support Tool, this tool was not meant to take the place of a decision-maker but rather to aid the decision-maker in sorting large amounts of uncertain information and quickly orienting this data into something more meaningful. Furthermore, WATCHDOG is the first step in establishing a universal framework for handling data and its associated uncertainty across multiple domains.

I designed and implemented this WATCHDOG tool. The tool operates by processing large amounts of domain data for the decision-maker. Each object of the domain data has a number of characteristics and attributes. Each attribute has a magnitude and a corresponding certainty value. The certainty value can be envisioned as the amount of confidence that can be associated with its magnitude. In addition, the certainty also relates to the proximity of the indicated value to the true value. WATCHDOG will sort through the domain data on a "best-attempt" level of effort and try to find possible solutions based on the attributes that the decision-maker tells it are important. It will ask the decision-maker for a series of attributes and uncertainty thresholds, one at a time. These uncertainty thresholds are the lowest possible certainty that the decision-maker is willing to accept for a particular attribute. Each time it accepts an attribute/threshold pair from the decision-maker, WATCHDOG will attempt to sift through the domain data and try to find solutions based on that pair. Once it has found an interim solution, WATCHDOG will ask for the next most important attribute. Once all of the significant attributes have been entered, WATCHDOG will display the final results providing the decision-maker with a list of objects that definitely meets his preferences and a list of uncertain objects. These uncertain objects are objects that may meet his preferences but their uncertainty should be resolved. By providing both lists, the decision-maker should be able to make a better-informed decision.

Furthermore, I evaluated the effectiveness of the WATCHDOG tool through a series of two validation tests. Ten test subjects were surveyed and asked to evaluate several scenarios of minivan data then select the minivan that they would buy, with and without the aid of the tool. Test One did not yield the expected results. It appeared as if the utility of the tool was not apparent because of the simplicity of the data. The tool was not needed. Consequently, the WATCHDOG tool was upgraded, the system documentation and test questions were refined, the data set was made more complex, and Test Two was administered. The consequences of Test Two demonstrated that the test subjects did focus on the uncertain data and that they felt more comfortable with making decisions with the aid of the tool. Several test subjects even changed their responses as a result of using the WATCHDOG tool.

6.2 Conclusion

This research was designed to test two hypotheses I had formulated regarding overcoming information overload and uncertain data. WATCHDOG was created in order to substantiate these hypotheses. My conclusions regarding each hypothesis will be presented individually, starting with the primary hypothesis.

Primary Hypothesis:

A method that can focus the attention of a decision-maker on uncertain information can be developed.

To test this hypothesis, I created the WATCHDOG Decision Support Tool to aid a decision-maker in sorting through large amounts of data, some of it uncertain. By reducing the clutter, WATCHDOG increases the proportion of important data. This ensures that the decision-maker is cognizant of the critical information since the irrelevant data has been eliminated.

From the Validation testing, I can support the claim that the WATCHDOG Decision Support Tool is an adequate method that can be used to focus the attention of the decision-maker on uncertain data. However, initially, the results from Test One did not indicate a focusing of attention. After a quick re-evaluation of the WATCHDOG tool as well as the accompany survey, some of the shortcomings were corrected and a new test was administered. This time, the results from Test Two did indicate that the test subjects' attention was being guided by the decision support tool to the appropriate objects of uncertainty. Once the non-critical objects were trimmed away, the test subjects' attention converged on the issue of mitigating the uncertainty of the alternatives in order to make a better decision.

The secondary hypothesis that was also substantiated is as follows:

Secondary Hypothesis:

As the data set becomes more complex,
the benefit of the WATCHDOG tool will be greater.

This hypothesis stated my belief that the overall usefulness of the WATCHDOG tool would become more evident as the data set become more complex. By allowing the tool to do the essential but laborious sorting and trimming process, the overall time to orient the data into manageable sets should be greatly reduced. The greater the amount of data, the greater the amount of time that would be saved if the decision-maker used the WATCHDOG tool. This would alleviate much of the burden from the decision-maker, making his job that much easier.

The change in the results between Test One and Test Two appear to elaborate the value-added of the WATCHDOG tool. Results from Test One indicated that the test subjects did not feel as comfortable with the tool as the data became slightly more complex. However, the benefits of the tool were very evident in Test Two. Because the data set in Test Two was much larger and more complex than the data set from Test One, the contrast between the comfort levels of these tests would seem to highlight the benefit of the tool.

6.3 Future Research

In the course of this research, a few alternatives for future enhancement have been considered that I was not able to investigate. These topics, if pursed, would become valuable, direct extensions of this research. The following paragraphs describe these future research alternatives.

6.3.1 Optimization of Code / Object-Oriented Approach

The WATCHDOG Decision Support Tool was designed and implemented on a rapid-prototyping scheme. It involved creating a very core, basic program and adding changes in a quick but incremental fashion. These changes would be added and then the whole system would be re-evaluated. New supplemental changes were devised and added. The process would then begin again. WATCHDOG underwent numerous iterations of code, test, re-evaluate, re-design, and code again. Because of this evolutionary development, attention was not adequately given to making the program efficient but simply on getting the program to work correctly. Fundamental verification testing occurred to ensure that the program worked correctly as coded but the final solution is far from optimal. An appropriate direction of subsequent research should be in the direction of the optimization of the program logic.

Furthermore, since this tool was written in JAVA, the next researcher should re-design this decision support tool to be more object-oriented. The JAVA programming language offers so many advantages because of its use of object-oriented programming (OOP) concepts. Although I tried to incorporate some object-oriented features, I realized that I did not even begin to tap into JAVA'S true potential as a robust programming language.

6.3.2 Graphical User Interface (GUI)

As mentioned in Chapter 3, there are two ways of dealing with information overload. First, through better information visualization, and second through the reduction of the non-essential data. I chose to attack the latter of the two and concentrate my efforts on designing WATCHDOG tool. I decided early on in the design process that console input and output would be adequate as a user interface. Any other programming effort would distract me from the true goal and would merely be "bells and whistles." Furthermore, Capt Evan Watkins was attacking a similar problem with the perspective of creating better visualization techniques. Because of his direction, I focused on the fundamental reduction problem.

Ironically, several test subjects commented on a perceived need for a better interface. I agree that before I would deliver this product to a customer, I would create a better graphical user interface (GUI). Moreover, I believe that I've reached the limit with how I can present the information to the user without creating a bigger information overload problem. If a simple GUI would better convey the results given by the WATCHDOG tool, then I believe a GUI should be developed to better aid the decision-maker in understanding the domain data. Better information visualization would seem to be a logical extension of my research

6.3.3 Reasoning with Uncertainty Applications

Most real-world problems involve uncertain data. Many reasoning with uncertainty (RWU) techniques are well suited for handling uncertainty. Furthermore, some RWU techniques, such as Fuzzy Logic, mirror human decision-making and are therefore very intuitive. It would seem logical and natural to introduce some form of RWU processing in a decision support tool such as WATCHDOG. Additionally, because they can be intuitive and seem natural, RWU techniques may provide a higher level of confidence in a decision support tool than was provided by the WATCHDOG tool.

6.3.4 Other "OODA" Tools

As discussed in Chapter 3, WATCHDOG is conceptually one of several OODA (observe, orient, decide, act) tools. WATCHDOG represents the "orient" phase of the OODA decision cycle designed to aid decision-makers in getting inside the enemy's OODA loop. A next step would be to create the other tools of the OODA loop. One of the assumptions of my research was that the domain data was already gathered and in a readable form available to the WATCHDOG tool. Just as WATCHDOG is independent of a domain, a generic tool or method of "observing" or gathering data could be developed. Furthermore, a domain-independent tool or method could be developed that would take the results from the WATCHDOG tool and actually devise rank-ordered course's of action (COAs) with detailed supporting information. This tool would represent the "decide" phase of the OODA loop, aiding the decision-making by providing possible actual decision. In addition, more "orient" phase tools can be created. In his research [WAT00], Capt Watkins has indicated three sources of uncertainty that a decision-maker must deal with. The WATCHDOG tool only addresses one of these sources. It would seem that future research could also address the development of other "orient" tools for the other uncertainty sources.

However, it is my opinion that no tool or method should be created to represent the "act" phase of the OODA loop. No tool or program, now or in the near future, would be able to account for the entire minutia of details that are necessary in many decision-making situations. Research into predicting an enemy's possible reaction have been developed in the form of "commander simulations" [TAL 99] however it is understood that these are merely attempts to mimic a human response and should not be considered without caution. The human element should never be taken out the decision process of vital matters.

6.4 Closing Thoughts

The WATCHDOG Decision Support Tool is a valid method of focusing the attention of the user on uncertain data. It is not meant to take the place of the decision-maker, but rather to aid the decision-maker by guiding his attention towards those objects that are uncertain which might have been overlooked. However, as I discovered through testing, the WATCHDOG tool is very limited and may even be considered brittle. The tool will give the decision-maker the results of its sorting and trimming but sometimes it is not obvious how the tool comes to its conclusions. The WATCHDOG tool still validates both of my research hypotheses. Nonetheless, I would not feel comfortable with delivering the WATCHDOG tool to a customer as a finished product. This tool would need quite a bit of more development to make it more robust as a final product.

Appendix A -- Source Code

The source code for WATCHDOG is not included as part of this document. Those interested in obtaining a copy of the source code should direct their requests to:

Dr. Gregg Gunsch

AFIT/ENG
2950 P Street
WPAFB, OH 45433-7765

gregg.gunsch@afit.af.mil

Bibliography

[Boy87] Boyd, John R. A Discourse on Winning and Losing. Unpublished manuscript. Air University Press, Maxwell AFB AL, 1987.

[Car99] Carlson, Kyla K. "Ultrafast Color Printers," *P.C. Computing Magazine*, March 1999. p. 175-180.

[Cle96] Clemen, Robert T. *Making Hard Decisions*. Duxbury Press, Belmont CA, 1996.

[CON 99] *Consumer Reports*. April 1999, Volume 64, No. 4.

[DL91] Dale, Nell & Susan C. Lilly. *Pascal Plus Data Structures, Algorithms, and Advanced Programming*. D.C. Heath and Company, Lexington MA, 1991.

[DIO99] National Air Intelligence Center. "Introduction to DIODE." Wright Patterson AFB, OH, November 1999.

[Fel97] Feldman, Michael B. *Software Construction and Data Structures with Ada 95*. Addison-Wesley Publishing Company, Reading MA, 1997.

[FM100-6] Field Manual No. 100-6. *Information Operations*. Headquarters, Department of the Army, Washington DC, 27 August 1996.

[GM88] Gulick, Roy M and Martin, Anne W. "Managing Uncertainty in Intelligenc– Data - An Intelligence Imperative." *Science of Command & Control; Coping with Uncertainty*. AFCEA International Press, Washington DC, MD, 1988.

[HC97] Horstmann, Cay S. & Gary Cornell. *Core Java 1.1: Volume 1 – Fundamentals*. Sun Microsystems Press, Mountain View, CA, 1997.

[Joi96] Joint Staff, Department of Defense. Joint Doctrine for Command and Control Warefare (C2W). JP 3-13.1, Director of Operations, Joint Staff, Washington, D.C., 7 February 1996.

[Osb96] Osborne, William B LtCol (USA); Bethel, Scott A Maj; et al. *Information Operations. A New War-Fighting Capability.* Air Force 2025 Future study presented to Joint Chiefs of Staff, August 1996.

[Old99] Oldham, Scott. "Family Trucksters," *Popular Mechanics*, November 1999, Volume 176, No. 11, p. 82-88, & 142.

[Ros97] Ross, Philip E. "The End of Infantry?" *Forbes* 160, no. 1(July 1997): 182-86.

[RN95] Russell, Stuart and Peter Norvig. *Artificial Intelligence A Modern Approach.* Prentice Hall, Upper Saddle River NJ, 1995.

[Sol91] Solso, R.L. Cognitive Psychology, 3rd Ed. Allyn and Bacon, 1991

[Sol93] Soltz, Brian B. *Graphical Tools for Situational Awareness for Large Synthetic Battlespaces.* Masters Thesis. Air Force Institute of Technology, Wright-Patterson AFB OH, December 1993.

[Smi89] Smithson, Michael. *Ignorance and Uncertainty: Emerging Paradigms.* Springer-Verlag, New York NY, 1989.

[Wat00] Watkins, Evan T., Captain, USAF, *Improving The Analysts' And Decision-Makers' Perspective Through Uncertainty Visualization.* Masters Thesis, Air Force Institute of Technology, Wright-Patterson AFB OH, March 2000

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** 2 March 2000	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE

A Method of Focusing the Attention of the Decision-Maker on Uncertain Information

5. FUNDING NUMBERS

6. AUTHOR(S)

John F Moesner IV, Captain, USAF

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 P Street, Building 640
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GCS/ENG/00M-17

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

NAIC/TAIT
Attn: Capt Allen Bobo
4180 Watson Way
Wright-Patterson AFB, OH 45433 DSN: 787-6327

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

Dr Gregg H. Gunsch, ENG DSN: 785-6565, Ext. 4182

12a. DISTRIBUTION / AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 Words)*

Military decision-makers need to be able to synthesize large amounts of information quickly and make accurate and timely decisions. However, all too often, when a decision-maker is bombarded with important information that has an uncertainty associated with it, that information is often neglected. One method of dealing with this type of information overload is through proper data orientation. By reducing the clutter of irrelevant information, the proportions of useful and relevant data can be increased. At the same time, the attention of the decision-maker is directed to the critical tasks or centers of gravity.

The WATCHDOG Decision Support Tool is an attempt to develop such a data orientation method. This tool can aid the decision-maker in sorting through large amounts of uncertain information and guiding his or her focus to the appropriate areas of concern. Conceptually, this tool would be used in the "orient" phase of OODA (Observe, Orient, Decide, and Act) decision cycle. By expediting the "orient" phase, the overall decision process would also be accelerated. The decision-maker's situational awareness would be amplified and the quality and the timelines of a decision would be increased. This thesis effort resulted in the development of the WATCHDOG tool. Experimentation was performed on ten test volunteers with a common underlying experience. Results indicated that the tool proved to be useful in support of the decision-maker when sorting through complex data with high degrees of uncertainty.

14. SUBJECT TERMS Uncertainty, Uncertain Information, Data Orientation, Focusing the Attention, Decision-Making, Decision Analysis, Watchdog, Decision Support Tool	**15. NUMBER OF PAGES** 126 **16. PRICE CODE**

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	**18. SECURITY CLASSIFICATION OF THIS PAGE** UNCLASSIFIED	**19. SECURITY CLASSIFICATION OF ABSTRACT** UNCLASSIFIED	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102